

A11100 984589

A UNITED STATES  
DEPARTMENT OF  
COMMERCE  
PUBLICATION



NBS SPECIAL PUBLICATION 339

Source Listing  
of OMNITAB II  
Program

U.S.  
DEPARTMENT  
OF  
COMMERCE  
National  
Bureau  
of  
Standards







UNITED STATES DEPARTMENT OF COMMERCE • MAURICE H. STANS, *Secretary*

NATIONAL BUREAU OF STANDARDS • LEWIS M. BRANSCOMB, *Director*

## Source Listing of OMNITAB II Program

**Sally T. Peavy, Ruth N. Varner, and David Hogben**

Statistical Engineering Laboratory  
Applied Mathematics Division  
Institute for Basic Standards  
National Bureau of Standards  
Washington, D.C. 20234



U.S. National Bureau of Standards Special Publication 339  
Nat. Bur. Stand. (U.S.), Spec. Publ. 339, 371 pages (Dec. 1970)

**CODEN: XNBSA**

Issued December 1970

NATIONAL BUREAU OF STANDARDS  
APR 5 1971  
L60322  
Q C100  
U 57  
No. 339  
1970  
Copy 2

Library of Congress Catalog Card Number: 79-609406

## **Contents**

	Page
Introduction.....	1
Description.....	1
Programming Techniques.....	2
OMNITAB II Operating System Routines.....	3
References.....	4
Table of Contents for Subprogram Listings.....	5
Listing of Subprograms.....	7
Appendix—OMNIT Flow Chart.....	346



# Source Listing of OMNITAB II Program

Sally T. Peavy, Ruth N. Varner, and David Hogben

OMNITAB II is a general-purpose interpretive computing system designed to allow a nonprogrammer to use a high-speed computer easily, accurately and effectively. The system permits the user to perform arithmetic operations including complex arithmetic, trigonometric calculations, miscellaneous function calculations, statistical analysis, Bessel function calculations, and operations on matrices and arrays.

The OMNITAB II system contains 177 subprograms written in the ANSI FORTRAN language. Every effort has been made to make the system transportable. This publication contains a complete listing of all these subprograms. The listing is preceded by a brief introduction which describes the programming techniques used; the use of system library functions; and the subprograms used to control the flow of operations in the OMNITAB system.

Key words: Accuracy; algorithms; ANSI FORTRAN; documentation; error checking; machine independent; OMNITAB II operating system subprograms; OMNITAB II source listing; programming techniques; transportable; user-oriented computing system.

## 1. Introduction

The OMNITAB II computing system enables the nonprogrammer to use a large digital computer to perform data, statistical, and numerical analysis without having any prior knowledge of computers or computer languages. The user writes simple English instructions to obtain accurate results easily and effectively. The instructions reference reliable, varied, and sophisticated algorithms for analysis and manipulation. The tedious task of formatting data for input and output is alleviated by OMNITAB II through free field input, readable printing, and automatic printouts. Formatting capability exists, if more flexibility is desired. The system permits the user to perform: arithmetic operations including complex arithmetic, trigonometric calculations, data manipulations, miscellaneous function calculations, statistical analysis, numerical analysis, Bessel function calculations, operations on matrices and arrays, and thermodynamic calculations. Although OMNITAB has been specifically designed for nonprogrammers, many experienced programmers find uses for OMNITAB.

OMNITAB was developed at the National Bureau of Standards under the guidance of Joseph Hilsenrath who contributed the basic ideas and philosophy, see Hilsenrath et al. [1966]. The program was written initially in a mixture of symbolic machine language and FORTRAN. With the advent of third generation computers and the success of OMNITAB, it became necessary to rewrite the program and to make it as machine independent as possible. Walter J. Gilbert undertook this task in 1966 and initiated a number of changes and new features. Since 1968, the Statistical Engineering Laboratory, Applied Mathematics Division, has been responsible for the development, maintenance, and management of OMNITAB. NBS OMNITAB II Version 5.0 (Hogben et al. [1970]) is the result of this effort.

## 2. Description

This publication is one of four which constitute the documentation for OMNITAB II. A user's guide is given in Hogben et al. [1970]. A systems programmer's guide is contained in Peavy et al. [1970]. Problems and results for assessing the success of implementing OMNITAB II are given in Varner et al. [1970]. No attempt is made to make this publication independent of the other three. In particular, the reader should consult Hogben et al. [1970] for a general understanding of OMNITAB II and for an explanation of terms used in conjunction with OMNITAB II.

The OMNITAB II software system contains 177 subprograms written in American National Standard (ANSI) FORTRAN language. This publication contains a listing of all of these subprograms in section 7. The subprograms are listed alphabetically by the subprogram name; except for the main program which appears first. Each subprogram has a unique identification in columns 73-75. Sequential line numbers appear in columns 76-79. This provides a means of easily updating the subprogram when improvements are made. Each subprogram has been tidied using the program of Murphy [1966].

A table of contents giving the subprogram name, subprogram identification, and page number is given in section 6 on pages 5 and 6. Along with the identification, symbols are used to indicate the main function of each subprogram. The symbol denotes the main purpose and not necessarily the sole function of the subprogram. For example, OMNIT is an executive subprogram, but it also executes the command NOTE.

The material contained herein is a complete listing of the OMNITAB II program with some exceptions. Subprograms which compute the square root, sin, cos, tan, input and output and other subprograms which are normally part of the operating system library are not included; see Peavy et al. [1970].

Each subprogram has been tested as thoroughly as possible. A few errors which escaped detection in the preliminary testing were subsequently detected in extensive use and corrected. Despite considerable effort to make OMNITAB II error free, a few errors are to be expected in such a large system. We believe that any errors that remain are few in number. Any reader who finds errors or has any comments is encouraged to write to us so that improvements can be passed on to other users.

### 3. Programming Techniques

The subprograms which comprise the OMNITAB II system were the efforts of many individuals working independently with considerable latitude. Thus, the programming techniques used in the subprograms are varied. However, each subprogram had to satisfy the following criteria.

(a) *Programming language.* All subprograms are written in the ANSI FORTRAN language. Thus, the routines which scan the OMNITAB II instructions, manipulate and pack characters, and involve input-output are lengthy and to a limited extent inefficient. These routines could be written in machine language for a particular computer installation if efficiency were sufficiently important. However, more attention has been devoted to developing a system which is transportable and easily used. Since OMNITAB II is designed primarily for the analysis of small-to-moderate amounts of data, these considerations are far more important than machine efficiency up to a certain point.

Every effort was made to produce machine independent subprograms in order to make the entire system transportable. Where it was necessary to supply alphanumeric information, a maximum of three characters per machine word was permitted. The only exception is in the case where user defined formats are packed to a full capacity of the machine word. Formats containing information for printout were restricted to 120 characters per line. Variable names were used in all references to input-output devices. These special features are discussed further in Peavy et al. [1970]. All FORTRAN compilers do not accept every ANSI FORTRAN statement and efforts were made to avoid the use of such statements.

(b) *Algorithms.* The policy is to use the most reliable and accurate algorithms. In some instances it was not feasible to adhere to this policy completely. Sometimes, a good algorithm was accepted with the intention of improving it at a later date. This is particularly true for the commands with comprehensive automatic printing. Again, efficiency was subordinated to accuracy when necessary. Most calculations are performed using single precision floating-point arithmetic. In a few instances, where more accuracy was deemed desirable, double precision floating-point arithmetic was used. Double

precision arithmetic was used primarily to obtain full single precision accuracy.

(c) *Error checking.* Every subprogram which executes a command is required to do extensive error checking. Besides checking for correct number and type of arguments and previous errors, the subprograms have to perform unique checks for the particular command that is to be executed. Specific messages are printed and execution of further OMNITAB instructions may be affected depending upon the severity of the error. However, this does not prevent OMNITAB from scanning the remaining instructions for errors unless the instruction is stored. See Hogben et al. [1970] and Peavy et al. [1970] for further details.

(d) *Elementary functions.* The original OMNITAB had its own programs to compute the elementary functions (SIN, LOGE, etc.) which were of known accuracy. In order to make OMNITAB II as transportable as possible, this procedure was not followed and the system library functions of the particular computer system are used. The accuracy of system library subroutines varies from one computer to another. Some are quite accurate; others are surprisingly inaccurate as was explained excellently by Cody [1970].

Many system library functions terminate execution if an arithmetic fault is encountered; for example, in computing the square root of a negative number. This confuses and hinders the nonprogrammer since the messages upon termination are cryptic and often the fault is not significant. Efforts have been made to eliminate this nuisance in OMNITAB II. The subprograms are not permitted to use any of the library functions directly. Instead, they reference other function routines which check the values of the arguments to be used by the system library functions. See Peavy et al. [1970] for a list of these function routines. Only if the values are arithmetically legitimate and within certain bounds will the system library functions be used. Otherwise, a very explicit diagnostic message will be printed. The result is set equal to zero and execution is continued.

(e) *Input-output routines.* Input-output conversion routines were also part of the old OMNITAB, but again these had to be dropped in favor of using operating system routines to make OMNITAB more transportable. There is one important exception. The subprogram RFORMAT does the necessary output conversion for many of the basic printing commands. RFORMAT does its own conversion to enable the printing of numbers in a more readable form. The conversion is as accurate as or more accurate than the standard operating system output conversion routines. See Hogben [1970] for further details.

(f) *Division by zero.* Computer operating systems handle division by zero in many different ways. In some instances a run may be terminated, in other cases it may be ignored, and sometimes just

a message is printed that division by zero was attempted. The OMNITAB II commands which specifically call for division, such as DIVIDE and ADIVIDE, provide a check on zero before division is attempted. If the divisor is zero, the result is set equal to zero, a message is printed, and execution is continued.

In some commands division by zero can occur in nonobvious places depending upon the algorithm which is used, as in the FIT command. When it is known that a divisor can equal zero, a check on the divisor was provided. However, a check on the divisor is not made in every instance that division is used. It is possible that an error termination could occur, but it is not likely.

(g) *Comment statements.* Comments in the form of FORTRAN comment statements are dispersed in most subprograms. Some of the comments are very detailed and complete while others are rather limited. The use of comments was at the discretion of the programmer and no standard style or format was required. Due to recent changes in OMNITAB II, some of the comments may not be pertinent. No effort has been made to edit the comments.

(h) *Stand alone subprograms.* Some of the subprograms may be used separately outside the context of the OMNITAB II system. More often, the programmer took advantage of the OMNITAB II system when writing a particular subprogram. A list of all the stand alone programs is given below. Some use the OMNITAB II elementary function routines and would have to be modified accordingly. These exceptions are shown in parentheses. A number of other subprograms, not listed here, such as RFORMAT could be easily modified to make them stand alone.

ACCDIG	(FLOG10)	computes the accuracy of one number compared to another
BEZERO	(FDSQRT)	computes zeros of Bessel function of order zero
BEZONE		computes zeros of Bessel function of order one
BJORCK		least squares quadratic fit for CORRELATION
CBEI	(FDSIN, FDCOS, FDSQRT, FDEXP)	computes Bessel functions of complex argument, with and without scale factors
DETRNK		computes the determinant and rank of a matrix
FREQCY	(FLOG10)	computes the frequency distribution of a column of numbers
HDIAG	(FSQRT)	computes eigenvalues and eigenvectors of a matrix
INTRP		provides n-point Lagrangian interpolation

MXTXP	(calls SORTSM below) performs matrix multiplication X'X or XX'
PVTRI	determines if matrix is upper or lower triangular
RANKO	(equivalent to RANKX below)
RANKX	determines ranks of a set of numbers
RCSUM	computes row and column sums of a matrix
RNJBK	machine independent pseudo random number generator
SKSYMV	tests for skew symmetry of a matrix
SORTSM	sorts products of matrix multiplication and sum
SPINV	matrix inversion with minimum round-off error accumulation
SYMV	test for matrix symmetry

## 4. OMNITAB II Operating System Routines

The main program of the OMNITAB II system is small and contains one call to the OMNIT subprogram. The OMNIT and XECUTE subprograms are the two major routines in the OMNITAB II system. A detailed flow chart of OMNIT, prepared using CAL-COMP [1968], is given in the appendix. A capsule summary of the chief function of each of the 34 executive subprograms is given below. In addition, there are four block data subprograms which are BLOCK, LBCONS, LOOKTB and PHYSIC. The reader can consult Peavy et al. [1970] for further details.

Main Program	calls OMNIT
AARGS	converts arguments of an instruction
ADRESS	computes memory location of first row of worksheet column
AERR	tallys and prints arithmetic faults
ASTER	checks proper use of asterisks
CHKCOL	checks for valid column numbers and computes memory locations
CKIND	checks mode of instruction arguments
ERROR	prints fatal errors
EXPAND	sets up IARGS, ARGS and KIND for use of arguments by subprograms
INFERR	prints informative diagnostics
INPUT	reads all cards
LOCATE	locates stored instructions
LOOKUP	assigns values to L1 and L2 which are used by XECUTE

MTXCHK	determines if matrix fits in worksheet and computes memory location
NNAME	converts command name to a unique number
NONBLA	looks for next non-blank character in scanning card
OMCONV	converts input card images to standard code
OMNIT	main subprogram which controls flow of operations
OUTPUT	outputs card images to scratch unit
PACK	packs or unpacks alphanumeric characters
PREPAK	controls packing/unpacking of characters for OMNITAB II system
READQ	converts and stores data
RNDOWN	prints location of error in stored instructions
SETQ	converts and stores data
SETUP	initializes system at the very beginning
STMT	assembles and checks an instruction number
STORE	stores numbered instructions
TAPOP	sets up arguments for tape operation commands
VARCON	checks for legitimate variables within asterisk
VECTOR	vectorizes a constant
XECUTE	calls appropriate subprogram to execute an instruction
XOMNIT	initializing for OMNITAB command
XPND	sets up IARGS, ARGS and KIND for stored instructions
XSTOP	terminates use of OMNITAB and returns control to operating system

---

authors, not only programmed many of the subprograms, but checked them out as well: Robert C. McClenon, Carla G. Messina, Bradley A. Peavy, and Philip J. Walsh. M. Stuart Scott did a fine job of developing CORRELATION. Irene A. Stegun and Ruth Zucker provided the subroutine ERRINT which is used for the commands ERROR and CERF; see Stegun and Zucker [1970]. Roy H. Wampler and William J. Hall provided the subroutine BJORCK for use with the subprogram CORREL. We thank John Mandel for providing an approximation to percentage points of the studentized range, which is used in the subprogram ONEWAY. Shirley G. Bremer gave valuable assistance in general maintenance and housekeeping. We thank Bruce W. Ramsay and Robert J. Arms of the Computer Services Division for their administrative and technical support. The staff of the Statistical Engineering Laboratory, under Joan R. Rosenblatt, made invaluable contributions and suggestions arising out of their varied experiences in statistical consulting. Carla G. Messina merits additional thanks for preparing section 7 for computerized phototypesetting. Last, but not least, special thanks are given to all the users who made valuable comments and stimulated modifications and additions to the system.

## 5. References

- American Standard Fortran (1966). American National Standards Institute, New York.
- CAL-COMP (1968). Flowgen/F—A Fortran Flowchart Generator. California Computer Products, Inc.
- Cody, W. J. (1970). Software for the elementary functions. Presented at Mathematical Software Symposium, Purdue University, April 1–3.
- Hilsenrath, J., Ziegler, G. G., Messina, C. G., Walsh, P. J., and Herbold, R. J. (1966). OMNITAB: A Computer Program For Statistical And Numerical Analysis. Nat. Bur. Stand. (U.S.), Handb. 101, Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402. Reissued January 1968 with corrections.
- Hogben, David (1970). Readable printing of single precision numbers. To be submitted to J. Res. Nat. Bur. Stand.
- Hogben, David, Peavy, S. T., and Varner, R. N. (1970). OMNITAB II User's Reference Manual. Nat. Bur. Stand. (U.S.), Tech. Note 552. Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.
- Murphy, H. M., Jr. (1966). Tidy, a computer code for renumbering and editing Fortran source programs. Air Force Systems Command, Kirkland Air Force Base, New Mexico.
- Peavy, S. T., Varner, R. N., and Bremer, S. G. (1970). A Systems Programmer's Guide For Implementing OMNITAB II. Nat. Bur. Stand. (U.S.), Tech. Note 550, Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.
- Stegun, Irene A., and Zucker, Ruth (1970). Automatic computing for special functions. J. Res. Nat. Bur. Stand. (U.S.), 74B (Math. Sci.), No. 3, 211–225, July–Sept.
- Varner, R. N., and Peavy, S. T. (1970). Test Problems And Results For OMNITAB II. Nat. Bur. Stand. (U.S.), Tech. Note 551, Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

OMNITAB II is a large system and its success is due in no small part to the fact that many specialists have contributed to its development. We especially thank J. Hilsenrath for his continued interest and advice. Walter J. Gilbert laid the foundation for OMNITAB II by developing the scan routine, many of the basic subroutines, and by adding several new features. Joseph M. Cameron formulated the set of instructions for matrix and array operations in addition to providing guidance, particularly in the development of the least squares curve fitting instructions. The following individuals, besides the

## 6. Table of Contents for Subprogram Listings

The following symbols are used to indicate the main function of each subprogram. The number of subprograms in each category is given in parentheses on the extreme right.

E	OMNITAB II executive system control subprogram	(34)
B	Block data executive subprogram	( 4)
F	Function routine as defined in 3 (d)	(14)
I	executes instruction(s) (and performs calculations)	(77)
C	performs computations for subprogram(s) which execute instruction(s)	(48)

Subprogram name	Identification	Page	Subprogram name	Identification	Page
Main Program					
AARGS	AAR E	8	EXPAND	EXD E	78
ABRIDG	ABR I	10	EXPCON	EXN I	80
ACCDIG	ACC C	11	EXTREM	EXT I	82
ADRESS	ADR E	11	FCOS	FCO F	83
AERR	AER E	12	FDCOS	FDC F	83
ALLSUB	ALL I	14	FDEXP	FDE F	83
APRINT	APR I	17	FDLOG	FDL F	84
ARITH	ARI I	20	FDPCON	FDP F	84
ARYVEC	ARY I	22	FDSIN	FDS F	84
ASTER	AST E	25	FDSQRT	FDQ F	85
BEGIN	BEG I	27	FEXP	FEX F	85
BEJN	BEJ C	28	FEXP2	FX2 F	85
BESSEL	BES I	30	FIXFLO	FIX I	86
BEZERO	BEZ C	36	FLIP	FLI I	87
BEZONE	BEO C	37	FLOG	FLE F	88
BINTJO	BIN C	38	FLOG10	FLT F	88
BJORCK	BJO C	39	FNEC	FNE I	89
BLOCK	BLO B	41	FNEIC	FNC I	90
CBEI	CBI C	42	FNKC	FKC I	91
CBEK	CBK C	44	FOURI	FOU C	92
CHANGE	CHA I	46	FPPT	FPP C	93
CHKCOL	CHK E	46	FPROB	FPR I	94
CKIND	CKI E	47	FRDIST	FRD I	95
CMPARA	CMP C	47	FREQCY	FRE C	96
CMSEPA	CMS I	48	FSIN	FSI F	97
COALES	COA I	51	FSQRT	FSQ F	97
COMELL	COM C	54	FTANH	FTA F	97
COMPLX	COX I	55	FUNCT	FUN I	98
CORREL	COR I	57	GENER	GEN I	102
CSPINV	CSP C	62	GQUAD	QUA I	103
DBEJ	DBE C	64	HDIAG	HDI C	104
DEFINE	DEF I	66	HEADS	HEA C	108
DETRNK	DET C	67	HISTGM	HIS I	110
DHRND	DHR C	68	IFS	IFS I	111
DIMENS	DIM I	69	INFERR	INF E	113
DUMMYA	DMA I	70	INPUT	INP E	116
DUMMYB	DMB I	70	INTERP	INT I	117
DUMMYC	DMC I	70	INTRP	INR C	119
DUMMYD	DMD I	70	INVCHK	INK C	121
DUMMYE	DME I	70	INVCOR	INC C	123
DUMMYF	DMF I	70	INVERT	INV I	125
ERASE	ERA I	71	ITERAT	ITE I	127
ERRINT	ERT C	72	LBCONS	LBC B	132
ERROR	ERR E	74	LIST	LIS I	133

## 6. Table of Contents for Subprogram Listings (continued)

Subprogram name	Identification	Page	Subprogram name	Identification	Page
LOCATE	LOC E	133	PUNCH	PUN I	255
LOOKTB	LOT B	134	PVTRI	PVT C	256
LOOKUP	LOU E	143	RANKO	RKO C	257
MATRIX	MAT I	145	RANKS	RAS I	258
MDAMAD	MDA I	149	RANKX	RAX C	259
MEIGEN	MEI I	151	RCSUM	RCS C	260
MISC2	MIS I	153	READQ	REQ E	261
MIST	MST C	156	READX	REX I	262
MKRON	MKR I	158	REPINC	REP I	264
MMULT	MMU I	160	RESET	RES I	268
MOP	MOP I	162	RFORMAT	RFO C	269
MOVE	MOV I	164	RNDOWN	RND E	273
MPROP	MPR I	165	RNJBK	RNJ C	273
MRAISE	MRA I	173	RPRINT	RPR C	274
MSCROW	MSC I	176	SELECT	SEL I	279
MTRIAN	MTR I	178	SET	SET I	282
MTXCHK	MCK E	181	SETQ	STQ E	283
MXTX	MXT I	182	SETUP	STP E	284
MXTXP	MXP C	184	SKSYMV	SKS C	285
NNAME	NNA E	185	SORDER	SOD I	286
NONBLA	NON E	186	SORTSM	SOM C	288
NOTEPR	NOT I	186	SPACE	SPA I	289
OANOVA	OAN C	187	SPINV	SPI C	290
OCOEFF	OCO C	189	STATIS	STA I	292
OCOVAR	OCV C	191	STMT	STM E	300
OMCONV	OMC E	192	STORE	STO E	301
OMNIT	OMN E	193	STORMT	STT C	302
ONEWAY	ONE I	199	STRUVE	STR C	303
OPONE	OPO C	206	SYMV	SYM C	303
ORTHO	ORT I	208	TAPOP	TAP E	304
ORTHRV	ORV C	225	TAPOP2	TP2 I	305
ORTPLT	ORP C	227	THERMO	THE I	307
OUTPUT	OUT E	230	TPCTPT	TPC C	313
PACK	PAC E	231	TRANSF	TRA C	314
PAGE	PAG I	232	TWOWAY	TWO I	316
PDMOTE	PDM I	233	VARCON	VAR E	332
PHYCON	PHY I	235	VECTOR	VEC E	332
PHYSIC	PHC B	236	XECUTE	XEC E	333
PLOT	PLO I	238	XFORMT	XFO I	341
PREPAK	PRE E	245	XHEAD	XHE I	341
PRINTX	PRI I	248	XOMNIT	XOM E	342
PROB	PRB C	250	XPND	XPN E	344
PROCHK	PRK C	252	XSTOP	XST E	345
PROROW	PRO I	253			

## 7. Listing of Subprograms

C	THIS IS THE MAIN PROGRAM OF OMNITAB	OMS 10
C	VERSION 5.00 OMNSYM 5/15/70	OMS 20
C		OMS 30
	CALL OMNIT	OMS 40
	STOP	OMS 50
	END	OMS 60

```

SUBROUTINE AARGS                                AAR  10
C      VERSION 5.00      AARGS      5/15/70        AAR  20
COMMON /BLOCKA/ MODE,M,KARD(83),KARG,ARG,ARG2,NEWCD(80),KRDEND AAR  30
COMMON /CONSTS/ PI,E,HALFPI,DEG,RAD,XALOG       AAR  40
C
C      THIS SUBROUTINE ASSEMBLES A FLOATING POINT NUMBER FROM A STRING OF AAR  60
C      DIGITS ETC. M INITIALLY POINTS AT THE FIRST NUMBER. IT IS LEFT      AAR  70
C      POINTING AT THE FIRST CHARACTER AFTER THE NUMBER.                  AAR  80
C
C      VALUE RETURNED IN ARG                                         AAR  90
C
C      KARG = 1 = FLOATING POINT, = 0 = INTEGER, -1 = ERROR.          AAR 100
C
C      ARG=KARD(M)                                              AAR 110
C      SIGN=1.                                                 AAR 120
C      JEXP=0                                                 AAR 130
C      IXS=1                                                 AAR 140
C      IEXP=0                                                 AAR 150
C      KARG=0                                                 AAR 160
C
C      LOOK BACK FOR MINUS SIGN AND/OR DECIMAL POINT             AAR 170
C
C      K=KARD(M-1)                                              AAR 180
C      IF (K.NE.37) GO TO 10                                     AAR 190
C      KARG=1                                                 AAR 200
C      IEXP=-1                                               AAR 210
C      K=KARD(M-2)                                              AAR 220
C      IF (K.EQ.38) SIGN=-1.                                     AAR 230
C      10 M=M+1                                                 AAR 240
C      K=KARD(M)                                              AAR 250
C      IF (K.GE.10) GO TO 30                                     AAR 260
C      IEXP=IEXP-KARG                                         AAR 270
C      ARG=10.*ARG+FLOAT(K)                                    AAR 280
C      GO TO 20                                               AAR 290
C      20 IF (K.NE.37) GO TO 50                                     AAR 300
C
C      DECIMAL POINT FOUND                                     AAR 310
C
C      IF (KARG.EQ.0) GO TO 40                                     AAR 320
C      CALL ERROR (3)                                           AAR 330
C      KARG=-1                                               AAR 340
C      RETURN                                                AAR 350
C
C      30 KARG=1                                               AAR 360
C      GO TO 20                                              AAR 370
C
C      CHECK FOR EXPONENT   E X, E+X, E-X, +X, -X           AAR 380
C
C      40 IF (K.NE.14) GO TO 65                                     AAR 390
C      M=M+1                                                 AAR 400
C      K=KARD(M)                                              AAR 410
C      IF (K.NE.44) IF (K-10) 70,65,65                         AAR 420
C
C      50 M=M+1                                                 AAR 430
C      K=KARD(M)                                              AAR 440
C      IF (K-10) 70,90,90                                       AAR 450
C
C      60 IF (K.NE.38) IF (K-39) 90,60,90                         AAR 460
C      IXS=-1                                               AAR 470
C      GO TO 60                                              AAR 480
C
C      70 KARG=KARG+1                                         AAR 490
C      JEXP=10.*JEXP+K                                       AAR 500
C
C      80

```

```

M=M+1                                AAR 580
K=KARD(M)                            AAR 590
IF (K.LT.10) GO TO 80                AAR 600
C                                         AAR 610
C   DONE WITH ARGUMENT               AAR 620
C                                         AAR 630
90  IF (KARG.NE.0) GO TO 110          AAR 640
100 ARG=SIGN*ARG                      AAR 650
    RETURN                           AAR 660
110 KARG=1                            AAR 670
    IEXP=IXS*JEXP+IEXP              AAR 680
C                                         AAR 690
C   THE FOLLOWING CODING YIELDS MORE ACCURATE RESULTS THAN THE AAR 700
C   OBVIOUS      ARG = ARG * 10. * IEXP                         AAR 710
C                                         AAR 720
    JEXP=IABS(IEXP)                  AAR 730
    IF (JEXP.GT.IFIX(XALOG)) GO TO 140             AAR 740
    IF (IEXP) 120,100,130                 AAR 750
120  ARG=ARG/FEXP2(10.0,FLOAT(JEXP))           AAR 760
    GO TO 100                           AAR 770
130  ARG=ARG*FEXP2(10.0,FLOAT(JEXP))           AAR 780
    GO TO 100                           AAR 790
140  CALL ERROR (102)                   AAR 800
    ARG=0.                             AAR 810
    GO TO 100                           AAR 820
    END                               AAR 830

```

```

SUBROUTINE ABRIDG          ABR 10
C VERSION 5.00    ABRIDG      5/15/70   ABR 20
C
C THE COMMAND ABRIDGE MAY BE USE IN THE FOLLOWING WAYS   ABR 30
C ABRIDGE ROW,, OF COL ++,++ , (USE RPRINT UNLESS IOSWT HAS BEEN   ABR 40
C SET BY FIXED OR FLOATING)   ABR 50
C ABRIDGE // ROW,, OF COL ++,++ (USE SPECIFIED FORMAT)   ABR 60
C ABRIDGE WITH FLOATING PT. ARGS USES RPRINT   ABR 70
C                                     IOSWT IS NOT RESET   ABR 80
C
C COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG   ABR 90
C COMMON /BLOCRC/ NRC,RC(12600)   ABR 100
C DIMENSION ARGS(100)   ABR 110
C EQUIVALENCE (ARGS(1),RC(12501))   ABR 120
C COMMON /FMAT/ IFMTX(6),IOSWT,IFMTS(6),LHEAD(96)   ABR 130
C COMMON /HEADER/ NOCARD(80),ITLE(60,6),LNCNT,IPRINT,NPAGE,IPUNCH   ABR 140
C COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NABR   ABR 150
1AARGS,VWXYZ(8),NERROR   ABR 160
COMMON /KFMT/ KFMT(100)   ABR 170
IF (NARGS.NE.0) GO TO 15   ABR 180
10 CALL ERROR (205)   ABR 190
RETURN   ABR 200
15 IF (L2.EQ.1) IF (IOSWT) 70,70,80   ABR 210
CALL PREPAK (4,IND,L2,IND,KFMT)   ABR 220
IF (IND.NE.0) GO TO 90   ABR 230
IP=1   ABR 240
IF (NARGS.LE.1) GO TO 10   ABR 250
20 LL=IARGS(1)   ABR 260
IARGS(1)=1   ABR 270
IF (LL.LE.0.OR.LL.GT.NROW) GO TO 10   ABR 280
CALL CHKCOL (I)   ABR 290
IF (I.NE.0) GO TO 10   ABR 300
IF (NERROR.NE.0) RETURN   ABR 310
DO 30 I=2,NARGS   ABR 320
J=IARGS(I)+LL   ABR 330
30 ARGS(I)=RC(J-1)   ABR 340
IF (NPAGE.EQ.0) CALL PAGE(0)   ABR 350
GO TO (40,50), IP   ABR 355
40 WRITE (IPRINT,KFMT) (ARGS(I),I=2,NARGS)   ABR 360
GO TO 60   ABR 370
50 WRITE (IPRINT,IFMTX) (ARGS(I),I=2,NARGS)   ABR 380
60 RETURN   ABR 390
70 IF(NPAGE.EQ.0) CALL PAGE(0)   ABR 400
CALL RPRINT   ABR 410
RETURN   ABR 420
C USE SPECIFIED FIXED OR FLOATING FORMAT   ABR 430
80 IP=2   ABR 440
GO TO 20   ABR 450
90 CALL ERROR (222)   ABR 460
GO TO 70   ABR 470
END   ABR 480

```

```

C SUBROUTINE ACCDIG (AX,X,AD,N) ACC 10
C VERSION 5.00 ACCDIG 5/15/70 ACC 20
C RETURNS NUMBER OF ACCURATE DIGITS, AD, IN AX AN APPROXIMATION TO XACC 30
C WRITTEN BY DAVID HOGBEN, SEL, NBS. 10/29/69. ACC 40
C DATA ADMAX /8.0/ ACC 50
C DIMENSION AX(1),X(1),AD(1) ACC 60
C DO 100 I=1,N ACC 70
C DIFF = AX(I)-X(I) ACC 80
C IF (DIFF) 20,10,20 ACC 90
10  AD(I) = ADMAX ACC 100
C GO TO 100 ACC 110
20  AD(I) = -FLOG10(ABS(DIFF)) + FLOG10(ABS(X(I))) ACC 120
C AD(I) = AMIN1(ADMAX,AD(I)) ACC 130
C AD(I) = AMAX1(-ADMAX,AD(I)) ACC 140
100 CONTINUE ACC 150
C RETURN ACC 160
C END ACC 170

```

```

C SUBROUTINE ADRESS (I,J) ADR 10
C VERSION 5.00 ADRESS 5/15/70 ADR 20
C COMMON /BLOCRC/ NRC,RC(12600) ADR 30
C COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NADR 40
1ARGS,VWXYZ(8),NERROR ADR 50
C DIMENSION ARGS(100) ADR 60
C EQUIVALENCE (ARGS(1),RC(12501)) ADR 70
C ADR 80
C CALCULATE ADDRESS OF ARGUMENT( I ). IF ARGUMENT( I ) IS A ADR 90
C FLOATING POINT NUMBER, J = -(I+ NRC). IF ILLEGAL COLUMN NUMBER ADR 100
C J = 0. IF OK, J = ADDRESS ADR 110
C ADR 120
C IF (KIND(I).EQ.0) GO TO 10 ADR 130
J=-(I+NRC) ADR 150
GO TO 30 ADR 160
10 IF (IARGS(I).GE.1.AND.IARGS(I).LE.NCOL) GO TO 20 ADR 170
J=0 ADR 180
GO TO 30 ADR 190
20 J=NROW*(IARGS(I)-1)+1 ADR 200
30 RETURN ADR 210
C END ADR 220

```

```

SUBROUTINE AERR (I)                                     AER 10
C VERSION 5.00      AERR      5/15/70                 AER 20
COMMON /BLOCKC/ KIO,INUNIT,ISCRAT,KBDOUT,KRDKNT,LLIST   AER 30
COMMON /BLOCKX/ INDEX(6,8),LEVEL                      AER 40
COMMON /SPRV/ NERCON,NERR,ISWERR                     AER 50
C                                                       AER 60
C WHEN ARITHMETIC TROUBLES DEVELOP, THIS ROUTINE TALLIES THEM AND THAER 70
C PRINTS THE RESULTS WHEN THE COMMAND IS DONE.          AER 80
C                                                       AER 90
C ARIHMETIC MESSAGES MUST HAVE THE FOLLOWING TYPE OF FORMAT: AER 100
C 1) THE FIRST TWO CHARACTERS MUST BE **                AER 110
C 2) MESSAGE PLUS OTHER INFO MUST NOT BE LONGER THAN 84 CHARACTERSAER 120
C 3) IF MESSAGE IS LESS THAN 84 CHAR , ADD NX AT END OF FORMAT AER 130
C                                                       AER 140
C IF MORE THAN 10 ARITHMETIC ERROR MESSAGES ARE NEEDED THEN AER 150
C DIMENSION OF MESS(10) MUST BE CHANGED AND KMESS MUST BE SET = AER 160
C TO DIMENSION SIZE OF MESS.                           AER 170
C ALSO COMPUTED GO TO MUST BE CHANGED. SEE NOTEAER 180
DIMENSION MESS(10)                                    AER 190
DATA KMESS/10/                                       AER 200
IF (I) 160,30,10                                      AER 210
C                                                       AER 220
C DATA COMING IN                                       AER 230
C                                                       AER 240
10 J=MIN0(I,KMESS)                                    AER 250
MESS(J)=MESS(J)+1                                     AER 260
20 RETURN                                            AER 270
C                                                       AER 280
C DUMP RESULTS, END OF COMMAND                       AER 290
C                                                       AER 300
30 IF(LLIST.LT.2.OR.LLIST.EQ.4) GO TO 160            AER 310
DO 150 J=1,KMESS                                     AER 320
IF (MESS(J).EQ.0) GO TO 150                         AER 330
WRITE (ISCRAT,250)                                   AER 340
WRITE (ISCRAT,180) MESS(J)                          AER 350
C                                                       AER 360
C THIS COMPUTED GO TO MUST BE CHANGED IF MORE THAT 10 ARITHMETIC AER 370
C ERRORS ARE ADDED                                     AER 380
C                                                       AER 390
GO TO (40,50,60,70,80,90,100,110,120,130), J       AER 400
40 WRITE (ISCRAT,101)                                 AER 410
GO TO 140                                           AER 420
50 WRITE (ISCRAT,102)                                 AER 430
GO TO 140                                           AER 440
60 WRITE (ISCRAT,103)                                 AER 450
GO TO 140                                           AER 460
70 WRITE (ISCRAT,104) MESS(J)                        AER 470
GO TO 140                                           AER 480
80 WRITE (ISCRAT,105) MESS(J)                        AER 490
GO TO 140                                           AER 500
90 WRITE (ISCRAT,106) MESS(J)                        AER 505
GO TO 140                                           AER 510
100 WRITE (ISCRAT,107) MESS(J)                        AER 515
GO TO 140                                           AER 517
110 WRITE (ISCRAT,108) MESS(J)                        AER 520
GO TO 140                                           AER 525
120 WRITE (ISCRAT,109) MESS(J)                        AER 530
GO TO 140                                           AER 535
130 WRITE (ISCRAT,240) J                            AER 540

```

```

140 IF (LEVEL.NE.0) CALL RNDOWN          AER 550
      WRITE (ISCRAT,250)
      MESS(J)=0
150 CONTINUE
      IF (LEVEL.NE.0) GO TO 20           AER 590
      ISWERR=0                          AER 600
      NERR=0                           AER 610
      GO TO 20                         AER 620
C
C   INITIALIZATION SECTION           AER 630
C
160 DO 170 J=1,KMESS                AER 640
170 MESS(J)=0
      ISWERR=0
      NERR=0
      GO TO 20                         AER 690
C
180 FORMAT (51H** ARITHMETIC FAULT IN ABOVE COMMAND, ZERO RETURNED,I4,AER 720
      16H TIMES,23X)                   AER 730
101 FORMAT (42H** NEGATIVE ARGUMENT TO SQRT, LOG OR RAISE,42X)    AER 740
102 FORMAT (43H** EVALUATION OF EXPONENT PRODUCES OVERFLOW,41X)    AER 750
103 FORMAT (45H** ARGUMENT OUT OF BOUNDS TO INVERSE FUNCTION,39X)    AER 760
104 FORMAT (51H**ARGUMENT TOO LARGE FOR SIN OR COS, ZERO. RETURNED,I4,AER 770
      16H TIMES,23X)                   AER 780
105 FORMAT (61H**BESSEL ARGUMENTS SCALED TO AVOID OVER/UNDER FLOW.RE AER 790
      1TURNED,I4,6H TIMES,13X)         AER 800
240 FORMAT (16H** ERROR MESSAGE,I2,66X)                   AER 810
250 FORMAT (84X)                               AER 820
106 FORMAT (33H** DIVISION BY ZERO, RESULT SET=0,I4,6H TIMES,41X)    AER 830
107 FORMAT(44H** TRIG FUNCTIONS NOT DEFINED, RESULTS SET=0,I4, 6H TIMEAER 840
      1S, 30X)                         AER 850
108 FORMAT(66H** ONE OF THE VALUES COMPARED IS ZERO, ABSOLUTE TOLERANCAER 860
      1E WAS USED,I4,6H TIMES,8X)       AER 870
109 FORMAT(71H* X FOR ELLIPTICAL INTEGRALS IS = 1.0 OR GREATER. RESULTAER 880
      1 IS SET TO 0.0.,I4,6H TIMES,3X)    AER 890
      END                                AER 900

```

```

SUBROUTINE ALLSUB
C VERSION 5.00      ALLSUB      5/15/70          ALL  10
C COMMON /BLOCRC/ NRC,RC(12600)          ALL  20
C COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NALL 40
IARGS,VWXYZ(8),NERROR          ALL  50
DIMENSION ARGS(100)          ALL  60
EQUIVALENCE (ARGS(1),RC(12501))          ALL  70
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG          ALL  80
COMMON /SCRAT/ NS,NS2,A(13500)          ALL  90
DIMENSION SCRA(1)          ALL 100
EQUIVALENCE (SCRA,A)          ALL 110
EQUIVALENCE (L11,LL1), (L22,LL2)          ALL 120
C PROGRAMMED BY PHILIP J. WALSH (NBS 453.40) MAY, 1967          ALL 130
C                                         ALL 140
C                                         ALL 150
C COMMAND IS OF THE FORM XXXX OF ORDER ++ OF COL ++, STORE IN ++
C XXXX MAY BE (A) NLSUB FOR NORMALIZED LAGUERRE POLYNOMIALS          ALL 160
C                  (B) LSUB FOR LAGUERRE POLYNOMIALS          ALL 170
C                  (C) HSUB FOR HERMITE POLYNOMIALS          ALL 180
C                  (D) USUB FOR CHEBYSHEV POLYNOMIALS          ALL 190
C                  (E) PSUB FOR LEGENDRE POLYNOMIALS          ALL 200
C                  (F) TSUB FOR CHEBYSHEV POLYNOMIALS          ALL 210
C SEE RECURSIVE FORMULAE FOR THESE POLYNOMIALS FURTHER IN CODE          ALL 220
C EACH OF THE COMMANDS REQUIRE THREE ARGUMENTS          ALL 230
IF (NARGS.EQ.3) GO TO 10          ALL 240
CALL ERROR (10)
GO TO 210
10 IF (KIND(1)+KIND(3).EQ.0) GO TO 30          ALL 250
20 CALL ERROR (3)
GO TO 210
30 CHECK THAT X IS WITHIN WORKSHEET AND GET ADDRESS OF ARGUMENT COLUMALL 310
CALL ADRESS (2,L11)
IF (L11) 20,40,50          ALL 320
40 CALL ERROR (11)
GO TO 210
50 IARGS(4)=IARGS(1)+IARGS(3)-1          ALL 330
KIND(4)=0          ALL 340
CALL ADRESS (4,L22)
IF (L22.LE.0) GO TO 40          ALL 350
CALL ADRESS (3,L22)
IF (NRMAX.NE.0) GO TO 60          ALL 360
CALL ERROR (9)
GO TO 210
60 IF (NERROR.NE.0) GO TO 210          ALL 370
IJK=LL1
IJ=LL2
DO 110 I=1,NRMAX          ALL 380
SCRA(1)=RC(IJK)
GO TO (70,70,80,80,90,90), L2          ALL 390
70 RC(IJ)=1.-SCRA(1)
GO TO 100
80 RC(IJ)=2.*SCRA(1)
GO TO 100
90 RC(IJ)=SCRA(1)
100 IJK=IJK+1
IJ=IJ+1
IF (IARGS(1).EQ.1) GO TO 210          ALL 400
N=IARGS(1)-1
DO 200 J=1,NRMAX          ALL 410
                                         ALL 420
                                         ALL 430
                                         ALL 440
                                         ALL 450
                                         ALL 460
                                         ALL 470
                                         ALL 480
                                         ALL 490
                                         ALL 500
                                         ALL 510
                                         ALL 520
                                         ALL 530
                                         ALL 540
                                         ALL 550
                                         ALL 560
                                         ALL 570
                                         ALL 580
                                         ALL 590

```

```

IJK=LL1+J          ALL 600
IJ=LL2+J          ALL 610
SCRA(1)=1.0        ALL 620
SCRA(2)=RC(IJK-1) ALL 630
SCRA(3)=RC(IJ-1)  ALL 640
SCRA(4)=1.0        ALL 650
SCRA(5)=2.0        ALL 660
DO 190 I=1,N      ALL 670
IARGS(4)=IARGS(3)+I ALL 680
CALL ADRESS (4,LL22) ALL 690
GO TO (120,130,140,150,160,170), L2 ALL 700
C   L2 = 1      NLSUB      NORMALIZED LAGUERRE POLYNOMIALS ALL 710
C     RECURSION FORMULA  L(N+1) =(1.+2.*N-X)*L(N)-N**2 *L(N-1) ALL 720
C       L(0) = 1.          ALL 730
C       L(1) = -X+1.        ALL 740
C       L(2) = X**2 - 4.0*X +2.    ALL 750
C       L(3) = -X**3 + 9.0*X**2-18.0*X+6.    ALL 760
C
C   L(N)= EXP(X)*(DN/DXN(X**N*EXP(-X))) ALL 770
C
120  SCRA(4)=I          ALL 780
SCRA(6)=1.0+2.0*SCRA(4) ALL 790
SCRA(7)=SCRA(4)*SCRA(4) ALL 800
SCRA(8)=(SCRA(6)-SCRA(2))*SCRA(3)-SCRA(7)*SCRA(1) ALL 810
GO TO 180          ALL 820
C   L2 = 2      LSUB       LAGUERRE POLYNOMIALS ALL 830
C     RECURSION FORMULA  L(N+1)=(((2.*N+1)-X)*L(N)-N*L(N-1))/ ALL 840
C                           (N+1)          ALL 850
C       L(0) = 1.          ALL 860
C       L(1) = -X+1.        ALL 870
C       L(2) = .5 (XX*2 - 4.*X +2)    ALL 880
C       L(3) = (-X**3 + 9.*X**2 - 18.* X +6.)/6.    ALL 890
C
C   * SEE ABRAMOWITZ, M. AND STEGUN, I.A., HANDBOOK OF MATHEMATICAL ALL 900
C     FUNCTIONS, NATIONAL BUREAU OF STANDARDS APPLIED MATHEMATICSALL 910
C     SERIES 55, SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT    ALL 920
C     PRINTING OFFICE, WASHINGTON, D.C. 20402    ALL 930
C
C   * SEE HILSENRATH,ZIEGLER,MESSINA,WALSH,HERBOLD,, OMNITAB, NBS ALL 940
C     HANDBOOK 101 (MARCH 4, 1966) - FOR FORMULAE USED    ALL 950
C
130  SCRA(4)=I          ALL 960
SCRA(6)=SCRA(4)+1.0 ALL 970
SCRA(7)=SCRA(4)+SCRA(6) ALL 980
SCRA(8)=((SCRA(7)-SCRA(2))*SCRA(3)-SCRA(4)*SCRA(1))/SCRA(6) ALL 990
GO TO 180          ALL1000
C   L2 = 3      HSUB       HERMITE POLYNOMIALS ALL1010
C     RECURSION FORMULA  H(N+1) = 2.0*X*H(N)-2.0*N*H(N-1) ALL1020
C
C       H(0) = 1.          ALL1030
C       H(1) = 2.0*X        ALL1040
C       H(2) = 4.0*X**2-2.  ALL1050
C       H(3) = 8.0*X**3-12.*X ALL1060
C
140  SCRA(8)=2.0*(SCRA(2)*SCRA(3)-SCRA(4)*SCRA(1)) ALL1070
SCRA(4)=SCRA(4)+1.0 ALL1080
GO TO 180          ALL1090
C   L2 = 4      USUB       CHEBYSHEV POLYNOMIALS ALL1100
C
C     RECURSION FORMULA  U(N) = 2.0*X*U(N-1)-U(N-2)    ALL1110
C
C

```

```

C          U(0) = 1.                                ALL1190
C          U(1) = 2.0*X                                ALL1200
C          U(2) = 4.0*X**2-1.0                            ALL1210
C          U(3) = 8.0*X**3-4.0*X                            ALL1220
C
C 150    SCRA(8)=2.0*SCRA(2)*SCRA(3)-SCRA(1)          ALL1230
C          GO TO 180
C          L2      = 5      PSUB          LEGENDRE POLYNOMIALS ALL1240
C
C          RECUSION FORMULA  P(N+1) =X*P(N)+(N/N+1)*(X*P(N)-P(N-1)) ALL1250
C
C          P(0) = 1.                                ALL1260
C          P(1) = X.                                ALL1270
C          P(2) = (3./2.)*X**2-(1./2.)                ALL1280
C          P(3) = 2.5*X**3-1.5*X                  ALL1290
C
C 160    SCRA(6)=SCRA(4)/SCRA(5)                      ALL1300
C          SCRA(8)=(1.0+SCRA(6))*SCRA(2)*SCRA(3)-SCRA(6)*SCRA(1) ALL1310
C          SCRA(4)=SCRA(5)                            ALL1320
C          SCRA(5)=SCRA(5)+1.0                        ALL1330
C          GO TO 180
C          L2      = 6      TSUB          CHEBYSHEV POLYNOMIALS ALL1340
C
C          RECUSION FORMULA
C
C          T(0) = 1.                                ALL1400
C          T(1) = X.                                ALL1410
C          T(2) = 2.*X**2-1.                            ALL1420
C          T(3) = 4.*X**3-3.*X                  ALL1430
C
C 170    SCRA(8)=2.0*SCRA(2)*SCRA(3)-SCRA(1)          ALL1440
C 180    CONTINUE
C          LJMN=LL22+J                                ALL1450
C          RC(LJMN-1)=SCRA(8)                          ALL1460
C          SCRA(1)=SCRA(3)                            ALL1470
C          SCRA(3)=SCRA(8)                            ALL1480
C
C 190    CONTINUE
C 200    CONTINUE
C 210    RETURN
C          END

```

```

SUBROUTINE APRINT                               APR 10
C      VERSION 5.00     APRINT    5/15/70          APR 20
COMMON /BLOCRC/ NRC,RC(12600)                  APR 30
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NAPR 40
IARGS,VWXYZ(8),NERROR                         APR 50
DIMENSION ARGS(100)                           APR 60
EQUIVALENCE (ARGS(1),RC(12501))              APR 70
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG          APR 80
COMMON /ABCDEF/ L(48)                          APR 90
COMMON /HEADER/ NOCARD(80),ITLE(60,6),LNCNT,IPRINT,NPAGE,IPUNCH APR 100
COMMON /FMAT/ IFMTX(6),IOSWT,IFMTS(6),LHEAD(96) APR 110
COMMON /SCRAT/ NS,NS2,A(13500)                 APR 120
COMMON /KFMT/ KFMT(100)                        APR 130
DIMENSION IFRV(3)                            APR 140
DATA IFRV(1),IFRV(2),IFRV(3)/3H1X,,3H15,,3H2X,/ APR 150
C      L1 = 4 APRINT                           APR 160
C      L1 = 7 MPRINT                           APR 170
C      MPRINT PRINTS ROW/COL TITLE, APRINT DOES NOT. APR 180
C      ALL READABLE IF POSSIBLE, OTHERWISE ALL FLOATING. APR 190
C      WRITTEN BY DAVID HOBGEN, SEL, NBS.   8/18/69. APR 200
C      *****
C      6 FORMAT STATEMENTS FOLLOW WHICH MAY NEED MODIFY IF CHANGES MADE APR 220
C      *****
C      IF (NARGS.EQ.4) GO TO 30                APR 240
10     CALL ERROR (205)                         APR 250
20     RETURN                                APR 260
30     J=1                                    APR 270
      I=4                                    APR 280
      CALL CKIND (I)                          APR 290
      IF (I.NE.0) GO TO 10                     APR 300
      K=IARGS(1)                            APR 310
      CALL MTXCHK (J)                          APR 320
      IF (J.NE.0) GO TO 10                     APR 330
      IF (NERROR.NE.0) RETURN                 APR 340
C      CHECK TO SEE IF NPAGE=0. IF YES, BEGIN A NEW PAGE. CALL PAGE(0) APR 342
      IF (NPAGE.EQ.0) CALL PAGE(0)             APR 344
C      IF L2=1 IOSWT=0 USE READABLE FORMAT      APR 350
      IF (L2.EQ.1.AND.IOSWT.EQ.0) GO TO 190    APR 360
C      IF L2=1 IOSWT=1 USE FIXED OR FLOATING    APR 370
      IF (L2.EQ.1.AND.IOSWT.EQ.1) GO TO 60      APR 380
C      IF (L2.NE.1) USE SPECIFIED FORMAT        APR 390
C      FORMAT SHOULD SPECIFY FORMAT FOR ONLY ONE ROW APR 400
      CALL PREPAK (4,IND,L2,IND,KFMT)          APR 410
      IF (IND.NE.0) GO TO 50                     APR 420
      IA=IARGS(3)                            APR 440
      J1=IARGS(1)                            APR 450
      J2=J1+(IARGS(4)-1)*NROW                 APR 460
      DO 40 I=1,IA                           APR 470
      WRITE (IPRINT,KFMT) (RC(J),J=J1,J2,NROW)  APR 480
      J1=J1+1                                APR 490
      J2=J2+1                                APR 500
40     CONTINUE                                APR 510
      RETURN                                 APR 520
C      NO FORMAT IS FOUND SO USE READABLE FORMAT APR 530
50     CALL ERROR (222)                         APR 540
      GO TO 190                                APR 550
C      FIXED OR FLOATING FORMAT USED           APR 560
60     DO 70 I=1,100                           APR 570
70     KFMT(I)=L(45)                          APR 590

```

```

      IF (L1.EQ.7) GO TO 90                               APR 600
      DO 80 I=1,6                                         APR 610
80   KFMT(I)=IFMTX(I)                                   APR 620
      GO TO 110                                         APR 630
90   KFMT(1)=IFMTX(1)                                   APR 640
      KFMT(2)=IFRV(1)                                    APR 650
      KFMT(3)=IFRV(2)                                    APR 660
      KFMT(4)=IFRV(3)                                    APR 670
      DO 100 I=2,6                                     APR 680
100  KFMT(I+3)=IFMTX(I)                                 APR 690
110  KA=IARGS(1)                                       APR 700
      LL=IARGS(3)                                       APR 710
      IBB=IARGS(4)                                       APR 720
      IBBP=8                                           APR 730
      IF (L1.EQ.7) IBBP=7                                APR 740
      I1A=IARGS(2)                                       APR 750
120  IF (IBB.GT.IBBP) GO TO 130                           APR 760
      IB=IBB                                         APR 770
      IBB=0                                           APR 780
      GO TO 140                                         APR 790
130  IBB=IBB-IBBP                                       APR 800
      IB=IBBP                                         APR 810
140  KB=(IB-1)*NROW+KA                                 APR 820
      KBP=KB+NROW                                      APR 830
      I2A=I1A+IB-1                                     APR 840
      IF (L1.EQ.4) GO TO 150                                APR 850
      WRITE (IPRINT,280) L(28),L(25),L(33),L(37),L(13),L(25),L(22),(JJ,JAPR 860
1J=I1A,I2A)
      MRV=K                                           APR 870
150  DO 180 M=1,LL                                     APR 880
      IF (L1.EQ.4) GO TO 160                                APR 890
      WRITE (IPRINT,KFMT) MRV,(RC(K),K=KA,KB,NROW)        APR 900
      MRV=MRV+1                                         APR 910
      GO TO 170                                         APR 920
160  WRITE (IPRINT,KFMT) (RC(K),K=KA,KB,NROW)          APR 930
170  KA=KA+1                                         APR 940
      KB=KB+1                                         APR 950
180  CONTINUE                                         APR 960
      IF (IBB.EQ.0) RETURN                                APR 970
      WRITE (IPRINT,330)                                   APR 980
C     PRINT NEXT SET OF COLUMNS                         APR1000
      KA=KBP                                         APR1010
      I1A=I2A+1                                       APR1020
      GO TO 120                                         APR1030
C     THE NEXT CARD MUST BE CHANGED IF WIDTH OF COLUMN CHANGED APR1040
C     THE CARD AFTER IT MUST BE CHANGED IF NUMBER OF COLUMNS CHANGES APR1050
C     2 CALLS TO RFORMAT LATER NEED TO BE CHANGED IF NO. OF SD NOT 8. APR1060
C     NWMX IS DETERMINED BY 120/8-2 WHERE 120 IS THE NUMBER OF APR1070
C     CHARACTERS PER PRINTED LINE                      APR1080
190  NWMX=13                                         APR1100
      MCOL=8-L1/7                                       APR1110
      NSTART=IARGS(1)                                    APR1120
      KSTART=K-1                                       APR1130
      KR=IARGS(3)                                       APR1140
      KC=IARGS(4)                                       APR1150
      K1=1                                           APR1160
      K2=NSTART                                      APR1170
      DO 210 I=1,KC                                    APR1180
      DO 200 J=1,KR                                    APR1190

```

```

A(K1)=RC(K2)                               APR1200
K1=K1+1                                     APR1210
200 K2=K2+1                                     APR1220
210 K2=K2+NROW-KR                           APR1230
      KSIZE=KR*KC                           APR1240
      CALL RFORMAT (A(1),KSIZE,8,NWIDTH,NDECS,NWMX+1,A(1),A(1),0,0) APR1250
C      MINIMUM OF TWO BLANK SPACES ON LEFT   APR1260
      NBLANK=NWMX+2-NWIDTH                   APR1270
      I1=1                                     APR1280
      I1A=IARGS(2)                           APR1290
      K1=NSTART-1                           APR1300
C      LOOP ON BLOCKS                      APR1310
220 I2=I1+MCOL-1                           APR1320
      I2=MIN0(I2,KC)                         APR1330
      I2A=IARGS(2)+I2-1                      APR1340
      K2=K1+(I2-I1)*NROW                     APR1350
      K4=K2                                     APR1360
      IF (L1.EQ.4) GO TO 230                 APR1370
      WRITE (IPRINT,280) L(28),L(25),L(33),L(37),L(13),L(25),L(22),(JJ,JAPR1380
      IJ=I1A,I2A)                           APR1390
C      LOOP ON ROWS                        APR1400
230 DO 270 JJ=1,KR                           APR1410
      K1=K1+1                               APR1420
      K2=K2+1                               APR1430
      JJJ=KSTART+JJ                          APR1440
      IF (NWIDTH.LE.NWMX.OR.L1.NE.7) GO TO 240 APR1450
C      WRITE FLOATING IF MPRINT (L1=7) AND NWIDTHT GT NWMAX APR1460
      WRITE (IPRINT,300) JJJ,(RC(K3),K3=K1,K2,NROW) APR1470
      GO TO 270                           APR1480
240 LL=1                                     APR1490
      K=K1                                     APR1500
C      LOOP ON COLUMNS                    APR1510
      DO 250 II=I1,I2                      APR1520
      CALL RFORMAT (A,1,8,NWIDTH,NDECS,0,RC(K),A(LL),NBLANK,0) APR1530
      K=K+NROW                                APR1540
250 LL=LL+NWMX+2                           APR1550
      NL=LL-1                               APR1560
      IF (L1.EQ.7) GO TO 260                 APR1570
      WRITE (IPRINT,310) (A(LL),LL=2,NL)       APR1580
      GO TO 270                           APR1590
260 WRITE (IPRINT,290) JJJ,(A(LL),LL=2,NL)   APR1600
270 CONTINUE                                APR1610
      K1=K4+NROW                           APR1620
      I1=I1+MCOL                           APR1630
      I1A=I1A+MCOL                         APR1640
      IF (I2.GE.KC) GO TO 20                APR1650
C      LOOP ON BLOCKS                      APR1660
C      PUT IN BLANK LINE BETWEEN BLOCKS    APR1670
      WRITE (IPRINT,320)                   APR1680
      GO TO 220                           APR1690
C
280 FORMAT (1X,7A1,7(6X,I5,4X))           APR1700
290 FORMAT (1X,I5,2X,112A1)               APR1720
300 FORMAT (1X,I5,2X,1P7E15.6)           APR1730
310 FORMAT (1X,119A1)                     APR1740
320 FORMAT (1H )                         APR1750
330 FORMAT (1X)                          APR1760
      END                                     APR1770

```

```

SUBROUTINE ARITH                                ARI 10
C   VERSION 5.00      ARITH    5/15/70          ARI 20
COMMON /BLOCRC/ NRC,RC(12600)                  ARI 30
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NARI 40
IARGS,VWXYZ(8),NERROR                         ARI 50
DIMENSION ARGS(100)                           ARI 60
EQUIVALENCE (ARGS(1),RC(12501))              ARI 70
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG          ARI 80
DIMENSION II(5),KK(5)                          ARI 90
EQUIVALENCE (II(1),I1),(II(2),I2),(II(3),I3),
1 (II(4),I4),(II(5),I5)                      ARI 100
ARI 105
C
C   THIS SUBROUTINE PERFORMS ADD,SUB,MULT,DIV AND RAISE FOR ARI 110
C   THREE, FOUR AND FIVE ARGUMENTS               ARI 120
C   L2=1  ADD                                    ARI 130
C   L2=2  SUBTRACT                             ARI 140
C   L2=3  MULTIPLY                            ARI 150
C   L2=4  DIVIDE                               ARI 160
C   L2=5  RAISE                                 ARI 170
C   L2=6  ACCURATE DIGITS                     ARI 180
C
C
IF(NARGS.LT.3.OR.NARGS.GT.5) CALL ERROR(10)      ARI 200
IF (NARGS.EQ.4)CALL ERROR (29)                   ARI 215
IF(KIND(NARGS).NE.0) CALL ERROR (20)             ARI 220
IF (L2.NE.6) GO TO 5                            ARI 222
IF (NARGS.EQ.3) GO TO 5                          ARI 224
CALL ERROR (212)                                ARI 226
NARGS=3                                         ARI 228
5   DO 30  I=1,NARGS                           ARI 230
KK(I)=1                                         ARI 240
CALL ADRESS (I,II(I))                         ARI 250
IF(II(I)) 20,10,30                            ARI 260
10  CALL ERROR(11)                            ARI 270
RETURN                                         ARI 280
20  KK(I)=0                                     ARI 290
II(I)=-II(I)                                  ARI 300
30  CONTINUE                                    ARI 310
IF(NRMAX.LE.0) CALL ERROR(9)                   ARI 320
IF(NERROR.NE.0) RETURN                         ARI 330
JJ=II(NARGS)+NRMAX-1                          ARI 380
IF(NARGS.NE.3) GO TO 120                       ARI 390
DO 110  I=I3,JJ                                ARI 400
GO TO (50,60,70,80,90,95),L2                  ARI 410
50  RC(I)=RC(I1)+RC(I2)                         ARI 420
GO TO 100                                       ARI 430
60  RC(I)=RC(I2)-RC(I1)                         ARI 440
GO TO 100                                       ARI 450
70  RC(I)=RC(I1)*RC(I2)                         ARI 460
GO TO 100                                       ARI 470
80  IF(RC(I2).NE.0.0) GO TO 85                 ARI 480
RC(I)=0.0                                      ARI 490
CALL ERROR (106)                                ARI 500
GO TO 100                                       ARI 510
85  RC(I)=RC(I1)/RC(I2)                         ARI 520
GO TO 100                                       ARI 530
90  RC(I)=FEXP2(RC(I1),RC(I2))                ARI 540
GO TO 100                                       ARI 545
95  CALL ACCDIG(RC(I1),RC(I2),RC(I),1)          ARI 547
100 I1=I1+KK(1)                                ARI 550

```

110	I2=I2+KK(2)	ARI 560
	RETURN	ARI 570
120	IF(NARGS.EQ.5) GO TO 130	ARI 580
	I5=I4	ARI 590
	KK(5)=KK(4)	ARI 600
130	DO 200 I=I5,JJ	ARI 610
	GO TO (140,150,160,170,180),L2	ARI 620
140	X=RC(I1)+RC(I2)	ARI 630
	GO TO 190	ARI 640
150	X=RC(I2)-RC(I1)	ARI 650
	GO TO 190	ARI 660
160	X=RC(I1)*RC(I2)	ARI 670
	GO TO 190	ARI 680
170	IF(RC(I2).NE.0.0) GO TO 175	ARI 690
	X=0.0	ARI 700
	CALL ERROR (106)	ARI 710
	GO TO 190	ARI 720
175	X=RC(I1)/RC(I2)	ARI 730
	GO TO 190	ARI 740
180	X=FEXP2(RC(I1),RC(I2))	ARI 750
190	RC(I)=X*RC(I3)+RC(I4)	ARI 760
	I1=I1+KK(1)	ARI 765
	I2=I2+KK(2)	ARI 770
	I3=I3+KK(3)	ARI 775
	I4=I4+KK(4)	ARI 780
200	CONTINUE	ARI 785
	RETURN	ARI 790
	END	ARI 800

```

SUBROUTINE ARYVEC
C VERSION 5.00      ARYVEC      5/15/70          ARY  10
C SUBROUTINE ARYVEC R.VARNER   9/27/67          ARY  20
C *
C SUBROUTINE TO MULTIPLY MATRIX TIME VECTOR          ARY  30
C           OR VECTOR TRANSPOSE TIME MATRIX          ARY  40
C L2=1 MULTIPLY MATRIX TIME VECTOR          ARY  50
C           GENERAL FORM OF COMMAND          ARY  60
C           M(AV) A (,) N,K VECTOR IN COL I STORE IN COLUMN J ARY  70
C           M(AV) A (,) N,K VECTOR IN COL I STORE IN ROW K COL J ARY  80
C           N AND K MUST BE SPECIFIED          ARY  90
C L2=2 MULTIPLY VECTOR TRANSPOSE TIMES MATRIX          ARY 100
C           GENERAL FORM OF COMMAND          ARY 110
C           M(V,A) A (,) N,K VECTOR IN COL I STORE IN ROW J ARY 120
C           M(V,A) A (,) N,K VECTOR IN COL I STORE IN ROW K COL J ARY 130
C           N AND K MUST BE SPECIFIED          ARY 140
C           IF ONLY ROW IS GIVEN FOR STORAGE COL 1 IS ASSUMED ARY 150
C *
C           N AND K MUST BE SPECIFIED          ARY 160
C           IF ONLY ROW IS GIVEN FOR STORAGE COL 1 IS ASSUMED ARY 170
C           *          ARY 180
C COMMON /BLOCRC/ NRC,RC(12600)          ARY 200
C COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NARY 210
IARGS,VWXYZ(8),NERROR          ARY 220
DIMENSION ARGS(100)          ARY 230
EQUIVALENCE (ARGS(1),RC(12501))          ARY 240
COMMON /SCRAT/ NS,NS2,A(13500)          ARY 250
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG          ARY 260
DIMENSION X(1)          ARY 270
DOUBLE PRECISION X,SUM          ARY 280
EQUIVALENCE (X,A)          ARY 290
C *
C CHECK FOR CORRECT NUMBER OF ARGUMENTS          ARY 310
C *
C IF (NARGS.NE.6.AND.NARGS.NE.7) CALL ERROR (10)          ARY 320
C *
C CHECK TO SEE IF ALL ARGUMENTS ARE INTEGERS          ARY 330
C *
C J=NARGS          ARY 340
C CALL CKIND (J)          ARY 350
C IF (J.NE.0) CALL ERROR (3)          ARY 360
C *
C CHECK TO SEE IF DIMENSIONS ARE OUT OF RANGE          ARY 370
C *
C GO TO (10,20), L2          ARY 380
10 IADD=IARGS(4)          ARY 390
IADD2=IARGS(3)          ARY 400
ICOMP=NROW          ARY 410
GO TO 30          ARY 420
C *
C GO TO (10,20), L2          ARY 430
20 IADD=IARGS(3)          ARY 440
IADD2=IADD          ARY 450
ICOMP=NCOL          ARY 460
GO TO 30          ARY 470
C *
C IF (NARGS.NE.6) GO TO 30          ARY 480
C IF (IARGS(6).GT.NROW.OR.IARGS(4).GT.NCOL) CALL ERROR (17)          ARY 490
C *
C COMPUTE ADDRESSES OF COLUMNS          ARY 500
C *
C 30 IARGS(10)=IARGS(NARGS)          ARY 510
IARGS(8)=1          ARY 520
GO TO (70,40), L2          ARY 530
40 IF (NARGS.EQ.7) GO TO 50          ARY 540
J=2          ARY 550

```

```

IROWSV=IARGS(6)                               ARY 620
GO TO 60                                     ARY 630
50   IARGS(12)=IARGS(4)                         ARY 640
     IARGS(11)=1                                ARY 650
     IARGS(9)=IARGS(6)                            ARY 660
     J=3                                         ARY 670
60   IARGS(7)=IARGS(3)                            ARY 680
     GO TO 90                                    ARY 690
70   J=3                                         ARY 700
     IARGS(12)=1                                ARY 710
     IARGS(11)=IARGS(3)                          ARY 720
     IARGS(7)=IARGS(4)                            ARY 730
     IF (NARGS.EQ.6) GO TO 80                    ARY 740
     IARGS(9)=IARGS(6)                            ARY 750
     GO TO 90                                    ARY 760
80   IARGS(9)=1                                  ARY 770
90   IARGS(6)=IARGS(5)                            ARY 780
     IARGS(5)=1                                ARY 790
     CALL MTXCHK (J)                           ARY 800
     IF (J-1) 120,100,110                      ARY 810
100  CALL ERROR (3)                             ARY 820
     RETURN                                      ARY 830
110  CALL ERROR (17)                            ARY 840
     RETURN                                      ARY 850
C   *
C   CHECK FOR PREVIOUS ERRORS                  ARY 860
C   *
120  IF (NERROR.NE.0) RETURN                   ARY 890
     GO TO (130,140), L2                      ARY 900
130  ICS=IARGS(9)                             ARY 910
     IAP=IARGS(1)                            ARY 920
     IP=IARGS(3)                             ARY 930
     JP=IARGS(4)                            ARY 940
     IAD1=NROW                                ARY 950
     IAD2=1                                 ARY 960
     IBP=IARGS(5)                            ARY 970
     GO TO 170                                ARY 980
140  IBP=IARGS(1)                             ARY 990
     IAP=IARGS(5)                            ARY1000
     IP=IARGS(4)                            ARY1010
     IF (NARGS.EQ.7) GO TO 150                ARY1020
     JP=IARGS(3)                             ARY1030
     ICS=IROWSV
     GO TO 160                                ARY1040
150  JP=IARGS(3)                             ARY1060
     ICS=IARGS(9)                            ARY1070
160  IAD1=1                                 ARY1080
     IAD2=NROW                                ARY1090
170  IC=1                                   ARY1100
     DO 210 I=1,IP                           ARY1110
     IA=IAP                                 ARY1120
     IB=IBP                                 ARY1130
     IS=NS2                                 ARY1140
     DO 180 J=1,JP                           ARY1150
     X(IS)=RC(IA)*RC(IB)                     ARY1160
     IS=IS-1                                ARY1170
     IA=IA+IAD1                            ARY1180
180  IB=IB+1                                ARY1190
     CALL SORTSM (JP,SUM)                     ARY1200

```

A(IC)=SUM	ARY1210
IC=IC+1	ARY1220
GO TO (190,200), L2	ARY1230
190 IAP=IAP+1	ARY1240
GO TO 210	ARY1250
200 IBP=IBP+NROW	ARY1260
210 CONTINUE	ARY1270
C *	ARY1280
C STORE RESULTS IN WORKSHEET	ARY1290
C *	ARY1300
IS=1	ARY1310
DO 220 I=1,IP	ARY1320
RC(ICS)=A(IS)	ARY1330
IS=IS+1	ARY1340
ICS=ICS+IAD2	ARY1350
220 CONTINUE	ARY1360
RETURN	ARY1370
END	ARY1380

```

SUBROUTINE ASTER                               AST 10
C      VERSION 5.00    ASTER      5/15/70          AST 20
COMMON /BLOCKA/ MODE,M,KARD(83),KARG,ARG,ARG2,NEWCD(80),KRDEND   AST 30
DIMENSION NAM(2)                                AST 40
C
C      ASTERISKS HAVE BEEN FOUND, LOOK FOR A SPECIAL FORM OF ARGUMENT.  AST 50
C
C      FORMS CAN BE..                                         AST 70
C
C      *PHYSCON*     A PHYSICAL CONSTANT NAME, FL.PT.          AST 80
C      **VARCON**    A -VARIABLE- CONSTANT TO BE USED AS AN INTEGER (TRUN)AST 90
C      *VARCON*     A -VARIABLE- CONSTANT TO BE USED AS A FL.PT. NUMBER AST 100
C      **ROW,COLUMN** A WORKSHEET ENTRY TO BE TRUNCATED AND USED AS INTAST 110
C      *ROW,COLUMN*   A WORKSHEET ENTRY TO BE USED AS FLOATING POINT AST 120
C
C      NONBLA IS A FUNCTION WHICH RETURNS THE NEXT NON-BLANK CHARACTER  AST 130
C      IN THE CARD AND ALSO POINTS M AT IT                         AST 140
C
C      KARG = 1, SINGLE *.  KARG = 0, DOUBLE *.                  AST 150
C
C      L=KARG                                         AST 160
C      K=NONBLA(M)                                     AST 170
C      IF (K.NE.40) GO TO 20                          AST 180
C
C      A LONG LINE OF ASTERISKS FOUND, SKIP OVER THEM AND IGNORE        AST 190
C
C      KARG=7                                         AST 200
C
10     M=M+1                                         AST 210
IF (KARD(M)-40) 100,10,100                      AST 220
20     IF (K.GE.36) GO TO 60                         AST 230
IF (K.GE.10) GO TO 40                           AST 240
C
C      NUMBER IS FIRST NON-BLANK CHARACTER, SET N = COMMA           AST 250
C
C      N=43                                           AST 260
C
30     CALL AARGS                                      AST 270
IF (KARG.NE.0) GO TO 60                         AST 280
IF (NONBLA(M).EQ.N) IF (N-40) 35,37,35          AST 290
GO TO 60                                         AST 300
35     IF (NONBLA(M+1).GE.10) GO TO 60             AST 310
C
C      SET N = ASTERISK                                AST 320
C
C      N=40                                           AST 330
C
30     T=ARG                                         AST 340
GO TO 30                                         AST 350
37     ARG2=ARG                                      AST 360
ARG=T                                           AST 370
KARG=5                                         AST 380
GO TO 90                                         AST 390
C
C      LETTER FOUND FIRST                            AST 400
C
C      SET N = ASTERISK                                AST 410
C
C      N=40                                           AST 420
C
30     T=ARG                                         AST 430
GO TO 30                                         AST 440
37     ARG2=ARG                                      AST 450
ARG=T                                           AST 460
KARG=5                                         AST 470
GO TO 90                                         AST 480
C
C      PHYSICAL CONSTANT FOUND, SET KARG = 1          AST 490
C
C      LETTER FOUND FIRST                            AST 500
C
C      CALL NNAME (NAM(1))                           AST 510
C      CALL PHYCON (NAM(1))                          AST 520
IF (ARG.EQ.0.) GO TO 50                         AST 530
C
C      PHYSICAL CONSTANT FOUND, SET KARG = 1          AST 540
C
C      LETTER FOUND FIRST                            AST 550
C
C      CALL NNAME (NAM(1))                           AST 560
C      CALL PHYCON (NAM(1))                          AST 570
IF (ARG.EQ.0.) GO TO 50                         AST 580

```

```

KARG=1          AST 590
IF (L) 60,60,80 AST 600
C               AST 610
NAME NOT IN PHYSICAL CONSTANT LIST, TRY VARIABLE LIST AST 620
C               AST 630
50  CALL VARCON (NAM(1)) AST 640
    IF (ARG.NE.0.) GO TO 70 AST 650
60  KARG=1          AST 660
    RETURN          AST 670
70  KARG=3          AST 680
80  IF (NONBLA(M).NE.40) GO TO 60 AST 690
90  M=M+1          AST 700
C               AST 710
CHECK THAT THE NUMBER OF ASTERISKS AT THE END OF THE EXPRESSION AST 720
C               IS THE SAME AS AT THE BEGINNING. L=0 MEANS 2, L=1 MEANS 1 AST 730
C               AST 740
95  IF (L.NE.0) IF (KARD(M)-40) 95,60,95 AST 745
    IF (KARD(M).NE.40.OR.KARD(M+1).EQ.40) GO TO 60 AST 750
95  M=M+1          AST 760
    KARG=KARG+L          AST 770
100 RETURN          AST 780
    END          AST 800

```

```

SUBROUTINE BEGIN BEG 10
C VERSION 5.00 BEGIN 5/15/70 BEG 20
COMMON /BLOCKA/ MODE,M,KARD(83),KARG,ARG,ARG2,NEWCD(80),KRDEND BEG 30
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG BEG 40
COMMON /BLOCKB/ NSTMT,NSTMTH,NCOM,LCOM,IOVFL,COM(2000) BEG 50
COMMON /BLOCRC/ NRC,RC(12600) BEG 60
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NBEG 70
IARGS,VWXYZ(8),NERROR BEG 80
DIMENSION ARGS(100) BEG 90
EQUIVALENCE (ARGS(1),RC(12501)) BEG 100
C BEG 110
C THIS SUBROUTINE CONTAINS THE CODING FOR BEGIN AND SCAN BEG 120
C BEG 140
C IF (L2.EQ.1) GO TO 20 BEG 150
C BEG 160
C SCAN (CARD UP TO AND INCLUDING CARD COLUMN ++ ) BEG 170
IF (NARGS.GT.1) CALL ERROR (221) BEG 180
IF (NARGS.GE.1.AND.KIND(1).EQ.0.AND.IARGS(1).GE.6.AND.IARGS(1).LE.190
180) GO TO 10 BEG 200
K=205 BEG 210
GO TO 30 BEG 220
10 KRDEND=IARGS(1) BEG 230
GO TO 40 BEG 240
C BEG 250
C BEGIN STORING INSTRUCTIONS AT NUMBER ++ BEG 260
C IF NO NUMBER IS GIVEN, 1 IS ASSUMED. BEG 270
C BEG 280
20 IF (MODE.EQ.1) GO TO 50 BEG 290
K=5 BEG 300
30 CALL ERROR (K) BEG 310
40 RETURN BEG 320
50 IF (NARGS-1) 70,90,60 BEG 330
60 K=10 BEG 340
GO TO 30 BEG 350
70 NSTMT=0 BEG 360
80 MODE=3 BEG 370
GO TO 40 BEG 380
90 IF (KIND(1).EQ.0) GO TO 100 BEG 390
K=20 BEG 400
GO TO 30 BEG 410
100 IF (IARGS(1).GT.0.AND.IARGS(1).LT.1000) GO TO 110 BEG 420
K=7 BEG 430
GO TO 30 BEG 440
110 NSTMT=10*(IARGS(1)-1) BEG 450
GO TO 80 BEG 460
END BEG 470

```

```

SUBROUTINE BEJN (IST,R,Z)          BEJ   10
C VERSION 5.00      BEJN      5/15/70    BEJ   20
C IF IST=0 ENTRY IS FOR BEJN     BEJ   30
C IF IST=1 ENTRY IS FOR BEIN    BEJ   40
C DIMENSION R(1)                BEJ   50
C DOUBLE PRECISION X,R,Z,A,B,C,D,E,F,G,P,Q,Y  BEJ   60
C Y=1.D0                         BEJ   70
C IF (IST.NE.0) Y=-1.D0           BEJ   80
C X=Z                           BEJ   90
C DO 10 N=1,100                 BEJ  100
10  R(N)=0.0                      BEJ  110
C LA=0                          BEJ  120
C IF (X.LE.60.) GO TO 30         BEJ  130
C LA=1                          BEJ  140
C IF (X.LE.100.) GO TO 20        BEJ  150
C CALL ERROR (225)              BEJ  160
C GO TO 130                     BEJ  170
20  X=X/2.D0                      BEJ  180
30  A=X/2.D0                      BEJ  190
C IF (X.GT.15.) GO TO 100        BEJ  200
C B=1.D0                         BEJ  210
C C=1.D0                         BEJ  220
C DO 40 N=1,100                 BEJ  230
C J=N                           BEJ  240
C B=B*A/C                       BEJ  250
C C=C+1.D0                      BEJ  260
C IF (B.LT..5D-30) GO TO 50      BEJ  270
40  CONTINUE                      BEJ  280
50  D=B*A/C                      BEJ  290
C A=A**2                         BEJ  300
C K=X+6.D0                      BEJ  310
C E=K                           BEJ  320
C F=K+J                         BEJ  330
C G=F+1.D0                      BEJ  340
C P=1.D0                         BEJ  350
C Q=1.D0                         BEJ  360
C DO 60 N=1,K                   BEJ  370
C P=1.D0-P*A/(E*F)*Y            BEJ  380
C Q=1.D0-Q*A/(E*G)*Y            BEJ  390
C E=E-1.D0                      BEJ  400
C F=F-1.D0                      BEJ  410
60  G=G-1.D0                      BEJ  420
C R(J+1)=B*P                     BEJ  430
C R(J+2)=D*Q                     BEJ  440
70  DO 80 N=1,J                   BEJ  450
C K=J-N+1                        BEJ  460
C A=K                           BEJ  470
80  R(K)=2.D0*A*R(K+1)/X-R(K+2)*Y  BEJ  480
C IF (LA.EQ.0) GO TO 130          BEJ  490
C LA=LA-1                        BEJ  500
C A=R(1)*R(100)                  BEJ  510
C B=.0D0                         BEJ  520
C DO 90 N=1,99                    BEJ  530
C K=100-N                        BEJ  540
C A=A+R(N+1)*R(K)                BEJ  550
90  B=B+R(N)*R(K)                BEJ  560
C J=98                           BEJ  570
C R(100)=A                        BEJ  580
C R(99)=B                        BEJ  590

```

X=Z	BEJ 600
GO TO 70	BEJ 610
K=1.5*X	BEJ 620
B=1.D0	BEJ 630
C=K	BEJ 640
DO 110 N=1,K	BEJ 650
B=A*B/C	BEJ 660
110 C=C-1.D0	BEJ 670
P=2.D-9	BEJ 680
IF (LA.EQ.1) P=5.D-20	BEJ 690
C=K+1	BEJ 700
DO 120 N=1,30	BEJ 710
J=K+N	BEJ 720
B=B*A/C	BEJ 730
C=C+1.D0	BEJ 740
IF (B.LT.P) GO TO 50	BEJ 750
IF (J.EQ.98) GO TO 50	BEJ 760
120 CONTINUE	BEJ 770
GO TO 50	BEJ 780
130 RETURN	BEJ 790
END	BEJ 800

```

SUBROUTINE BESSEL                                BES 10
C      VERSION 5.00      BESSEL      5/15/70      BES 20
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NBES 30
IARGS,VWXYZ(8),NERROR                           BES 40
COMMON /BLOCRC/ NRC,RC(12600)                   BES 50
EQUIVALENCE (ARGS(1),RC(12501))                BES 60
DIMENSION ARGS(100)                            BES 70
COMMON /SCRAT/ NS,NS2,A(13500)                  BES 80
DOUBLE PRECISION DBEJ,X,Y,E,P,Q,S,T,BINTJO,COMELL,Z,DXEX,XEX BES 90
DOUBLE PRECISION FDCOS,FDEXP                   BES 100
DOUBLE PRECISION AA(1000),B(1000),W(100)        BES 110
EQUIVALENCE (A(1),AA), (A(2001),B), (A(4001),W) BES 120
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG           BES 130
DIMENSION R(1), IA(1), KI(1)                   BES 140
EQUIVALENCE (R,RC), (IA,IARGS), (KI,KIND), (NR,NRMAX) BES 150
COMMON /ABEKI/ X,Y,P,Q,S,T                      BES 160
DOUBLE PRECISION DSNCOS,DXEXP                  BES 170
COMMON /CONSLB/ XTRIG,XEXP                     BES 180
COMMON /DCONLB/ DSNCOS,DXEXP                  BES 190
C
C      WJG PATCH      )()()()()()()()()()()()()()()()()()()()()()()()() BES 200
C
C      XEX=XEXP-3.0                         BES 210
C      DXEX=DXEXP-4.0D0                        BES 220
C      IF (NARGS.GE.2) GO TO 10                 BES 225
C      CALL ERROR (10)                         BES 227
C      RETURN                                  BES 230
C
C      END PATCH      )()()()()()()()()()()()()()()()()()()()()()()() BES 240
C
C      IF (L2.GT.28) GO TO 250                 BES 250
C      N=0                                     BES 260
C      L=L2/2                                 BES 270
C      L=2*L                                 BES 280
C      IF (L.EQ.L2) N=1                         BES 290
C      IF (L2.GT.12) GO TO 110                 BES 300
C      IF (NARGS.GT.2) CALL ERROR (10)          BES 310
C      CALL ADRESS (NARGS,J)                   BES 320
C      IF (J.LE.0) CALL ERROR (11)              BES 330
C      LT=1                                    BES 340
C      IF (KI(1).EQ.1) GO TO 20                 BES 350
C      CALL ADRESS (1,JA)                      BES 360
C      IF (JA.LE.0) CALL ERROR (11)              BES 370
C      LT=2                                    BES 380
C      M=1                                     BES 390
C      IF (NERROR.NE.0) RETURN                 BES 400
C      IF (L2.GT.2) M=5                         BES 410
C      IF (L2.GT.4) M=3                         BES 420
C      IF (L2.GT.6) M=7                         BES 430
C      IF (L2.GT.8) M=3                         BES 440
C      IF (L2.GT.10) M=7                        BES 450
C      L=0                                     BES 460
C      IF (L2.GT.4) L=1                         BES 470
C      IF (L2.GT.8) L=2                         BES 480
C      IF (LT.EQ.1) GO TO 70                   BES 490
C      DO 50 I=1, NR                          BES 500
C      X=R(JA)                                BES 510
C      JA=JA+1                               BES 520
C      Y=1.D0                                 BES 530
C

```

```

IF (L.EQ.0) GO TO 33                                BES 580
IF (L.EQ.2) GO TO 30                                BES 590
IF(DABS(X).LT.XEX) GO TO 33                         BES 600
C IF 1X1 IS GREATER THAN XEXP AND LESS THAN DXEXP THE RESULTS BES 610
C WILL BE SCALED BY EXP (X) OR EXP(-X).             BES 620
C IF 1X1 IS GREATER THAN DXEXP THE SUBROUTINE DBEJ DOES THE SCALINGBES 630
C AND A MESSAGE IS PRINTED.                         BES 640
C THIS APPLIES TO THE FOLLOWING COMMANDS BIZERO, BIONE, BKZERO, BES 650
C BKONE                                              BES 660
CALL ERROR (105)                                    BES 670
30 IF (DABS(X).GT.DXEX) GO TO 33                  BES 680
Y=FDEXP(X)                                         BES 690
IF (M.EQ.3) Y=1.D0/Y                             BES 700
33 IF (M.EQ.5.OR.M.EQ.7) IF(X) 35,35,40          BES 705
GO TO 40                                           BES 710
35 R(J)=0.                                         BES 720
CALL ERROR (101)                                    BES 730
GO TO 50                                           BES 740
C L2=1,M=0,LT=2,L=0 :BJZERO OF ++ STORE IN ++  BES 750
C L2=2,M=1,N=1,LT=2,L=0 :BJONE OF ++ STORE IN ++ BES 760
C L2=3,M=5,N=0,LT=2,L=0 :BYZERO OF ++ STORE IN ++ BES 770
C L2=4,M=5,N=1,LT=2,L=0 :BYONE OF ++ STORE IN ++  BES 780
C L2=5,M=3,N=0,LT=2,L=1 :BIZERO OF ++ STORE IN ++ BES 790
C L2=6,M=3,N=1,LT=2,L=1 :BIONE OF ++ STORE IN ++  BES 800
C L2=7,M=7,N=0,LT=2,L=1 :BKZERO OF ++ SOTRE IN ++ BES 810
C L2=8,M=7,N=1,LT=2,L=1 :BKONE OF ++ STORE IN ++  BES 820
C L2=9,M=3,N=0,LT=2,L=2 :EXJZERO OF ++ STORE IN ++ BES 830
C L2=10,M=3,N=1,LT=2,L=2 :EXIONE OF ++ STORE IN ++ BES 840
C L2=11,M=7,N=0,LT=2,L=2 :EXKZERO OF ++ STORE IN ++ BES 850
C L2=12,M=7,N=1,LT=2,L=2 :EXKONE OF ++ STORE IN ++ BES 860
40 R(J)=Y*DBEJ(X,N,M)                            BES 870
50 J=J+1                                         BES 880
60 RETURN                                         BES 890
70 X=ARGS(1)                                       BES 900
Y=1.D0                                         BES 910
IF (L.EQ.0) GO TO 90                           BES 920
IF (L.EQ.2) GO TO 80                           BES 930
IF (DABS(X).LT.XEX) GO TO 90                  BES 940
C SEE COMMENTS ABOVE ON BOUNDS OF X BEFORE DBEJ IS CALLED BES 950
CALL ERROR (105)                                 BES 960
80 IF (DABS(X).GT.DXEX) GO TO 90              BES 970
Y=FDEXP(X)                                         BES 980
IF (M.EQ.3) Y=1.D0/Y                           BES 990
C L2=1,M=1,N=0,LT=1,L=0 :BJZERO OF ** STORE IN ++ BES1000
C L2=2,M=1,N=1,LT=1,L=0 :BJONE OF ** STORE IN ++ BES1010
C L2=3,M=5,N=0,LT=1,L=0 :BYZERO OF ** STORE IN ++ BES1020
C L2=4,M=5,N=1,LT=1,L=0 :BYONE OF ** STORE IN ++ BES1030
C L2=5,M=3,N=0,LT=1,L=1 :BIZERO OF ** STORE IN ++ BES1040
C L2=6,M=3,N=1,LT=1,L=1 :BIONE OF ** SOTRE IN ++ BES1050
C L2=6,M=3,N=1,LT=1,L=1 :BIONE OF ** STORE IN ++ BES1060
C L2=7,M=7,N=0,LT=1,L=1 :BKZERO OF ** SOTRE IN ++ BES1070
C L2=8,M=7,N=1,LT=1,L=1 :BKONE OF ** STORE IN ++ BES1080
C L2=9,M=3,N=0,LT=1,L=2 :EXIZERO OF ** STORE IN ++ BES1090
C L2=10,M=3,N=1,LT=1,L=2 :EXIONE OF ** STORE IN ++ BES1100
C L2=11,M=7,N=0,LT=1,L=2 :EXKZERO OF ** STORE IN ++ BES1110
C L2=12,M=7,N=1,LT=1,L=2 :EXKONE OF ** STORE IN ++ BES1120
90 X=Y*DBEJ(X,N,M)                            BES1130
DO 100 I=1,NR                                  BES1140
R(J)=X                                         BES1150

```

```

100 J=J+1                                BES1160
     GO TO 60                             BES1170
110 IF (L2.GT.20) GO TO 210               BES1180
     IF (NARGS.GT.3) CALL ERROR (10)       BES1190
     M=1                                  BES1200
     IF (L2.GT.14) M=2                   BES1210
     IF (L2.GT.16) M=1                   BES1220
     IF (L2.GT.18) M=2                   BES1230
     L=0                                  BES1240
     IF (L2.GT.16) L=1                  BES1250
     Y=.785398163397D0                 BES1260
     LV=0                                 BES1270
     JX=0                                 BES1280
120 CALL ADRESS (NARGS,J2)               BES1290
     IF (J2.LE.0) CALL ERROR (11)         BES1300
     CALL ADRESS (NARGS-1,J1)            BES1310
     IF (J1.LE.0) CALL ERROR (11)         BES1320
     LT=0                                 BES1330
     IF (KI(1).EQ.1) GO TO 130          BES1340
     CALL ADRESS (1,JA)                 BES1350
     IF (JA.LE.0) CALL ERROR (11)         BES1360
     LT=1                                 BES1370
130 K=0                                  BES1380
     KA=0                                BES1390
     IF (LT+LV.EQ.0) GO TO 200          BES1400
     IF (LV.EQ.0) GO TO 230             BES1410
     IF (LT.EQ.0) GO TO 240             BES1420
140 IF (NERROR.NE.0) RETURN             BES1430
     DO 190 I=1,NRMAX                  BES1440
     IF (KA.EQ.0) X=R(JA)              BES1450
     JA=JA+1                            BES1460
     E=1.D0                             BES1470
     IF (JX.NE.0) Y=R(JB)              BES1480
     JB=JB+1                            BES1490
C   L2=15,M=2,N=0,L=0                  :KBKZERO OF $$ PUT REAL IN ++ IMAGINARY ++ BES1500
C   L2=16,M=2,N=1,L=0                  :KBKONE OF $$ PUT REAL IN ++ IMAGINARY ++ BES1510
C   L2=19,M=2,N=0,L=1                  :KEXKZR OF $$ PUT REAL IN ++ IMAGINARY ++ BES1520
C   L2=20,M=2,N=1,L=1                  :KEXKONE OF $$ PUT REAL IN ++ IMAGINARY ++ BES1530
C   L2=23,M=2,N=0,L=0                  :CKZERO OF $$ OHI $$ UT REAL ++ IMAG ++ BES1540
C   L2=24,7=2,N=1,L=0                  :CKONE OF $$ PHI $$ PUT REAL ++ IMAG ++ BES1550
C   L2=27,M=2,N=0,L=1                  :CEKZERO OF $$ PHI $$ PUT REAL ++ IMAG ++ BES1560
C   L2=28,M=2,N=1,L=1                  :CEKONE OF $$ PHI $$ PUT REAL ++ IMAG ++ BES1570
C   IF (M.EQ.2) CALL CBEK             BES1580
C   L2=13,M=1,N=0,L=0                  :KBIZERO OF $$ PUT REAL IN ++ IMAGINARY ++ BES1590
C   L2=14,M=1,N=1,L=0                  :KBIONE OF $$ PUT REAL IN ++ IMAGINARY ++ BES1600
C   L2=18,M=1,N=1,L=1                  :KEXIONE OF $$ PUT REAL IN ++ IMAGINARY ++ BES1610
C   L2=21,M=1,N=0,L=0                  :CIZERO OF $$ PHI $$ PUT REAL ++ IMAG ++ BES1620
C   L2=22,M=18N=1,L=0                  :CIONE OF $ PHI $ PUT REAL ++ IMAG ++ BES1630
C   L2=25,M=1,N=0,L=1                  :CEIZERO OF $$ HI $ UT REAL ++ IMAG ++ BES1640
C   L2=26,M=1,N=1,L=1                  :CEIONE OF $$ PHI $$ PUT REAL ++ IMAG ++ BES1650
C   IF (M.EQ.1) CALL CBEI             BES1660
Z=X*FDCOS(Y)
IF (L.EQ.1) GO TO 150                 BES1670
IF (DABS(Z).LT.XEX) GO TO 160         BES1680
CALL ERROR (105)                      BES1690
150 E=FDEXP(Z)                        BES1700
     IF (M.EQ.1) E=1.D0/E              BES1710
160 IF (N.EQ.0) GO TO 170             BES1720
C   STORE INTO WORK SHEET RESULTS OF COMMANDS KBIONE, KBKONE BES1730
                                         BES1740

```

C	KEXIONE, KEXKONE, CIONE, CEIONE, CEKONE	BES1750
	R(J1)=E*S	BES1760
	R(J2)=E*T	BES1770
	GO TO 180	BES1780
C	STORE INTO WORK SHEET RESULTS OF COMMANDS KBIZERO, KBKZERO,	BES1790
C	KEXIZER, KEXKZER, CIZERO, CEIZERO, CEKZERO	BES1800
170	R(J1)=E*P	BES1810
	R(J2)=E*Q	BES1820
180	J1=J1+1	BES1830
190	J2=J2+1	BES1840
	RETURN	BES1850
200	IF (JX.EQ.0) GO TO 240	BES1860
	Y=ARGS(2)	BES1870
	X=ARGS(1)	BES1880
	KA=1	BES1890
	JX=0	BES1900
	GO TO 140	BES1910
210	IF (NARGS.GT.4) CALL ERROR (10)	BES1920
	JX=1	BES1930
	LV=0	BES1940
	IF (KI(2).EQ.1) GO TO 220	BES1950
	CALL ADRESS (2,JB)	BES1960
	IF (JB.LE.0) CALL ERROR (11)	BES1970
	LV=1	BES1980
220	M=1	BES1990
	IF (L2.GT.22) M=2	BES2000
	IF (L2.GT.24) M=1	BES2010
	IF (L2.GT.26) M=2	BES2020
	L=0	BES2030
	IF (L2.GT.24) L=1	BES2040
	GO TO 120	BES2050
230	IF (JX.EQ.0) GO TO 140	BES2060
	Y=ARGS(2)	BES2070
	JX=0	BES2080
	GO TO 140	BES2090
240	KA=1	BES2100
	X=ARGS(1)	BES2110
	GO TO 140	BES2120
250	IF (L2.GT.32) GO TO 350	BES2130
260	IF (NARGS.GT.2) CALL ERROR (10)	BES2140
	CALL ADRESS (NARGS,J)	BES2150
	IF (J.LE.0) CALL ERROR (11)	BES2160
	LT=0	BES2170
	IF (KI(1).EQ.1) GO TO 270	BES2180
	CALL ADRESS (1,JA)	BES2190
	IF (JA.LE.0) CALL ERROR (11)	BES2200
	LT=1	BES2210
270	IF (NERROR.NE.0) RETURN	BES2220
	IF (LT.EQ.0) X=ARGS(1)	BES2230
	IF (L2.GT.37) GO TO 310	BES2240
	IF (L2.EQ.32) GO TO 310	BES2250
	IF (L2.GT.29) GO TO 290	BES2260
	DO 280 N=1,NR	BES2270
	IF (LT.EQ.1) X=R(JA)	BES2280
	JA=JA+1	BES2290
C	L2=29 :INTJO OF \$\$ STORE IN ++	BES2300
	R(J)=BINTJO(X,W,Z)	BES2310
280	J=J+1	BES2320
	RETURN	BES2330

```

290 K=1                                BES2340
    IF (L2.EQ.31) K=2                  BES2350
    DO 300 N=1,NR                      BES2360
    IF (LT.EQ.1) X=R(JA)                BES2370
    JA=JA+1                            BES2380
C   L2=30,K=1                         :ELLIPTICAL FIRST OF $$ STORE IN ++
C   L2=31,K=2                         :ELLIPTICAL SECOND OF $$ STORE IN ++
    R(J)=COMELL(X,K)                  BES2390
300 J=J+1                             BES2400
    RETURN                            BES2410
310 IF (LT.EQ.1) CALL ERROR (20)       BES2420
    Z=X                               BES2430
    K=NR                             BES2440
    IF (K.LE.100) GO TO 330           BES2450
    K=100                            BES2460
    JA=J+100                          BES2470
    DO 320 I=K,NR                      BES2480
    R(JA)=0.0                          BES2490
320 JA=JA+1                           BES2500
C   L2=32                           :BESJN X= ** STORE IN ++
330 IF (L2.EQ.32) CALL BEJN (0,W,Z)  BES2510
C   L2=38                           :BESIN X= ** STORE IN ++
    IF (L2.EQ.38) CALL BEJN (1,W,Z)  BES2520
    IF (L2.EQ.39) GO TO 460          BES2530
    DO 340 N=1,K                      BES2540
    R(J)=W(N)                        BES2550
340 J=J+1                             BES2560
    RETURN                            BES2570
350 IF (L2.GT.34) GO TO 390          BES2580
    L=NR                             BES2590
    IF (NR.GT.1000) L=1000            BES2600
    IF (NARGS.GT.2) CALL ERROR (10)  BES2610
    CALL ADRESS (NARGS,J)            BES2620
    IF (J.LE.0) CALL ERROR (11)      BES2630
    IF (KI(1).EQ.1) CALL ERROR (20)  BES2640
    CALL ADRESS (1,JA)                BES2650
    IF (JA.LE.0) CALL ERROR (11)      BES2660
    IF (NERROR.NE.0) RETURN          BES2670
    IF (L2.EQ.33) GO TO 360          BES2680
C   L2=34                           :ZEROS BJZERO STORE IN ++ AND ++
    CALL BEZONE (AA,B,1,L)           BES2690
    GO TO 370                         BES2700
C   L2=33                           :ZEROS BJONE STORE IN ++ AND ++
360 CALL BEZERO (AA,B,1,L)             BES2710
370 DO 380 N=1,L                      BES2720
    R(JA)=AA(N)                      BES2730
    R(J)=B(N)                        BES2740
    JA=JA+1                          BES2750
380 J=J+1                             BES2760
    RETURN                            BES2770
390 IF (L2.GT.36) GO TO 430          BES2780
    IF (NARGS.GT.2) CALL ERROR (10)  BES2790
    CALL ADRESS (NARGS,J)            BES2800
    IF (J.LE.0) CALL ERROR (11)      BES2810
    LT=0                             BES2820
    IF (KI(1).EQ.1) GO TO 400        BES2830
    CALL ADRESS (1,JA)                BES2840
    IF (JA.LE.0) CALL ERROR (11)      BES2850
    LT=1                             BES2860

```

```

400 IF (LT.EQ.0) X=ARGS(1) BES2930
    IF (NERROR.NE.0) RETURN BES2940
    K=0 BES2950
    IF (L2.EQ.36) K=1 BES2960
    DO 420 N=1,NR BES2970
    IF (LT.NE.0) X=R(JA) BES2980
    JA=JA+1 BES2990
C   L2=35,K=0 :STRUVE ZERO OF $$ STORE IN ++ BES3000
C   L2=36,K=1 :STRUVE ONE OF $$ STORE IN ++ BES3010
    CALL STRUVE (X,Y,Z,W) BES3020
    IF (K.EQ.0) GO TO 410 BES3030
C   STORE RESULTS FOR STRUVE ONE BES3040
    R(J)=Z BES3050
    GO TO 420 BES3060
C   STORE RESULTS OF STRUVE ZERO BES3070
410 R(J)=Y BES3080
420 J=J+1 BES3090
    RETURN BES3100
430 IF (L2.GT.37) GO TO 260 BES3110
    IF (NARGS.GT.3) CALL ERROR (10) BES3120
    CALL ADRESS (NARGS,J) BES3130
    IF (J.LE.0) CALL ERROR (11) BES3140
    IF (KI(1).EQ.1) CALL ERROR (20) BES3150
    CALL ADRESS (1,JA) BES3160
    IF (JA.LE.0) CALL ERROR (11) BES3170
    JB=IA(2) BES3180
    IF (KI(2).NE.0) CALL ERROR (3) BES3190
    IF (NERROR.NE.0) RETURN BES3200
    K=IA(2) BES3210
    LNR=NRMAX BES3220
    IF (LNR.GT.1000) LNR=1000 BES3230
    DO 440 N=1,LNR BES3240
    AA(N)=R(JA) BES3250
440 JA=JA+1 BES3260
C   L2=37 :HARMONIC OF ++ ,, STORE IN ++ BES3270
    CALL FOURIA (AA,B(1),B(2),K,L) BES3280
    DO 450 N=1,JB BES3290
    R(J)=B(N) BES3300
450 J=J+1 BES3310
    RETURN BES3320
460 IF (X.LT.XEXP) GO TO 470 BES3330
    CALL ERROR (225) BES3340
    RETURN BES3350
C   L2=39 :BESKN X= ** STORE IN ++ BES3360
470 AA(1)=DBEJ(X,0,7) BES3370
    AA(2)=DBEJ(X,1,7) BES3380
    R(J)=AA(1) BES3390
    R(J+1)=AA(2) BES3400
    J=J+2 BES3410
    DO 480 I=3,K BES3420
    Z=I-2 BES3430
    AA(I)=AA(I-2)+2.*Z*AA(I-1)/X BES3440
    R(J)=AA(I) BES3450
    IF (AA(I).GT.3.E37) GO TO 490 BES3460
480 J=J+1 BES3470
    RETURN BES3480
490 DO 500 JA=I,K BES3490
    R(J)=0.0 BES3500
500 J=J+1 BES3510
    RETURN BES3520
    END BES3530

```

```

C      SUBROUTINE BEZERO (A,B,M,L)
      VERSION 5.00      BEZERO      5/15/70
      DOUBLE PRECISION A(1),B(1),X,Y,AA,AB,AC,FDSQRT
      KB=1
      N=M
10     J=4*N-1
      IF (J.GT.44) GO TO 130
      GO TO (20,30,40,50,60,70,80,90,100,110,120), N
20     X=2.404825577D0
      Y=.5191474973D0
      GO TO 140
30     X=5.5200781103D0
      Y=-.3402648065D0
      GO TO 140
40     X=8.6537279129D0
      Y=.2714522999D0
      GO TO 140
50     X=11.7915344391D0
      Y=-.2324598314D0
      GO TO 140
60     X=14.9309177086D0
      Y=.2065464331D0
      GO TO 140
70     X=18.0710639679D0
      Y=-.187728803D0
      GO TO 140
80     X=21.2116366299D0
      Y=.1732658942D0
      GO TO 140
90     X=24.3524715308D0
      Y=-.1617015507D0
      GO TO 140
100    X=27.493479132D0
      Y=.1521812138D0
      GO TO 140
110    X=30.6346064684D0
      Y=-.1441659777D0
      GO TO 140
120    X=33.7758202136D0
      Y=.1372969434D0
      GO TO 140
130    X=J
      X=X*3.1415926536D0
      AA=1.D0/X**2
      AB=1.D0+2.D0*AA*(1.D0-AA*(31.D0-AA*6277237.D0/7.D0)/
      15.D0)/3.D0)
      J=N/2
      J=2*J
      AC=1.D0
      IF (J.EQ.N) AC=-1.D0
      Y=AC*1.595769122D0*(1.D0-AA**2*56.D0/3.D0)/FDSQRT(X)
      X=X*AB/4.D0
140    A(KB)=X
      B(KB)=Y
      N=N+1
      KB=KB+1
      IF (KB.LE.L) GO TO 10
      RETURN
      END

```

```

C      SUBROUTINE BEZONE (A,B,M,L)                                BEO  10
      VERSION 5.00      BEZONE      5/15/70                         BEO  20
      DOUBLE PRECISION A(1),B(1),R,S,T,X,Y,FDSQRT                  BEO  30
      KB=1                                         BEO  40
      N=M                                         BEO  50
10     J=4*N+1                                         BEO  60
      IF (J.GT.46) GO TO 130                           BEO  70
      GO TO (20,30,40,50,60,70,80,90,100,110,120), N          BEO  80
20     X=3.8317059702D0                               BEO  90
      Y=-.4027593957D0                               BEO 100
      GO TO 140                                       BEO 110
30     X=7.0155866698D0                               BEO 120
      Y=.3001157525D0                               BEO 130
      GO TO 140                                       BEO 140
40     X=10.1734681351D0                             BEO 150
      Y=-.2497048771D0                             BEO 160
      GO TO 140                                       BEO 170
50     X=13.3236919363D0                             BEO 180
      Y=.2183594072D0                             BEO 190
      GO TO 140                                       BEO 200
60     X=16.4706300509D0                             BEO 210
      Y=-.1964653715D0                             BEO 220
      GO TO 140                                       BEO 230
70     X=19.6158585105D0                             BEO 240
      Y=.180063375D0                               BEO 250
      GO TO 140                                       BEO 260
80     X=22.7600843806D0                             BEO 270
      Y=-.1671846005D0                             BEO 280
      GO TO 140                                       BEO 290
90     X=25.9036720876D0                             BEO 300
      Y=.1567249863D0                               BEO 310
      GO TO 140                                       BEO 320
100    X=29.0468285349D0                            BEO 330
      Y=-.1480111100D0                            BEO 340
      GO TO 140                                       BEO 350
110    X=32.1896799110D0                            BEO 360
      Y=.1406057982D0                            BEO 370
      GO TO 140                                       BEO 380
120    X=35.3323075501D0                            BEO 390
      Y=-.1342112403D0                            BEO 400
      GO TO 140                                       BEO 410
130    X=J                                         BEO 420
      X=X*3.1415926536D0                            BEO 430
      R=1.D0/X**2                               BEO 440
      S=1.D0-6.D0*R*(1.D0-R*(1.D0-R*(157.2D0-130080.6D0*R/7.D0))) BEO 450
      J=N/2                                         BEO 460
      J=2*J                                         BEO 470
      T=1.D0                                         BEO 480
      IF (J.NE.N) T=-1.D0                           BEO 490
      Y=T*1.595769122D0*(1.D0+R**2*24.D0*(1.D0-81.6D0*R))/FDSQRT(X) BEO 500
      X=S*X/4.D0                               BEO 510
140    A(KB)=X                                         BEO 520
      B(KB)=Y                                         BEO 530
      N=N+1                                         BEO 540
      KB=KB+1                                         BEO 550
      IF (KB.LE.L) GO TO 10                          BEO 560
      RETURN                                         BEO 570
      END                                           BEO 580

```

```

C      FUNCTION BINTJO (X,A,Z)          BIN   10
      VERSION 5.00      BINTJO      5/15/70    BIN   20
      DIMENSION A(1)          BIN   30
      DOUBLE PRECISION BINTJO,A,Z,X,B,C,DBEJ    BIN   40
      Z=DABS(X)          BIN   50
      IF (Z.GT.100.) GO TO 20    BIN   60
      CALL BEJN (0,A,Z)        BIN   70
      IF (Z.GT.60.) GO TO 30    BIN   80
      B=.0D0          BIN   90
      DO 10 N=2,100,2        BIN  100
10     B=B+A(N)          BIN  110
      B=2.D0*B          BIN  120
      GO TO 40          BIN  130
20     A(1)=DBEJ(Z,0,1)        BIN  140
      A(2)=DBEJ(Z,1,1)        BIN  150
30     C=1.D0/Z**2        BIN  160
      B=1.D0+A(2)*(1.D0-C*(1.D0-C*(9.D0-C*(225.D0-C*11025.D0))))    BIN  170
      C=1.D0-C*(3.D0-C*(45.D0-C*(1575.D0-99225.D0*C)))    BIN  180
      B=B-A(1)*C/Z        BIN  190
40     BINTJO=B        BIN  200
      RETURN          BIN  210
      END          BIN  220

```

```

C SUBROUTINE BJORCK (X,B,NP,A,F) BJO 10
C VERSION 5.00      BJORCK      5/15/70 BJO 20
C INPUT IS X,B AND NP. BJO 30
C OUTPUT IS F. BJO 40
C THIS SUBROUTINE WAS ADAPTED BY ROY H. WAMPLER AND M. STUART SCOTT, BJO 50
C NATIONAL BUREAU OF STANDARDS, WASHINGTON, D. C., JULY 1969, FROM BJO 60
C A SUBROUTINE CALLED 'BJORCK' WHICH WAS WRITTEN BY WILLIAM J. HALL, BJO 70
C NATIONAL BUREAU OF STANDARDS, BOULDER, COL. THIS ROUTINE USES THE BJO 80
C MODIFIED GRAM-SCHMIDT ALGORITHM GIVEN BY AKE BJORCK IN 'SOLVING BJO 90
C LINEAR LEAST SQUARES PROBLEMS BY GRAM-SCHMIDT ORTHOGONALIZATION' BJO 100
C 'BIT' VOL. 7 (1967), PAGES 1-21. BJO 110
C DOUBLE PRECISION C,D,R,Y,FDSQRT BJO 115
C DIMENSION X(1), B(1), A(3,1) BJO 120
C DIMENSION C(3,3), D(3), R(3), Y(4) BJO 130
C BJO 140
C C INITIALIZE A AND FORM SUM OF SQUARES OF THE B VECTOR BJO 150
Y(4)=0.D0 BJO 160
DO 10 I=1,NP BJO 170
A(1,I)=1.0 BJO 180
A(2,I)=X(I) BJO 190
A(3,I)=X(I)*X(I) BJO 200
10 Y(4)=Y(4)+B(I)*B(I) BJO 210
NF=3 BJO 220
D(1)=0.D0 BJO 230
Y(1)=0.D0 BJO 240
DO 20 I=1,NP BJO 250
D(1)=A(1,I)*A(1,I)+D(1) BJO 260
20 Y(1)=A(1,I)*B(I)+Y(1) BJO 270
Y(1)=Y(1)/D(1) BJO 280
IR=0 BJO 290
DO 60 K=2,NF BJO 300
DO 40 J=K,NF BJO 310
IR=IR+1 BJO 320
R(IR)=0.D0 BJO 330
DO 30 I=1,NP BJO 340
30 R(IR)=A(K-1,I)*A(J,I)+R(IR) BJO 350
R(IR)=R(IR)/D(K-1) BJO 360
DO 40 I=1,NP BJO 370
40 A(J,I)=A(J,I)-A(K-1,I)*R(IR) BJO 380
D(K)=0.D0 BJO 390
Y(K)=0.D0 BJO 400
DO 50 I=1,NP BJO 410
50 B(I)=B(I)-A(K-1,I)*Y(K-1) BJO 420
Y(K)=A(K,I)*B(I)+Y(K) BJO 430
D(K)=A(K,I)*A(K,I)+D(K) BJO 440
60 Y(K)=Y(K)/D(K) BJO 450
IRS=-NF BJO 460
DO 90 K=1,NF BJO 470
IRS=IRS+NF-K+1 BJO 480
IR=IRS BJO 490
DO 90 JJ=1,K BJO 500
JJ=K-JJ+1 BJO 510
C(K,J)=Y(J) BJO 520
IF (JJ-1) 90,90,70 BJO 530
70 DO 80 I=2,JJ BJO 540
C(K,J)=C(K,J)-C(K,K-I+2)*R(IR) BJO 550
80 IR=IR-1 BJO 560
IR=IR-NF+K BJO 570
90 DO 100 I=1,NF BJO 580

```

```
100 Y(I)=Y(I)*FDSQRT(D(I))          BJ0 590
    F=Y(3)*Y(3)*FLOAT(NP-3)/(Y(4)-Y(1)*Y(1)-Y(2)*Y(2)-Y(3)*Y(3))  BJ0 600
    RETURN                                BJ0 610
    END                                    BJ0 620
```

C BLOCK DATA BLOCK BLO 10  
 C VERSION 5.00 BLOCK 5/15/70 BLO 20  
 C BLOCK DATA BLO 30  
 COMMON / ABCDEF / L( 48 ) BLO 40  
 COMMON/HEADER/NOCARD(80),ITLE(60,6),LNCNT,IPRINT,NPAGE,IPUNCH BLO 50  
 COMMON/FMAT/IFMTX( 6 ),IOSWT,IFMTS( 6 ),LHEAD(96) BLO 60  
 C BLOCK BLO 70  
 C ABCDEF BLO 80  
 C           L(1) =1H0 L(2) =1H1 L(3) =1H2 L(4) =1H3 L(5) =1H4 BLO 90  
 C           L(6) =1H5 L(7) =1H6 L(8) =1H7 L(9) =1H8 L(10)=1H9 BLO 100  
 C           L(11)=1HA L(12)=1HB L(13)=1HC L(1J)=1HD L(15)=1HE BLO 110  
 C           L(16)=1HF L(17)=1HG L(18)=1HH L(19)=1HI L(20)=1HJ BLO 120  
 C           L(21)=1HK L(22)=1HL L(23)=1HM L(24)=1HN L(25)=1HO BLO 130  
 C           L(26)=1HP L(27)=1HQ L(28),1HR L(29)=1HS L(30)=1HT BLO 140  
 C           L(31)=1HU L(32)=1HV L(33)=1HW L(34)=1HX L(35)=1HY BLO 150  
 C           L(36)=1HZ L(37)=1H/ L(38)=1H. L(39)=1H- L(40)=1H+ BLO 160  
 C           L(41)=1H\* L(42)=2H( L(43)=1H) L(44)=1H, L(45)=1H BLO 170  
 C           L(46)=1H= L(47)=1H\$ ( 48)=1H, BLO 180  
 DATA L(1),L(2),L(3),L(4),L(5),L(6),L(7),L(8),L(9),L(10)/ BLO 190  
 1 1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/ BLO 200  
 DATA L(11),L(12),L(13),L(14),L(15),L(16),L(17),L(18),L(19),L(20)/BLO 210  
 1 1HA,1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ/ BLO 220  
 DATA L(21),L(22),L(23),L(24),L(25),L(26),L(27),L(28),L(29),L(30)/BLO 230  
 1 1HK,1HL,1HM,1HN,1HO,1HP,1HQ,1HR,1HS,1HT/ BLO 240  
 DATA L(31),L(32),L(33),L(34),L(35),L(36),L(37),L(38),L(39),L(40)/BLO 250  
 1 1HU,1HV,1HW,1HX,1HY,1HZ,1H/,1H.,1H-,1H+/ BLO 260  
 DATA L(41),L(42),L(43),L(44),L(45),L(46),L(47),L(48)/ BLO 270  
 1 1H\*,1H(,1H),1H,,1H,1H=,1H\$,1H,/ BLO 280  
 C NOCARD CONTAINS THE MESSAGE WRITTEN SAVED FROM THE OMNITAB CARD BLO 290  
 DATA NOCARD(1),NOCARD(2),NOCARD(3),NOCARD(4),NOCARD(5),NOCARD(6), BLO 300  
 1NOCARD(7),NOCARD(8),NOCARD(9),NOCARD(10),NOCARD(11),NOCARD(12), BLO 310  
 2NOCARD(13),NOCARD(14),NOCARD(15),NOCARD(16),NOCARD(17),NOCARD(18),BLO 320  
 3NOCARD(19),NOCARD(20),NOCARD(21),NOCARD(22),NOCARD(23),NOCARD(24),BLO 330  
 4NOCARD(25),NOCARD(26),NOCARD(27),NOCARD(28),NOCARD(29),NOCARD(30),BLO 340  
 5NOCARD(31),NOCARD(32),NOCARD(33),NOCARD(34),NOCARD(35),NOCARD(36),BLO 350  
 6NOCARD(37),NOCARD(38),NOCARD(39),NOCARD(40)/ BLO 360  
 71H ,1H , BLO 370  
 81H ,1H , BLO 380  
 9 1H ,1H ,1H ,1H ,1H0,1HM,1HN,1HI/ BLO 390  
 DATA NOCARD(41),NOCARD(42),NOCARD(43),NOCARD(44),NOCARD(45), BLO 400  
 1 NOCARD(46),NOCARD(47),NOCARD(48),NOCARD(49),NOCARD(50), BLO 410  
 2 NOCARD(51),NOCARD(52),NOCARD(53),NOCARD(54),NOCARD(55), BLO 420  
 3 NOCARD(56),NOCARD(57),NOCARD(58),NOCARD(59),NOCARD(60), BLO 430  
 4 NOCARD(61),NOCARD(62),NOCARD(63),NOCARD(64),NOCARD(65), BLO 440  
 5 NOCARD(66),NOCARD(67),NOCARD(68),NOCARD(69),NOCARD(70), BLO 450  
 6 NOCARD(71),NOCARD(72),NOCARD(73),NOCARD(74),NOCARD(75), BLO 460  
 7 NOCARD(76),NOCARD(77),NOCARD(78),NOCARD(79),NOCARD(80)/ BLO 470  
 81HT,1HA,1HB,1H ,1H , BLO 480  
 91H ,1H , BLO 490  
 A1H ,1H ,1H ,1H ,1H ,1H ,1H / BLO 500  
 C IFMTS CONTAINS FORMAT USED BY PRINT BLO 510  
 DATA IFMTS(1),IFMTS(2),IFMTS(3),IFMTS(4),IFMTS(5),IFMTS(6)/ BLO 520  
 1 1H(,2H1P,2H8E,3H15.,1H6,1H)/ BLO 530  
 END BLO 540

```

SUBROUTINE CBEI          CBI   10
VERSION 5.00      CBEI      5/15/70      CBI   20
      COMPUTES I0(Z) AND I1(Z) FOR COMPLEX ARGUMENT R*E(IS)=Z      CBI   30
COMMON /ABEKI/ R,S,A,B,C,D      CBI   40
DOUBLE PRECISION A,B,C,D,R,S      CBI   50
DOUBLE PRECISION E,F,G,H,P,Q,T,X,Y,Z,V,U,W,AA      CBI   60
DOUBLE PRECISION FDSIN,FDCOS,FDEXP,FDSQRT      CBI   70
E=FDCOS(S)      CBI   80
F=FDSIN(S)      CBI   90
IF (R.GT.15.5) GO TO 30      CBI 100
P=1.D0-2.D0**F**2      CBI 110
AA=P      CBI 120
Q=2.D0**E**F      CBI 130
W=Q      CBI 140
A=1.D0      CBI 150
B=0.D0      CBI 160
C=1.D0      CBI 170
U=0.D0      CBI 180
G=1.D0      CBI 190
T=2.D0      CBI 200
X=(R/2.D0)**2      CBI 210
V=X      CBI 220
Y=X      CBI 230
DO 10 N=1,60      CBI 240
Z=1.D0/G**2      CBI 250
H=1.D0/(G*T)      CBI 260
A=A+X*Z*P      CBI 270
B=B+X*Z*Q      CBI 280
C=C+V*H*P      CBI 290
U=U+V*H*Q      CBI 300
X=X*Y*Z      CBI 310
IF (X.LT..5D-10) GO TO 20      CBI 320
V=V*Y*H      CBI 330
Z=P      CBI 340
P=Z*AA-Q*W      CBI 350
Q=Q*AA+Z*W      CBI 360
G=G+1.D0      CBI 370
10 T=T+1.D0      CBI 380
20 D=R*(C*F+U*E)/2.D0      CBI 390
C=R*(C*E-U*F)/2.D0      CBI 400
GO TO 60      CBI 410
30 Z=FDEXP(R*E)/FDSQRT(6.283185307D0*R)      CBI 420
X=S/2.D0-R*F      CBI 430
Y=Z*FDCOS(X)      CBI 440
Z=Z*FDSIN(X)      CBI 450
W=-1.D0      CBI 460
G=1.D0      CBI 470
H=3.D0      CBI 480
P=E      CBI 490
Q=F      CBI 500
T=1.D0      CBI 510
U=0.D0      CBI 520
V=1.D0      CBI 530
X=0.D0      CBI 540
A=1.D0      CBI 550
B=1.D0/(8.D0*R)      CBI 560
C=B      CBI 570
D=B      CBI 580
DO 40 N=1,20      CBI 590

```

	AA=B**G**2/A	CBI 600
	T=T+AA*P	CBI 610
	U=U+AA*Q	CBI 620
	AA=C**W**H/A	CBI 630
	V=V+AA*P	CBI 640
	X=X+AA*Q	CBI 650
	B=B*D**G**2/A	CBI 660
	IF (B.LT..5D-10) GO TO 50	CBI 670
	C=C*D**W**H/A	CBI 680
	W=W+2.D0	CBI 690
	G=G+2.D0	CBI 700
	H=H+2.D0	CBI 710
	A=A+1.D0	CBI 720
	AA=P*E-Q*F	CBI 730
	Q=F*P+E*Q	CBI 740
40	P=AA	CBI 750
50	A=Y*T-Z*U	CBI 760
	B=-(Y*U+T*Z)	CBI 770
	C=Y*V-Z*X	CBI 780
	D=-(Y*X+Z*V)	CBI 790
60	RETURN	CBI 800
	END	CBI 810

```

SUBROUTINE CBEK                               CBK 10
C      VERSION 5.00      CBEK      5/15/70      CBK 20
C      COMPUTES K0(Z) AND K1(Z) FOR COMPLEX ARGUMENT Z=R*E(IS)      CBK 30
COMMON /ABEKI/ R,S,A,B,C,D      CBK 40
DOUBLE PRECISION A,B,C,D,R,S      CBK 50
DOUBLE PRECISION E,F,G,H,P,Q,T,U,V,W,X,Y,Z,AA(40),AB(40),AC,AD,AE      CBK 60
DOUBLE PRECISION FDLOG,FDCOS,FDEXP,FDSIN,FDSQRT      CBK 70
COMMON /SCRAT/ NS,NS2,SCRAT(13500)      CBK 80
EQUIVALENCE (SCRAT(1700),AA), (SCRAT(1800),AB)      CBK 90
IF (R.LE.0.0) GO TO 80      CBK 100
E=FDCOS(S)      CBK 110
F=FDSIN(S)      CBK 120
IF (R.GT.8.) GO TO 40      CBK 130
P=1.D0-2.D0*F**2      CBK 140
Q=2.D0*E*F      CBK 150
W=P      CBK 160
Z=Q      CBK 170
X=(R/2.D0)**2      CBK 180
Y=X      CBK 190
V=X      CBK 200
G=E*(4.D0*E**2-3.D0)      CBK 210
H=F*(3.D0-4.D0*F**2)      CBK 220
T=FDLOG(R/2.D0)+.5772156649D0      CBK 230
A=-T      CBK 240
B=-S      CBK 250
C=E*(T-0.5D0)-S*F      CBK 260
U=F*(T-0.5D0)+S*E      CBK 270
AC=1.D0      CBK 280
AD=2.D0      CBK 290
AA(1)=1.D0      CBK 300
AB(1)=1.25D0      CBK 310
DO 10 N=2,40      CBK 320
AE=N      CBK 330
AA(N)=AA(N-1)+1.D0/AE      CBK 340
10 AB(N)=AA(N)+1.D0/(2.D0*(AE+1.D0))      CBK 350
DO 20 N=1,40      CBK 360
AE=T-AA(N)      CBK 370
D=P*AE-S*Q      CBK 380
AE=Q*AE+S*P      CBK 390
A=A-D*X/AC**2      CBK 400
B=B-AE*X/AC**2      CBK 410
AE=T-AB(N)      CBK 420
D=G*AE-H*S      CBK 430
AE=H*AE+G*S      CBK 440
C=C+D*Y/(AC*AD)      CBK 450
U=U+AE*Y/(AC*AD)      CBK 460
X=X*V/AC**2      CBK 470
IF (X.LT..5D-10) GO TO 30      CBK 480
Y=Y*V/(AC*AD)      CBK 490
AC=AC+1.D0      CBK 500
AD=AD+1.D0      CBK 510
AE=P      CBK 520
P=AE*W-Q*Z      CBK 530
Q=Q*W+AE*Z      CBK 540
AE=G      CBK 550
G=AE*W-H*Z      CBK 560
20 H=H*W+AE*Z      CBK 570
30 C=E/R+R*C/2.D0      CBK 580
D=-F/R+R*U/2.D0      CBK 590

```

```

GO TO 70                               CBK 600
40  U=FDEXP(-R*E)*FDSQRT(1.5707963268D0/R)   CBK 610
V=R*F+S/2.D0                           CBK 620
Y=U*FDCOS(V)                          CBK 630
Z=U*FDSIN(V)                          CBK 640
W=-1.D0                                CBK 650
G=1.D0                                 CBK 660
H=3.D0                                 CBK 670
P=E                                    CBK 680
Q=F                                    CBK 690
T=1.D0                                CBK 700
U=0.D0                                CBK 710
V=1.D0                                CBK 720
X=0.D0                                CBK 730
A=1.D0                                CBK 740
B=1.D0/(8.D0*R)                      CBK 750
C=B                                    CBK 760
D=B                                    CBK 770
AC=-1.D0                             CBK 780
DO 50 N=1,12                           CBK 790
AD=AC*B**2/A                         CBK 800
AE=AC*C*W*H/A                       CBK 810
T=T+AD*P                            CBK 820
U=U-AD*Q                            CBK 830
V=V+AE*P                            CBK 840
X=X-AE*Q                            CBK 850
AD=B                                  CBK 860
B=B*D**2/A                         CBK 870
IF (B.GT.AD) GO TO 60                CBK 880
IF (B.LT..5D-10) GO TO 60           CBK 890
C=C*D*W*H/A                         CBK 900
W=W+2.D0                            CBK 910
H=H+2.D0                            CBK 920
G=G+2.D0                            CBK 930
A=A+1.D0                            CBK 940
AC=-1.D0*AC                         CBK 950
AD=P                                  CBK 960
P=AD*E-Q*F                          CBK 970
50  Q=Q*E+AD*F                         CBK 980
60  A=Y*T+U*Z                          CBK 990
B=Y*U-T*Z                           CBK1000
C=Y*V+X*Z                           CBK1010
D=Y*X-V*Z                           CBK1020
70  RETURN                             CBK1030
80  A=0.D0                            CBK1040
B=-.785398163397D0                  CBK1050
C=0.D0                                CBK1060
D=0.D0                                CBK1070
CALL ERROR (101)                     CBK1080
RETURN                                CBK1090
END                                   CBK1100

```

```

SUBROUTINE CHANGE                                CHA 10
C      VERSION 5.00      CHANGE      5/15/70      CHA 20
C
C      CHANGE SIGNS OF COLS ++, ++, ++, ETC.      CHA 30
C
      COMMON /BLOCRC/ NRC,RC(12600)                CHA 60
      COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NCHA 70
      IARGS,VWXYZ(8),NERROR                         CHA 80
      DIMENSION ARGS(100)                           CHA 90
      EQUIVALENCE (ARGS(1),RC(12501))              CHA 100
      IF (NARGS) 50,50,10                            CHA 110
10     DO 30 I=1,NARGS                           CHA 120
      CALL ADDRESS (I,J)                           CHA 130
      IF (J) 40,60,20                            CHA 140
20     IF (NERROR.NE.0) RETURN                  CHA 150
      DO 30 N=1,NRMAX                           CHA 160
      JJ=J+N-1                                 CHA 170
30     RC(JJ)=-RC(JJ)                           CHA 180
      GO TO 70                                 CHA 190
40     CALL ERROR (3)                           CHA 200
      GO TO 70                                 CHA 210
50     CALL ERROR (10)                          CHA 220
      GO TO 70                                 CHA 230
60     CALL ERROR (11)                          CHA 240
70     RETURN                                  CHA 250
      END                                     CHA 260

```

```

SUBROUTINE CHKCOLUMN (J)                         CHK 10
C      VERSION 5.00      CHKCOLUMN      5/15/70      CHK 20
      COMMON /BLOCRC/ NRC,RC(12600)                CHK 30
      COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NCHK 40
      IARGS,VWXYZ(8),NERROR                         CHK 50
      DIMENSION ARGS(100)                           CHK 60
      EQUIVALENCE (ARGS(1),RC(12501))              CHK 70
C
C      THIS ROUTINE CHECKS THAT ALL ,NARGS, ARGUMENTS ARE LEGAL      CHK 90
C      COLUMN NUMBERS AND CONVERTS THEM IN IARGS TO THEIR BEGINNING      CHK 100
C      ADDRESSES.                                              CHK 110
      IF (NARGS.GT.0) GO TO 20                          CHK 120
10     J=1                                         CHK 130
      GO TO 40                                       CHK 140
20     DO 30 I=1,NARGS                           CHK 150
      CALL ADDRESS (I,IARGS(I))                     CHK 160
      IF (IARGS(I).LE.0) GO TO 10                  CHK 170
30     CONTINUE                                    CHK 180
      J=0                                         CHK 190
40     RETURN                                     CHK 200
      END                                         CHK 210

```

```

SUBROUTINE CKIND (J)                               CKI 10
C      VERSION 5.00      CKIND      5/15/70          CKI 20
C      CKIND                                         CKI 30
C      S PEAVY   5/22/67                           CKI 40
C      THE FIRST J VALUES OF KIND ARE CHECKED      CKI 50
C      IF ALL ARE =0 THEN J=0                      CKI 60
C      IF ALL ARE =1 THEN J=1                      CKI 70
C      IF SOME ARE 0 AND SOME 1    J=2            CKI 80
C      COMMON /BLOCRC/ NRC,RC(12600)              CKI 90
C      COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NCKI 100
1ARGS,VWXYZ(8),NERROR                           CKI 110
DIMENSION ARGS(100)                            CKI 120
EQUIVALENCE (ARGS(1),RC(12501))               CKI 130
JA=J                                           CKI 140
J=0                                           CKI 150
DO 10 I=1,JA                                     CKI 160
IF (KIND(I).NE.0) GO TO 20                     CKI 170
10 CONTINUE                                       CKI 180
RETURN                                         CKI 190
20 J=1                                           CKI 200
DO 30 I=1,JA                                     CKI 210
IF (KIND(I).NE.1) GO TO 40                     CKI 220
30 CONTINUE                                       CKI 230
RETURN                                         CKI 240
40 J=2                                           CKI 250
RETURN                                         CKI 260
END                                            CKI 270

```

```

SUBROUTINE CMPARA (X1,X2,X3,Y1,Y2,Y3,X,Y)      CMP 10
C      VERSION 5.00      CMPARA      5/15/70          CMP 20
C      PROGRAM CMPARA WRITTEN BY MRS. CARLA MESSINA NBS-NSRDS JUNE 68 CMP 30
C      CMPARA IS USED BY THE INSTRUCTION MAXMIN IN PROGRAM CMSEPA  CMP 40
A=((Y2-Y3)*(X2-X1)-(Y2-Y1)*(X2-X3))/((X2-X1)*(X2**2-X3**2)-(X2-X3)CMP 50
1*(X2**2-X1**2))                                CMP 60
B=((Y2-Y1)-A*(X2**2-X1**2))/(X2-X1)           CMP 70
C=-A*X3**2-B*X3+Y3                             CMP 80
X=-B/(2.*A)                                      CMP 90
Y=A*X**2+B*X+C                                  CMP 100
RETURN                                         CMP 110
END                                            CMP 120

```

```

SUBROUTINE CMSEPA
C VERSION 5.00      CMSEPA      5/15/70          CMS 10
COMMON /BLOCRC/ NRC,RC(12600)          CMS 20
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NCMS 40
IARGS,VWXYZ(8),NERROR          CMS 50
DIMENSION ARGS(100)          CMS 60
EQUIVALENCE (ARGS(1),RC(12501))          CMS 70
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG          CMS 80
COMMON /SCRAT/ NS,NS2,A(13500)          CMS 90
C L2=2 SEPARATE FROM COL ++ EVERY , , ROW START WITH ROW , , STORE INCMS 100
C L2=3 INSERT IN COL ++ FROM COL ++ AT EVERY , , ROW STARTING AS , , CMS 120
C           STORE IN COL ++
C L2=4 MAXMIN X IN ++ Y IN ++, XMAX IN ++ YMAX IN ++, XMIN ++ YMIN CMS 140
C PROGRAM CMSEPA WRITTEN BY MRS. CARLA MESSINA NBS-NSRDS JUNE 196CMS 150
C GO TO (10,10,30,310), L2          CMS 160
10 NARGS=NARGS+1          CMS 170
J=NARGS          CMS 180
DO 20 I=2,NARGS          CMS 190
IARGS(J)=IARGS(J-1)          CMS 200
KIND(J)=KIND(J-1)          CMS 210
20 J=J-1          CMS 220
30 CALL CKIND (J)          CMS 230
IF (J) 40,70,40          CMS 240
40 K=3          CMS 250
50 CALL ERROR (K)          CMS 260
60 RETURN          CMS 270
70 IF (NARGS-5) 80,100,80          CMS 280
80 K=10          CMS 290
IF (L2-2) 90,90,50          CMS 300
90 NARGS=NARGS-1          CMS 310
GO TO 50          CMS 320
100 M=IARGS(3)          CMS 330
N=IARGS(4)          CMS 340
DO 120 I=3,4          CMS 350
IF (IARGS(I)) 130,130,110          CMS 360
110 IF (IARGS(I)-NROW) 120,120,130          CMS 370
120 IARGS(I)=IARGS(1)
CALL CHKCOL (J)
IF (J) 40,140,40          CMS 400
130 K=16          CMS 410
GO TO 50          CMS 420
140 IF (NERROR.NE.0) GO TO 60          CMS 430
IF (NRMAX) 150,150,160          CMS 440
150 K=9          CMS 450
GO TO 50          CMS 460
160 DO 170 I=1,NARGS          CMS 470
170 IARGS(I)=IARGS(I)-1          CMS 480
L=IARGS(5)          CMS 490
IF (L2-2) 180,180,210          CMS 500
C SEPARATE          CMS 510
180 DO 190 I=1,NRMAX          CMS 520
J=IARGS(1)+I          CMS 530
190 A(I)=RC(J)          CMS 540
DO 200 K=N,NRMAX,M          CMS 550
L=L+1          CMS 560
200 RC(L)=A(K)          CMS 570
GO TO 60          CMS 580
C INSERT          CMS 590
210 M=M-1          CMS 600

```

	IF (M) 130,130,220	CMS 610
220	N=N-1	CMS 620
	IF(N) 130,130,230	CMS 630
230	KA=0	CMS 640
	I=IARGS(1)	CMS 650
	DO 240 K=1,N	CMS 660
	I=I+1	CMS 670
	KA=KA+1	CMS 680
240	A(KA)=RC(I)	CMS 690
	NN=KA+IARGS(1)	CMS 700
	MM=IARGS(2)	CMS 710
	DO 260 K=N,NRMAX,M	CMS 720
	KA=KA+1	CMS 730
	MM=MM+1	CMS 740
	A(KA)=RC(MM)	CMS 750
	DO 250 LL=1,M	CMS 760
	KA=KA+1	CMS 770
	NN=NN+1	CMS 780
250	A(KA)=RC(NN)	CMS 790
260	CONTINUE	CMS 800
	I=(NRMAX-N)/M+1	CMS 810
	IF (I+NRMAX-NROW) 270,270,280	CMS 820
270	NRMAX=NRMAX+I	CMS 830
	GO TO 290	CMS 840
280	NRMAX=NROW	CMS 850
	CALL ERROR (219)	CMS 860
290	DO 300 K=1, NRMAX	CMS 870
	L=L+1	CMS 880
300	RC(L)=A(K)	CMS 890
	GO TO 60	CMS 900
310	IF (NARGS-6) 80,320,80	CMS 910
320	CALL CHKCOL (J)	CMS 920
	IF (J) 40,330,40	CMS 930
330	IF (NERROR.NE.0) GO TO 60	CMS 940
	KA=0	CMS 950
	IUP=-1	CMS 960
	IF (NRMAX) 150,150,340	CMS 970
340	IF (NRMAX-2) 350,350,390	CMS 980
350	IF (KA) 370,370,360	CMS 990
360	CALL ERROR (220)	CMS1000
370	IF (IUP) 380,60,60	CMS1010
380	K=219	CMS1020
	GO TO 50	CMS1030
390	DO 400 K=1, NRMAX	CMS1040
	I=IARGS(1)+K-1	CMS1050
	J=IARGS(2)+K-1	CMS1060
	A(K)=RC(I)	CMS1070
	K2=K+NRMAX	CMS1080
400	A(K2)=RC(J)	CMS1090
	I1=IARGS(3)-1	CMS1100
	J1=IARGS(4)-1	CMS1110
	K1=IARGS(5)-1	CMS1120
	L1=IARGS(6)-1	CMS1130
	IF (NRMAX-4) 410,520,520	CMS1140
410	K2=NRMAX+1	CMS1150
	IF (A(K2)-A(K2+1)) 420,380,430	CMS1160
420	IF (A(K2+1)-A(K2+2)) 380,380,440	CMS1170
430	IF (A(K2+1)-A(K2+2)) 450,380,380	CMS1180
440	IUP=IUP+1	CMS1190

```

450 IUP=IUP+1 CMS1200
    IF (A(1)-A(2)) 460,480,460 CMS1210
460 IF (A(1)-A(3)) 470,480,470 CMS1220
470 IF (A(2)-A(3)) 490,480,490 CMS1230
480 KA=1 CMS1240
    GO TO 350 CMS1250
490 CALL CMPARA (A(1),A(2),A(3),A(K2),A(K2+1),A(K2+2),X1,Y1) CMS1260
    IF (IUP) 500,500,510 CMS1270
500 RC(K1+1)=X1 CMS1280
    RC(L1+1)=Y1 CMS1290
    GO TO 60 CMS1300
510 RC(I1+1)=X1 CMS1310
    RC(L1+1)=Y1 CMS1320
    GO TO 60 CMS1330
520 I=NRMAX-2 CMS1340
    DO 730 K=1,I CMS1350
    IEQUAL=1 CMS1360
    K2=K+NRMAX CMS1370
    IF (A(K2)-A(K2+1)) 530,610,540 CMS1380
530 IF (A(K2+1)-A(K2+2)) 730,730,550 CMS1390
540 IF (A(K2+1)-A(K2+2)) 560,730,730 CMS1400
550 IUP=1 CMS1410
    GO TO 570 CMS1420
560 IUP=0 CMS1430
570 IF (A(K)-A(K+1)) 580,600,580 CMS1440
580 IF (A(K)-A(K+2)) 590,600,590 CMS1450
590 IF (A(K+1)-A(K+2)) 680,600,680 CMS1460
600 KA=KA+1 CMS1470
    GO TO 730 CMS1480
610 IF (K-1) 730,730,620 CMS1490
620 IEQUAL=2 CMS1500
    IF (A(K-1)-A(K)) 630,600,630 CMS1510
630 IF (A(K-1)-A(K+1)) 640,600,640 CMS1520
640 IF (A(K-1)-A(K+2)) 650,600,650 CMS1530
650 IF (A(K2-1)-A(K2)) 660,730,670 CMS1540
660 IF (A(K2+1)-A(K2+2)) 730,730,550 CMS1550
670 IF (A(K2+1)-A(K2+2)) 560,730,730 CMS1560
680 CALL CMPARA (A(K),A(K+1),A(K+2),A(K2),A(K2+1),A(K2+2),X1,Y1) CMS1570
    IF (IEQUAL-2) 700,690,690 CMS1580
690 CALL CMPARA (A(K-1),A(K),A(K+1),A(K2-1),A(K2),A(K2+1),X2,Y2) CMS1590
    X1=0.5*(X1+X2) CMS1600
    Y1=0.5*(Y1+Y2) CMS1610
700 IF (IUP) 710,710,720 CMS1620
710 IUP=0 CMS1630
    K1=K1+1 CMS1640
    L1=L1+1 CMS1650
    RC(K1)=X1 CMS1660
    RC(L1)=Y1 CMS1670
    GO TO 730 CMS1680
720 I1=I1+1 CMS1690
    J1=J1+1 CMS1700
    RC(I1)=X1 CMS1710
    RC(J1)=Y1 CMS1720
730 CONTINUE CMS1730
    GO TO 350 CMS1740
    END CMS1750

```

```

SUBROUTINE COALES          COA  10
C VERSION 5.00   COALES   5/15/70      COA  20
C ACOALESCE AND AAVERAGE COMMANDS      COA  30
C WRITTEN BY R. MCCLENON, NSRDS-NBS, NOV. 1969      COA  40
C L1 = ACOALESCE      L2 = AAVERAGE      COA  50
C COMMAND FORM IS --      COA  60
C     ACOALESCE MATRIX STARTING IN R++ C++      COA  70
C     ACOALESCE ON FIRST COL OF ARRAY IN , , ++ R=,, C=,, START STORING COA  80
C     IN , , ++
C     AAVERAGE ON FIRST COL OF ARRAY IN , , ++ R=,, C=,, START STORING COA 100
C     IN , , ++
C     OR      COA 120
C     ACOALESCE ON ** IN FIRST COL OF , , ++ R=,, C=,, START STORING COA 130
C     IN , , ++
C     AAVERAGE ON ** IN FIRST COL OF , , ++ R=,, C=,, START STORING COA 150
C     IN , , ++
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG      COA 170
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NCOA 180
1ARGS,VWXYZ(8),NERROR      COA 190
COMMON /SCRAT/ NS,NS2,A(13500)      COA 200
COMMON /BLOCRC/ NRC,RC(12600)      COA 210
DIMENSION ARGS(100)      COA 220
EQUIVALENCE (ARGS(1),RC(12501))      COA 230
L2=L2-8      COA 240
IF (NARGS-6) 410,10,20      COA 250
10 KL=1      COA 260
I=6      COA 270
CALL CKIND (I)      COA 280
IF (I-1) 60,420,420      COA 290
20 KL=2      COA 300
DO 30 J=2,7      COA 310
IF (KIND(J)) 420,30,420      COA 320
30 CONTINUE      COA 330
IF (NARGS-8) 40,410,410      COA 340
40 IF (KIND(1)) 420,420,50      COA 350
50 Y=ARGS(1)      COA 360
60 KL5=KL+5      COA 370
DO 70 J=KL,KL5      COA 380
IF (IARGS(J)) 430,430,70      COA 390
70 CONTINUE      COA 400
LROW=IARGS(KL+2)      COA 410
LCOL=IARGS(KL+3)      COA 420
KROW=IARGS(KL)      COA 430
KCOL=IARGS(KL+1)      COA 440
IF (KROW+LROW-NROW-1) 80,80,440      COA 450
80 IF (KCOL+LCOL-NCOL-1) 90,90,440      COA 460
90 MROW=IARGS(KL+4)      COA 470
MCOL=IARGS(KL+5)      COA 480
IF (MROW+LROW-NROW-1) 100,100,440      COA 490
100 IF (MCOL+LCOL-NCOL-1) 110,110,440      COA 500
110 IF (NERROR) 400,120,400      COA 510
120 KRR=KROW+LROW-1      COA 520
KCC=KCOL+LCOL-1      COA 530
MRR=MROW+LROW-1      COA 540
MCC=MCOL+LCOL-1      COA 550
N=0      COA 560
IF (KL-1) 230,230,130      COA 570
130 DO 140 J=2,LCOL      COA 580
140 A(J)=0.0      COA 590

```

```

A(1)=Y COA 600
DO 170 J=1,LROW COA 610
I=KROW+J-1 COA 620
II=NROW*(KCOL-1)+I COA 630
IF (RC(II)-Y) 170,150,170 COA 640
150 DO 160 JJ=2,LCOL COA 650
I=II+(JJ-1)*NROW COA 660
160 A(JJ)=A(JJ)+RC(I) COA 670
N=N+1 COA 680
170 CONTINUE COA 690
IF (N) 460,460,180 COA 700
180 M=1 COA 710
GO TO (190,200), L2 COA 720
190 DIV=1.0 COA 730
GO TO 210 COA 740
200 DIV=N COA 750
210 DO 220 JJ=2,LCOL COA 760
220 A(JJ)=A(JJ)/DIV COA 770
GO TO 380 COA 780
230 MAT=LCOL*LROW COA 790
IF (MAT-NS+LROW) 240,240,450 COA 800
240 M1=MAT+1 COA 810
M2=MAT+LROW COA 820
DO 250 J=M1,M2 COA 830
250 A(J)=0. COA 840
M=0 COA 850
DO 370 J=1,LROW COA 860
M1=MAT+J COA 870
IF (A(M1)) 370,260,370 COA 880
260 K1=LCOL*M+1 COA 890
K2=LCOL*(M+1) COA 900
DO 270 K=K1,K2 COA 910
270 A(K)=0.0 COA 920
N=0 COA 930
L=NROW*(KCOL-1)+J COA 940
Y=RC(L) COA 950
A(K1)=Y COA 960
DO 300 JJ=J,LROW COA 970
II=NROW*(KCOL-1)+KROW+JJ-1 COA 980
IF (RC(II)-Y) 300,280,300 COA 990
280 K3=K1+1 COA1000
M1=MAT+JJ COA1010
A(M1)=1.0 COA1020
DO 290 K=K3,K2 COA1030
I=II+(K-K1)*NROW COA1040
290 A(K)=A(K)+RC(I) COA1050
N=N+1 COA1060
300 CONTINUE COA1070
IF (N) 370,370,310 COA1080
310 M=M+1 COA1090
320 GO TO (330,340), L2 COA1100
330 DIV=1.0 COA1110
GO TO 350 COA1120
340 DIV=N COA1130
350 DO 360 K=K3,K2 COA1140
360 A(K)=A(K)/DIV COA1150
370 CONTINUE COA1160
380 DO 390 J=1,M COA1170
DO 390 JJ=1,LCOL COA1180

```

I=LCOL*(J-1)+JJ	COA1190
II=NROW*(MCOL-2+JJ)+J+MROW-1	COA1200
RC(II)=A(I)	COA1210
390 CONTINUE	COA1220
400 RETURN	COA1230
410 CALL ERROR (10)	COA1240
GO TO 400	COA1250
420 CALL ERROR (20)	COA1260
GO TO 400	COA1270
430 CALL ERROR (11)	COA1280
GO TO 400	COA1290
440 CALL ERROR (17)	COA1300
GO TO 400	COA1310
450 CALL ERROR (23)	COA1320
GO TO 400	COA1330
460 CALL ERROR (203)	COA1340
M=1	COA1350
GO TO 380	COA1360
END	COA1370

```

FUNCTION COMELL (Z,I)                                     COM 10
C VERSION 5.00      COMELL    '5/15/70                  COM 20
C   COMPLETE ELLIPTIC INTEGRALS - FIRST AND SECOND KIND  COM 30
C DOUBLE PRECISION Z,X,A,B,C,D,E,P,Q                   COM 40
C DOUBLE PRECISION FDLOG,FDSQRT,COMELL                 COM 50
C X=Z                                                     COM 60
C IF (DABS(Z).LT.1.D0) GO TO 10                         COM 70
C IF(DABS(Z).EQ.1.D0.AND.I.EQ.2) GO TO 10               COM 80
C CALL ERROR (109)                                       COM 83
C Z=0.0D0                                                 COM 85
C RETURN                                                 COM 90
10 A=X                                                   COM 100
B=FDSQRT(1.D0-A)                                         COM 110
IF (X.GT..996D0) GO TO 50                               COM 120
B=(1.D0-B)/(1.D0+B)                                       COM 130
A=B**2                                                 COM 140
B=1.D0+B                                              COM 150
C=1.D0                                                 COM 160
D=C                                                   COM 170
E=2.D0                                                 COM 180
IF (I.EQ.1) GO TO 20                         COM 190
B=1.D0/B                                              COM 200
D=-1.D0                                              COM 210
20 P=A                                                   COM 220
DO 30 N=1,90                                           COM 230
C=C+P*(D/E)**2                                         COM 240
P=P*A*(D/E)**2                                         COM 250
IF (P.LT..1D-9) GO TO 40                           COM 260
D=D+2.D0                                              COM 270
30 E=E+2.D0                                             COM 280
40 A=B*C*1.570796326D0                                COM 290
GO TO 70
50 A=FDLOG(4.0D0/B)                                    COM 310
Q=B**2                                                 COM 320
IF (I.GT.1) GO TO 60                         COM 330
B=.25D0*(A-1.D0)                                         COM 340
C=.140625D0*(A-1.666666666D0)                      COM 350
D=9.765625D-2*(A-1.233333333D0)                    COM 360
E=1255.D0*(A-1.27904761904D0)/16384.D0            COM 370
A=A+Q*(B+Q*(C+Q*(D+Q*E)))                         COM 380
GO TO 70
60 B=.5D0*(A-.5D0)                                         COM 400
C=.1875D0*(A-1.083333333D0)                         COM 410
D=.1171875D0*(A-1.2D0)                                COM 420
E=175.D0*(A-1.251190476D0)/2048.D0                COM 430
A=1.+Q*(B+Q*(C+Q*(D+Q*E)))                         COM 440
70 COMELL=A                                              COM 450
RETURN                                                 COM 460
C
END                                                 COM 470
                                         COM 490

```

```

SUBROUTINE COMPLX                               COX  10
C      VERSION 5.00      COMPLX    5/15/70       COX  20
C      ***** COMMON *****                           COX  30
C      COMMON /BLOCRC/ NRC,RC(12600)               COX  40
C      COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NCOX 50
C      IARGS,VWXYZ(8),NERROR                      COX  60
C      DIMENSION ARGS(100)                         COX  70
C      EQUIVALENCE (ARGS(1),RC(12501))            COX  80
C      COMMON /SCRAT/ NS,NS2,A(13500)              COX  90
C      COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG        COX 100
C      COMMON /CONSTS/ PI,E,HALFPI,DEG,RAD,XALOG   COX 110
C      *****                                         COX 120
C      DIMENSION KK(6)                            COX 130
C      EQUIVALENCE (I1,IARGS(1)), (I2,IARGS(2)), (I3,IARGS(3)), (I4,IARGS) COX 140
C      I(4)), (I5,IARGS(5)), (I6,IARGS(6))          COX 150
C      DOUBLE PRECISION D(5),X,Y,FDCOS,FDSIN,FDSQRT COX 160
C      *****                                         COX 170
C      WRITTEN BY DAVID HOBGEN, SEL, NBS. 4/24/70. COX 180
C      CADD (E),(E), TO (E),(E) AND PUT IN COLUMNS (C),(C) COX 190
C      CSUBTRACT (E),(E), FROM (E),(E) AND PUT IN COLUMNS (C),(C) COX 200
C      CMULTIPLY (E),(E) BY (E),(E) AND PUT IN COLUMNS (C),(C) COX 210
C      CDIVIDE (E),(E) BY (E),(E) AND PUT IN COLUMNS (C),(C) COX 220
C      CRECTANGULAR R IN (E) THETA IN (E), PUT X IN COL (C), Y IN COL (C) COX 230
C      CPOLAR OF X IN (E) Y IN (E) PUT R IN COL (C) THETA IN COL (C) COX 240
C      *****                                         COX 250
C      FIRST ARGUMENT OF EACH PAIR IS REAL, SECOND IS IMAGINARY COX 260
C      *****                                         COX 270
C      VALUES OF L2 ARE ***                           COX 280
C      1=CADD,     2=CSUB,     3=CMULT,     4=CDIV,     5=CRECTAN  6=CPOLAR COX 290
C      *****                                         COX 300
C      NARGS = 6 FOR ADD,SUB,MULT,DIV               COX 310
C      NARGS = 4 FOR CRECTANGULAR AND CPOLAR        COX 320
C      *****                                         COX 330
C      ERROR CHECKING                             COX 340
10     IF (NARGS.NE.6.AND.L2.LT.5) CALL ERROR (10) COX 350
      IF (NARGS.NE.4.AND.L2.GT.4) CALL ERROR (10) COX 360
      IF (KIND(NARGS).NE.0.AND.KIND(NARGS-1).NE.0) CALL ERROR (20) COX 370
      IF (NRMAX.EQ.0) CALL ERROR (9)                COX 380
      DO 30 I=1,NARGS                            COX 390
      KK(I)=1                                     COX 400
      CALL ADRESS (I,IARGS(I))                   COX 410
      IF (IARGS(I)) 20,40,30                     COX 420
20     KK(I)=0                                     COX 430
      IARGS(I)=-IARGS(I)                         COX 440
30     CONTINUE                                    COX 450
      GO TO 50                                     COX 460
40     CALL ERROR (11)                           COX 470
50     IF (NERROR.NE.0) RETURN                  COX 480
      JJ=IARGS(NARGS)+NRMAX-1                   COX 490
      IF (L2.GT.4) GO TO 140                     COX 500
      DO 130 I=I6,JJ                            COX 510
      DO 60 J=1,4                                COX 520
      MM=IARGS(J)                                COX 530
60     D(J)=RC(MM)                                COX 540
      GO TO (70,80,90,100), L2                  COX 550
C     CADD
70     X=D(1)+D(3)                                COX 560
      Y=D(2)+D(4)                                COX 570
      GO TO 120                                  COX 580
                                              COX 590

```

```

C CSUBTRACT COX 600
80 X=D(3)-D(1) COX 610
Y=D(4)-D(2) COX 620
GO TO 120 COX 630
C CMULTIPLY COX 640
90 X=D(1)*D(3)-D(2)*D(4) COX 650
Y=D(1)*D(4)+D(3)*D(2) COX 660
GO TO 120 COX 670
C CDIVIDE COX 680
C ZERO RETURNED IF DIVISION BY ZERO, DIAGNOSTIC GIVEN. ERROR(104). COX 690
100 D(5)=D(3)**2+D(4)**2 COX 700
IF (D(5).GT.0.) GO TO 110 COX 710
CALL ERROR(106)
X=0.0D0 COX 720
Y=0.0D0 COX 730
GO TO 120 COX 740
110 X=(D(1)*D(3)+D(2)*D(4))/D(5) COX 750
Y=(D(3)*D(2)-D(1)*D(4))/D(5) COX 760
COX 770
120 RC(I5)=FDPCON(X) COX 780
RC(I)=FDPCON(Y) COX 790
I1=I1+KK(1) COX 800
I2=I2+KK(2) COX 810
I3=I3+KK(3) COX 820
I4=I4+KK(4) COX 830
130 I5=I5+KK(5) COX 840
RETURN COX 850
C CRECTANGULAR AND CPOLAR COX 860
140 MM=L2-4 COX 870
DO 260 I=I4,JJ COX 880
D(1)=RC(I1) COX 890
D(2)=RC(I2) COX 900
GO TO (150,190), MM COX 910
C CRECTANGULAR - R,THETA TO X,Y COX 920
150 IF (RC(I1)) 180,160,180 COX 930
160 X=0.0D0 COX 940
170 Y=0.0D0 COX 950
GO TO 250 COX 960
180 X=D(1)*FDCOS(D(2)) COX 970
Y=D(1)*FDSIN(D(2)) COX 980
GO TO 250 COX 990
C CPOLAR X,Y TO R,THETA COX1000
190 IF (RC(I2)) 220,200,220 COX1010
200 IF (RC(I1)) 210,160,210 COX1020
C Y=0, X NE 0 COX1030
210 X=DABS(D(1)) COX1040
GO TO 170 COX1050
220 IF (RC(I1)) 240,230,240 COX1060
C X=0, Y NE 0 COX1070
230 X=DABS(D(2)) COX1080
C IF X=0.0, THEN THETA=HALFPI*SIGN(Y) COX1090
Y=SIGN(HALFPI,RC(I2)) COX1100
GO TO 250 COX1110
240 X=FDSQRT(D(1)**2+D(2)**2) COX1120
Y=DATAN2(D(2),D(1)) COX1130
250 RC(I3)=FDPCON(X) COX1140
RC(I)=FDPCON(Y) COX1150
I1=I1+KK(1) COX1160
I2=I2+KK(2) COX1170
260 I3=I3+KK(3) COX1180
RETURN COX1190
END COX1200

```

```

SUBROUTINE CORREL                               COR 10
C      VERSION 5.00    CORREL   5/15/70          COR 20
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NCOR 30
IARGS,VWXYZ(8),NERROR                         COR 40
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG           COR 50
COMMON /BLOCRC/ NRC,RC(12600)                  COR 60
DIMENSION ARGS(100)                           COR 70
EQUIVALENCE (ARGS(1),RC(12501))              COR 80
COMMON /HEADER/ NOCARD(80),ITLE(60,6),LNCNT,IPRINT,NPAGE,IPUNCH COR 90
COMMON /SCRAT/ NS,NS2,A(13500)                COR 100
DIMENSION ERR(3), AVG(100), SD(100), T(100)  COR 110
EQUIVALENCE (A(13301),AVG(1)), (A(13401),SD(1),T(1)) COR 120
L2=L2-10                                     COR 122
IF (L2.EQ.1.OR.NARGS.NE.IARGS(1) +1) GO TO 10 COR 124
CALL ERROR (233)                            COR 126
RETURN                                         COR 128
10     MVAR=99                                  COR 130
IF (NARGS.LT.3) CALL ERROR (10)              COR 140
NVAR=IARGS(1)                                COR 150
IF (NVAR.LT.2.OR.NVAR.GT.MVAR) CALL ERROR (3) COR 160
KEEP=(NARGS-NVAR+1)/2                         COR 170
IF (KEEP.LT.1.OR.KEEP.GT.3.OR.MOD(NARGS-NVAR 2).EQ.0) CALL ERROR COR 180
110    GO TO (40,30,20), KEEP                  COR 190
20     K2=IARGS(NVAR+4)                        COR 200
IF (K2.LT.1.OR.K2.GT.NROW) CALL ERROR (16)  COR 210
IARGS(NVAR+4)=1                               COR 220
COR 230
30     K1=IARGS(NVAR+2)                        COR 240
IF (K1.LT.1.OR.K1.GT.NROW) CALL ERROR (16)  COR 250
IARGS(NVAR+2)=1                               COR 260
COR 270
40     CALL CHKCOLUMN (J)                      COR 280
IF (J.EQ.0) GO TO 50                         COR 290
CALL ERROR (11)                            COR 300
RETURN                                         COR 310
50     GO TO (80,70,60), KEEP                  COR 320
60     IARGS(NVAR+4)=IARGS(NVAR+5)+K2-1       COR 330
70     IARGS(NVAR+2)=IARGS(NVAR+3)+K1-1       COR 340
80     LOTTE=NVAR*NVAR                         COR 350
KURT=2*LOTTE                                 COR 360
C      LOT IS SPACE IN ARRAY A RESERVED FOR RANKED DATA COR 370
LOT=MAX0(NRMAX*(NVAR+1),3*LOTTE+8*NVAR+8,4*LOTTE)
IF (NRMAX.LT.3) CALL ERROR (9)               COR 380
IF (NRMAX*NVAR.GT.NRC) CALL ERROR (15)       COR 390
IF (MAX0(LOT+LOTTE+100, NRMAX*4+3*LOTTE).GT.NS) CALL ERROR (23) COR 400
IF (NERROR.NE.0) RETURN                      COR 410
NVA=NVAR-1                                    COR 420
IF (L2.EQ.2) GO TO 130                      COR 430
C      RANKS OF OBSERVATIONS                  COR 440
IND=NVAR*NRMAX+1                            COR 450
DO 90 I=1,NVAR                             COR 460
K1=(I-1)*NRMAX+1                          COR 470
K2=IARGS(I+1)                                COR 480
90     CALL RANKX (NRMAX,RC(K2),A(IND),A(K1),T(I)) COR 490
C      SPEARMAN RANK CORRELATION COEFFICIENT COR 500
F=((NRMAX-1)*NRMAX*(NRMAX+1))/6            COR 510
I1=LOT+LOTTE                                COR 520
A(I1)=1.                                      COR 530
DO 120 J=1,NVA                            COR 540
IND=(J-1)*NVAR+LOT                          COR 550

```

```

I1=IND+J                                     COR 560
A(I1)=1.                                      COR 570
IJ=J+1                                         COR 580
DO 120 I=IJ ,NVAR                           COR 590
I1=IND+I                                       COR 600
I2=(I-1)*NVAR+J+LOT                          COR 610
K1=2.*T(I)+0.4                               COR 620
K2=2.*T(J)+0.4                               COR 630
IF (IFIX(F)-K1.GT.0.AND.IFIX(F)-K2.GT.0) GO TO 100 COR 640
A(I1)=0.                                       COR 650
GO TO 120                                     COR 660
100 D=0.                                         COR 670
DO 110 K=1 ,NRMAX                           COR 680
K1=(I-1)*NRMAX+K                            COR 690
K2=(J-1)*NRMAX+K                            COR 700
110 D=D+(A(K1)-A(K2))*(A(K1)-A(K2))        COR 710
A(I1)=(F-D-T(I)-T(J))/(FSQRT(F-2.*T(I))*FSQRT(F-2.*T(J))) COR 720
IF (ABS(A(I1)).GT.1.) A(I1)=AINT(A(I1))      COR 730
120 A(I2)=A(I1)                                COR 740
C MEANS OF OBSERVATIONS                      COR 750
130 DO 150 I=1 ,NVAR                           COR 760
AVG(I)=0.                                      COR 770
SUMPOS=0.                                       COR 780
SUMNEG=0.                                       COR 790
DO 140 J=1 ,NRMAX                           COR 800
I1=IARGS(I+1)-1+J                           COR 810
SUMPOS=SUMPOS+AMAX1(RC(I1),0.)                COR 820
140 SUMNEG=SUMNEG+AMAX1(-RC(I1),0.)           COR 830
150 AVG(I)=(SUMPOS-SUMNEG)/FLOAT(NRMAX)       COR 840
C STANDARD DEVIATIONS                      COR 850
DO 170 I=1 ,NVAR                           COR 860
SD(I)=0.                                       COR 870
DO 160 J=1 ,NRMAX                           COR 880
I1=IARGS(I+1)-1+J                           COR 890
160 SD(I)=SD(I)+(RC(I1)-AVG(I))*(RC(I1)-AVG(I)) COR 900
170 SD(I)=FSQRT(SD(I))                      COR 910
C SIMPLE CORRELATION COEFFICIENT            COR 920
A(LOTTE)=1.                                    COR 930
Z=FLOAT(NRMAX-2)                             COR 940
DO 200 J=1 ,NVA                            COR 950
IND=(J-1)*NVAR                           COR 960
I1=IND+J                                     COR 970
A(I1)=1.                                       COR 980
IJ=J+1                                         COR 990
DO 200 I=IJ ,NVAR                           COR1000
I1=IND+I                                     COR1010
I2=(I-1)*NVAR+J                            COR1020
IF (SD(I).GT.0..AND.SD(J).GT.0.) GO TO 180 COR1030
A(I1)=0.                                       COR1040
GO TO 200                                     COR1050
180 SUMPOS=0.                                    COR1060
SUMNEG=0.                                       COR1070
DO 190 K=1 ,NRMAX                           COR1080
K1=IARGS(I+1)-1+K                           COR1090
K2=IARGS(J+1)-1+K                           COR1100
CP=(RC(K1)-AVG(I))*(RC(K2)-AVG(J))         COR1110
SUMPOS=SUMPOS+AMAX1(CP,0.)                   COR1120
190 SUMNEG=SUMNEG+AMAX1(-CP,0.)               COR1130
A(I1)=(SUMPOS-SUMNEG)/(SD(I)*SD(J))         COR1140

```

```

      IF (ABS(A(I1)).GT.1.) A(I1)=AINT(A(I1))          COR1150
200    A(I2)=A(I1)                                     COR1160
      IF (NVAR.LE.2) GO TO 210                         COR1170
C     PARTIAL CORRELATION COEFFICIENT                COR1180
      CALL INVCOR (A(1),NVAR,NVAR,A(LOTTE+1),NVAR+2,A(1),1,ERR,IND) COR1190
      IF (IND.EQ.0) GO TO 220                         COR1200
      WRITE (IPRINT,480)                               COR1210
210    KEEP=MIN0(KEEP,2)                             COR1220
      A(LOTTE+1)=0.                                    COR1230
      IF (L2.EQ.1) GO TO 300                         COR1240
      GO TO 420                                      COR1250
220    DO 230 J=2,NVAR                            COR1260
      K1=2*(J-1)                                    COR1270
      K2=LOTTE+(J-1)*NVAR                          COR1280
      DO 230 I=1,NVAR                            COR1290
      I1=K2+I                                      COR1300
      I2=I1+K1                                     COR1310
230    A(I1)=A(I2)                                COR1320
      DO 240 J=1,NVA                            COR1330
      IND=LOTTE+(J-1)*NVAR                        COR1340
      K1=IND+J                                     COR1350
      IJ=J+1                                       COR1360
      DO 240 I=IJ,NVAR                           COR1370
      K2=LOTTE+(I-1)*NVAR+I                      COR1380
      I1=IND+I                                     COR1390
      I2=LOTTE+(I-1)*NVAR+J                      COR1400
      A(I1)=-A(I1)/(FSQRT(A(K1))*FSQRT(A(K2))) COR1410
      IF (ABS(A(I1)).GT.1.) A(I1)=AINT(A(I1))          COR1420
240    A(I2)=A(I1)                                COR1430
      DO 250 I=1,NVAR                            COR1440
      I1=LOTTE+(I-1)*NVAR+I                      COR1450
250    A(I1)=1.                                    COR1460
      IF (L2.EQ.2) GO TO 420                      COR1470
      IF (NRMAX.LE.NVAR) GO TO 300                COR1480
C     SIGNIFICANCE LEVEL OF PARTIAL CORRELATION COEFFICIENT COR1490
      Z=FLOAT(NRMAX-NVAR)                         COR1500
      DO 290 J=1,NVAR                            COR1510
      IJ=(J-1)*NVAR+LOTTE                       COR1520
      IND=IJ+KURT                                COR1530
      DO 290 I=J,NVAR                           COR1540
      I1=IND+I                                     COR1550
      I2=(I-1)*NVAR+J+3*LOTTE                   COR1560
      K1=IJ+I                                     COR1570
      IF (A(K1)) 270,260,270                     COR1580
260    A(I1)=1.                                    COR1590
      GO TO 290                                  COR1600
270    IF (ABS(A(K1)).LT.1.) GO TO 280          COR1610
      A(I1)=0.                                     COR1620
      GO TO 290                                  COR1630
280    F=A(K1)*A(K1)                            COR1640
      F=Z*F/(1.-F)                              COR1650
      CALL PROB (1.,Z,F,A(I1))                  COR1660
290    A(I2)=A(I1)                            COR1670
300    Z=FLOAT(NRMAX-2)                         COR1680
      DO 340 J=1,NVAR                            COR1690
      IJ=(J-1)*NVAR                           COR1700
      IND=IJ+KURT                                COR1710
      DO 340 I=J,NVAR                           COR1720
      I1=IND+I                                     COR1730

```

```

I2=(I-1)*NVAR+J+KURT COR1740
K1=IJ+I COR1750
IF (A(K1)) 320,310,320 COR1760
310 A(I1)=1. COR1770
GO TO 340 COR1780
320 IF (ABS(A(K1)).LT.1.) GO TO 330 COR1790
A(I1)=0. COR1800
GO TO 340 COR1810
330 F=A(K1)*A(K1) COR1820
F=Z*F/(1.-F) COR1830
CALL PROB (1.,Z,F,A(I1)) COR1840
340 A(I2)=A(I1) COR1850
C FIRST PRINTING STAGE (SCC,PCC,SIGNIFICANCE LEVELS, SRCC) COR1860
NLA=1 COR1870
CALL PAGE (4) COR1880
WRITE (IPRINT,490) NVAR,NRMAX COR1890
CALL MIST (NVAR,A(1),1,NLA,1) COR1900
CALL MIST (NVAR,A(KURT+1),1,NLA,2) COR1910
IF (A(LOTTE+1).LE.0.) GO TO 350 COR1920
CALL MIST (NVAR,A(LOTTE+1),1,NLA,3) COR1930
F=ABS(AMAX1(ERR(1),ERR(2),ERR(3))) COR1940
C FOLLOWING STATEMENT WILL BE RESTORED LATER COR1950
C WRITE (IPRINT, 45) F COR1960
C 45 FORMAT (/1H , 72ERROR BOUND FOR INVERSION OF MATRIX OF SIMPLE COR1970
C 1RELATION COEFFICIENTS =, E14.4) COR1980
NLA=NLA+2 COR1990
IF (NRMAX.LE.NVAR) GO TO 350 COR2000
CALL MIST (NVAR,A(3*LOTTE+1),1,NLA,4) COR2010
350 CALL MIST (NVAR,A(LOT+1),1,NLA,5) COR2020
IF (NRMAX.GT.3) GO TO 360 COR2030
WRITE (IPRINT,500) NRMAX COR2040
GO TO 420 COR2050
360 Z=FLOAT(NRMAX-3) COR2060
IND=3*LOTTE+NRMAX+1 COR2070
DO 400 J=1,NVAR COR2080
I1=IARGS(J+1) COR2090
IJ=(J-1)*NVAR COR2100
DO 400 I=1,NVAR COR2110
IF (I.NE.J) GO TO 370 COR2120
I2=KURT+IJ+J COR2130
A(I2)=1. COR2140
GO TO 400 COR2150
370 I2=IARGS(I+1)-1 COR2160
DO 380 K=1,NRMAX COR2170
K1=K+I2 COR2180
K2=K+3*LOTTE COR2190
380 A(K2)=RC(K1) COR2200
I2=KURT+IJ+I COR2210
CALL BJORCK (RC(I1),A(3*LOTTE+1),NRMAX,A(IND),F) COR2220
IF (F.GT.0.) GO TO 390 COR2230
A(I2)=1. COR2240
GO TO 400 COR2250
390 CALL PROB (1.,Z,F,A(I2)) COR2260
400 CONTINUE COR2270
CALL MIST (NVAR,A(KURT+1),0,NLA,6) COR2280
C CONFIDENCE LIMITS FOR SIMPLE CORRELATION COEFFICIENT COR2290
F=FSQRT(FLOAT(NRMAX-3)) COR2300
HL1=2.5758293/F COR2310
HL2=1.9599640/F COR2320

```

```

A(3*LOTTE)=99.          COR2330
A(5*LOTTE)=95.          COR2340
DO 410 J=1,NVA          COR2350
IND=(J-1)*NVAR          COR2360
K1=IND+J+KURT           COR2370
K2=K1+KURT               COR2380
A(K1)=99.                COR2390
A(K2)=95.                COR2400
IJ=J+1                  COR2410
DO 410 I=IJ,NVAR         COR2420
C INDEX OF SCC           COR2430
I1=IND+I                COR2440
C INDICES OF UPPER,LOWER SCC CONFIDENCE LIMITS (99 PER CENT LEVEL) COR2450
K1=(I-1)*NVAR+J+KURT    COR2460
K2=K1+KURT               COR2470
Z=.5*FLOG((1.+A(I1))/(1.-A(I1))) COR2480
A(K1)=AMIN1(FTANH(Z+HL1),1.) COR2490
A(K2)=AMAX1(FTANH(Z-HL1),-1.) COR2500
C INDICES OF UPPER, LOWER SCC CONFIDENCE LIMITS (95 PER CENT LEVEL) COR2510
K1=I1+KURT               COR2520
K2=K1+KURT               COR2530
A(K1)=AMIN1(FTANH(Z+HL2),1.) COR2540
410 A(K2)=AMAX1(FTANH(Z-HL2),-1.) COR2550
CALL MIST (NVAR,A(KURT+1),0,NLA,7) COR2560
420 GO TO (470,450,430), KEEP      COR2570
C STORE SIMPLE AND PARTIAL CORRELATION COEFFICIENTS IN WORKSHEET COR2580
430 I1=MINO(NVAR,NCOL-(IARGS(NVAR+5)-1)/NROW) COR2590
I2=MINO(NVAR,NROW-(IARGS(NVAR+4)-IARGS(NVAR+5))) COR2600
IF (I1.LT.NVAR.OR.I2.LT.NVAR) CALL ERROR (213) COR2610
DO 440 J=1,I1             COR2620
DO 440 I=1,I2             COR2630
K1=(J-1)*NVAR+I+LOTTE    COR2640
K2=IARGS(NVAR+4)-1+(J-1)*NROW+I COR2650
440 RC(K2)=A(K1)           COR2660
450 I1=MINO(NVAR,NCOL-(IARGS(NVAR+3)-1)/NROW) COR2670
I2=MINO(NVAR,NROW-(IARGS(NVAR+2)-IARGS(NVAR+3))) COR2680
IF (I1.LT.NVAR.OR.I2.LT.NVAR) CALL ERROR (213) COR2690
DO 460 J=1,I1             COR2700
DO 460 I=1,I2             COR2710
K1=(J-1)*NVAR+I           COR2720
K2=IARGS(NVAR+2)-1+(J-1)*NROW+I COR2730
460 RC(K2)=A(K1)           COR2740
470 RETURN                 COR2750
C                           COR2760
480 FORMAT (1H ,32X,54HMATRIX IS SINGULAR. NO PARTIAL CORRELATIONS COCOR2770
1MPUTED.)                   COR2780
490 FORMAT (/30X,24HCORRELATION ANALYSIS FOR,I3,15H VARIABLES WITH,I5,COR2790
113H OBSERVATIONS)          COR2800
500 FORMAT (1H ,18X,83HNONLINEARITY TEST AND APPROXIMATION OF CONFIDENCOR2810
ICE INTERVALS NOT DEFINED FOR NRMAX =,I2) COR2820
END                         COR2830

```

```

SUBROUTINE CSPINV (A,M,KK,ISIG) CSP 10
C VERSION 5.00      CSPINV 5/15/70 CSP 20
C 7058MI MATRIX INVERSION WITH MINIMUM ROUNDOFF ERROR ACCUMULATION. CSP 30
COMMON/CONLB2/ER,ISIGD CSP 35
DATA ONE/1.0/,ZERO/0.0/ CSP 40
DIMENSION A(1) CSP 50
ISIG=0 CSP 60
N=M CSP 70
NN=KK CSP 80
N2=N+N CSP 90
DO 30 J=1,N CSP 100
NJCOL=(N+J-1)*NN CSP 110
DO 30 I=1,N CSP 120
KINJ=NJCOL+I CSP 130
IF (I-J) 10,20,10 CSP 140
10 A(KINJ)=ZERO CSP 150
GO TO 30 CSP 160
20 A(KINJ)=ONE CSP 170
30 CONTINUE CSP 180
C DETERMINE MAXIMUM ABS OF VARIABLE BEING ELIMINATED. THIS BECOMES CSP 190
L=0 CSP 200
40 L=L+1 CSP 210
LCOL=NN*L-NN CSP 220
KLL=LCOL+L CSP 230
IF (L-N) 50,100,200 CSP 240
C FIND THE LARGEST ELEMENT IN THE LTH COLUMN. CSP 250
50 J1=L CSP 260
C=ABS(A(KLL)) CSP 270
L1=L+1 CSP 280
DO 70 I=L1,N CSP 290
KIL=LCOL+I CSP 300
X=ABS(A(KIL)) CSP 310
IF (C-X) 60,70,70 CSP 320
C RECORD THE NUMBER OF THE ROW HAVING THE GREATER ELEMENT. CSP 330
60 J1=I CSP 340
C C BECOMES THE GREATER. CSP 350
C=X CSP 360
70 CONTINUE CSP 370
C INTERCHANGE ROW J1 WITH ROW L. J1 IS THE ROW WITH THE LARGEST ELEM CSP 380
C TEST TO SEE IF INTERCHANGING IS NECESSARY. CSP 390
IF (J1-L) 80,100,80 CSP 400
80 DO 90 J=L,N2 CSP 410
JCOL=NN*j-NN CSP 420
KJIJ=JCOL+J1 CSP 430
HOLD=A(KJIJ) CSP 440
KLJ=JCOL+L CSP 450
A(KJIJ)=A(KLJ) CSP 460
A(KLJ)=HOLD CSP 470
90 CONTINUE CSP 480
C IF THE LARGEST ABSOLUTE ELEMENT IN A COLUMN IS MACHINE ZERO WE CSP 490
C HAVE A SINGULAR MATRIX CSP 495
100 - IF (ABS(A(KLL))-ER) 110,110,120 CSP 500
110 ISIG=4 CSP 510
GO TO 200 CSP 520
C ZERO ALL THE ELEMENTS IN THE LTH COLUMN BUT THE PIVOTAL ELEMENT. CSP 530
120 L1=1 CSP 540
L2=L-1 CSP 550
IF (L2) 130,130,150 CSP 560
130 IF (L-N) 140,170,140 CSP 570

```

```

140 L1=L+1                               CSP 580
L2=N                                     CSP 590
150 DO 160 I=L1,L2                      CSP 600
KIL=LCOL+I                             CSP 610
Z=-A(KIL)/A(KLL)                      CSP 620
DO 160 J=L,N2                          CSP 630
JCOL=NN*N-J-NN                         CSP 640
KIJ=JCOL+I                            CSP 650
KLJ=JCOL+L                            CSP 660
160 A(KIJ)=A(KIJ)+Z*A(KLJ)            CSP 670
IF (N-L2) 40,40,130                    CSP 680
C DIVIDE BY DIAGONAL ELEMENTS.        CSP 690
170 DO 180 I=1,N                         CSP 700
KKK=NN*I-NN+I                         CSP 710
ZZ=A(KKK)                                CSP 720
DO 180 J=1,N2                          CSP 730
KKI=NN*N-J-NN+I                        CSP 740
180 A(KKI)=A(KKI)/ZZ                  CSP 750
C RETURN AFTER PUTTING A INVERSE INTO B CSP 760
DO 190 J=1,N                          CSP 770
JCOL=NN*N-J-NN                         CSP 780
NJCOL=NN*N+JCOL                        CSP 790
DO 190 I=1,N                          CSP 800
KIJ=JCOL+I                            CSP 810
KINJ=NJCOL+I                           CSP 820
190 A(KIJ)=A(KINJ)                   CSP 830
200 RETURN                                CSP 840
END                                     CSP 850

```

```

FUNCTION DBEJ (X,N,M)
C VERSION 5.00      DBEJ      5/15/70          DBE  10
      DOUBLE PRECISION DBEJ,E,H,X,A,B,C,D,Y,S(120),T(120)    DBE  20
      DOUBLE PRECISION FDSIN,FDCOS,FDLOG,FDSQRT,FDEXP        DBE  30
      COMMON /SCRAT/ NS,NS2,SCRAT(13500)                      DBE  40
      EQUIVALENCE (SCRAT(1201),S), (SCRAT(1451),T)           DBE  50
      DOUBLE PRECISION DSNCOS,DXEXP                          DBE  60
      COMMON /DCONLB/ DSNCOS,DXEXP                          DBE  70
10   IF (DABS(X).GT.16.5) GO TO 90                         DBE  80
      A=(X/2.D0)**2                                         DBE  90
      J=X/.4+6.8                                           DBE 100
      B=J                                                 DBE 110
      C=J+N                                              DBE 120
      D=-1.D0                                             DBE 130
      IF (M.GT.1) D=1.D0                                     DBE 140
      IF (M.GT.3) GO TO 30                                     DBE 150
      Y=1.D0                                              DBE 160
      DO 20 I=1,J                                         DBE 170
      Y=1.D0+Y*A/(B*C)*D                                    DBE 180
      B=B-1.D0                                            DBE 190
20   C=C-1.D0                                           DBE 200
      IF (N.GT.0) Y=X*Y/2.D0                                DBE 210
      GO TO 200                                         DBE 220
30   E=1.0D0                                           DBE 230
      S(1)=.5772156649015D0                               DBE 240
      S(61)=S(1)-.5D0                                      DBE 250
      DO 40 I=2,60                                         DBE 260
      S(I)=S(I-1)-1.D0/E                                 DBE 270
      S(I+60)=S(I)-1.D0/(2.D0*(E+1.D0))                DBE 280
40   E=E+1.D0                                           DBE 290
      E=FDLOG(X/2.D0)                                     DBE 300
      DO 50 I=1,120                                       DBE 310
50   T(I)=S(I)+E                                       DBE 320
      IF (M.LT.6) GO TO 60                                DBE 330
      IF (X.GT.8.D0) GO TO 90                            DBE 340
      IF (X.GT.8.D0) GO TO 90                            DBE 350
60   IA=0                                              DBE 360
      IF (N.GT.0) IA=60                                    DBE 370
      IF (M.GT.5) D=-1.D0                                DBE 380
      I=J+IA+1                                         DBE 390
      Y=T(I)                                             DBE 400
      DO 70 IB=1,J                                       DBE 410
      I=J-IB+IA+1                                      DBE 420
      Y=T(I)-D*A*Y/(B*C)                                DBE 430
      B=B-1.D0                                         DBE 440
70   C=C-1.D0                                           DBE 450
      IF (N.GT.0) Y=X*Y/2.D0                                DBE 460
      IF (M.GT.5) GO TO 80                                DBE 470
      Y=Y*.636619772368D0                                DBE 480
      IF (N.NE.0) Y=-.636619772368D0/X+Y                 DBE 490
      GO TO 200                                         DBE 500
80   Y=-Y                                              DBE 510
      IF (N.NE.0) Y=1.D0/X-Y                            DBE 520
      GO TO 200                                         DBE 530
90   A=8.D0*X                                         DBE 540
      H=N                                              DBE 550
      H=(2.*H)**2                                      DBE 560
      T(1)=(H-1.D0)/A                                  DBE 570
      D=T(1)                                           DBE 580
      DO 100 I=2,20                                     DBE 590

```

```

K=I                                     DBE 600
B=I                                     DBE 610
C=(2*I-1)**2                           DBE 620
T(I)=(H-C)/(A*B)                      DBE 630
E=D                                     DBE 640
D=T(I)*D                               DBE 650
E=DABS(D/E)                            DBE 660
IF (DABS(D).LT..5D-10) GO TO 110      DBE 670
IF (E.GT..91D0) GO TO 110              DBE 680
100 T(I+2)=0.0D0                         DBE 690
110 A=-1.D0                             DBE 700
    IF (M.LE.1) GO TO 150               DBE 710
    IF (M.LE.3) GO TO 120               DBE 720
    IF (M.LE.5) GO TO 150               DBE 730
    A=1.D0                             DBE 740
120 Y=1.D0                             DBE 750
    DO 130 I=1,K
    J=K-I+1
130 Y=1.D0+A*Y*T(J)                   DBE 780
    A=1.D0
    IF (X.LT.DXEXP) A=FDEXP(X)        DBE 800
    IF (M.LE.5) GO TO 140              DBE 810
    Y=Y/(A*FDSQRT(.636619772368D0*X))
    GO TO 200                          DBE 830
140 Y=Y*A/FDSQRT(6.283185307D0*X)    DBE 840
    GO TO 200                          DBE 850
150 Y=FDSQRT(3.1415926536D0*X)       DBE 860
    J=K/2
    K=2*J
    J=J-1
    A=1.D0
    H=A
    DO 160 I=1,J
    IA=K-2*I+1
    A=1.D0-A*T(IA)*T(IA+1)
160 H=1.D0-H*T(IA)*T(IA-1)
    A=(1.D0-T(1)*T(2)*A)/Y
    H=T(1)*H/Y
    B=FDSIN(X)
    C=FDCOS(X)
    D=A-H
    E=A+H
    IF (M.GT.2) GO TO 180
    IF (N.EQ.0) GO TO 170
    Y=E*B-D*C
    GO TO 200
170 Y=D*B+E*C
    GO TO 200
180 IF (N.EQ.0) GO TO 190
    Y=-D*B-E*C
    GO TO 200
190 Y=E*B-D*C
    GO TO 200
200 DBEJ=Y
    RETURN
    END

```

```

SUBROUTINE DEFINE                               DEF 10
C   VERSION 5.00      DEFINE      5/15/70        DEF 20
COMMON /BLOCRC/ NRC,RC(12600)                 DEF 30
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NDEF 40
IARGS,VWXYZ(8),NERROR                         DEF 50
DIMENSION ARGS(100)                           DEF 60
EQUIVALENCE (ARGS(1),RC(12501))              DEF 70
C                                         DEF 80
C   DEFINE $$ INTO COLUMN ++
C   DEFINE $$ INTO ROW ++, COL ++.             DEF 90
C   DEFINE ROW ++, COL ++ INTO ROW ++, COL ++. DEF 100
C   DEFINE ROW ++, COL ++ INTO COL ++.         DEF 110
C                                         DEF 120
C   IF(NARGS.NE.2) GO TO 5                     DEF 122
J=1                                         DEF 124
IF(KIND(1).EQ.0) CALL ADRESS(1,J)            DEF 125
CALL ADRESS (2,I)                           DEF 126
IF(I) 120,130,2                            DEF 127
2   IF(J) 120,130,60                          DEF 128
5   IF(NARGS.NE.3) IF(NARGS-4) 115,40,115       DEF 129
IF (KIND(1).EQ.0) GO TO 40                   DEF 130
10  I=NARGS                                DEF 140
GO TO 90                                  DEF 150
20  IF (NERROR.EQ.0) RC(L)=ARGS(1)           DEF 160
30  RETURN                                 DEF 170
40  I=2                                    DEF 180
GO TO 90                                  DEF 190
50  ARGS(1)=RC(L)                           DEF 200
IF (NARGS.EQ.4) GO TO 10                   DEF 210
CALL ADRESS (3,I)                           DEF 220
IF (I) 120,130,60                          DEF 230
60  IF (NERROR.NE.0) GO TO 30               DEF 240
IF (NRMAX.EQ.0) GO TO 70                   DEF 250
IF(KIND(1).EQ.0.AND.NARGS.EQ.2) GO TO 140    DEF 255
CALL VECTOR (ARGS(1),I)                     DEF 260
GO TO 30                                  DEF 270
70  I=9                                    DEF 280
80  CALL ERROR (I)                          DEF 290
GO TO 30                                  DEF 300
C                                         DEF 310
C   CHECK AND CALCULATE WORKSHEET ENTRY LOCATION INTO L  DEF 320
C                                         DEF 330
90  CALL ADRESS (I,L)                      DEF 340
IF (L) 120,130,100                         DEF 350
100 IF (KIND(I-1).EQ.0.AND.IARGS(I-1).GT.0.AND.IARGS(I-1).LE.NROW) GO  DEF 360
    1TO 110                                DEF 370
    I=16                                    DEF 380
    GO TO 80                                DEF 390
110 L=L+IARGS(I-1)-1                      DEF 400
    IF (I-2) 50,50,20                      DEF 410
115 I=10                                    DEF 420
    GO TO 80                                DEF 430
120 I=20                                    DEF 440
    GO TO 80                                DEF 450
130 I=11                                    DEF 460
    GO TO 80                                DEF 470
140 DO 150 IJ=1,NRMAX                      DEF 480
    RC(I)=RC(J)                           DEF 490
    I=I+1                                    DEF 500
150 J=J+1                                    DEF 510
    RETURN                                 DEF 520
    END                                   DEF 530

```

```

C SUBROUTINE DETRNK (A,NROW,N,DET,RANK)          DET    10
C VERSION 5.00      DETRNK      5/15/70        DET    20
C WRITTEN BY S PEAVY   2/ 1/68        DET    30
C EVALUATES THE DETERMINANT AND RANK OF A        DET    40
C A LOCATION OF MATRIX        DET    50
C NROW THE DIMENSION OF A IN DIMENSION STATEMENT DET    60
C N SIZE OF A        DET    70
C DET -THE VALUE OR DETERMINANT OF A        DET    80
C RANK-RANK OF A        DET    90
C ORIGANAL VALUES OF A ARE DESTROYED        DET   100
C                                         DET   110
C
C DIMENSION A(NROW,NROW)        DET   120
C NK=N        DET   130
C NN=N-1        DET   140
C VAL=1.0        DET   150
C DO 60 I=1,NN        DET   160
C II=I+1        DET   170
C IB=I        DET   180
C DO 10 J=II,N        DET   190
C IF (ABS(A(IB,I)).GE.ABS(A(J,I))) GO TO 10        DET   200
C IB=J        DET   210
10  CONTINUE        DET   220
C IF (ABS(A(IB,I)).GE.1.E-7) GO TO 20        DET   230
C NK=NK-1        DET   240
C GO TO 40        DET   250
20  IF (IB.EQ.I) GO TO 40        DET   260
C DO 30 J=I,N        DET   270
C T=A(I,J)        DET   280
C A(I,J)=A(IB,J)        DET   290
30  A(IB,J)=T        DET   300
C VAL=-VAL        DET   310
40  DO 50 J=II,N        DET   320
C X=A(J,I)/A(I,I)        DET   330
C DO 50 K=I,N        DET   340
50  A(J,K)=A(J,K)-X*A(I,K)        DET   350
60  CONTINUE        DET   360
C RANK=NK        DET   370
C IF (NK-N) 90,70,90        DET   380
70  PR=1.0        DET   390
C DO 80 I=1,N        DET   400
80  PR=PR*A(I,I)        DET   410
C DET=VAL*PR        DET   420
C RETURN        DET   430
90  DET=0.0        DET   440
C RETURN        DET   450
C END        DET   460

```

```

C      SUBROUTINE DHRND (X,N,XT)          DHR   10
C      VERSION 5.00      DHRND      5/15/70    DHR   20
C      SUBROUTINE TO ROUND X TO N SD AND STORE IN XT    DHR   30
C      WRITTEN BY DAVID HOGBEN, SEL, NBS. 4/09/70.    DHR   40
C      DOUBLE PRECISION Z    DHR   50
C      IF (X) 20,10,20    DHR   60
10     XT=0.0    DHR   70
      RETURN    DHR   80
20     IF (N.LT.1) N=1    DHR  100
      IF (N.GT.8) N=8    DHR  110
      Y=ABS(X)    DHR  120
      M=FLOG10(Y)    DHR  130
      IF (Y.LT.1.0) M=M-1    DHR  140
      Z=Y    DHR  150
      Z=Z*10.D0** (8-M)    DHR  160
      IF (Z.LT.1.0D+9) GO TO 30    DHR  170
      M=M+1    DHR  180
      Z=Z/10.0D0    DHR  190
      GO TO 40    DHR  200
30     IF (Z.GE.1.0D+8) GO TO 40    DHR  210
      M=M-1    DHR  220
      Z=10.0D0*Z    DHR  230
40     X1=Z    DHR  240
      LL1=X1    DHR  250
      X2=Z-DBLE(X1)    DHR  260
      LL2=X2    DHR  270
      LL=LL1+LL2+5    DHR  280
      LL1=LL/(10** (9-N))    DHR  290
      LL2=LL1*10** (9-N)    DHR  300
      LL2=LL-LL2    DHR  310
      IF (N.EQ.8) GO TO 70    DHR  320
      IF(LL2/10-5*10** (7-N)) 70,50,60    DHR  330
50     LL2=MOD(LL1,2)    DHR  340
      IF (LL2) 70,70,60    DHR  350
60     LL1=LL1+1    DHR  360
70     XT=FLOAT(LL1)    DHR  370
      IF (M.EQ.N-1) GO TO 80    DHR  380
C
      Z=XT    DHR  390
      Z=Z*10.0D0** (M-N+1)    DHR  400
      XT=FDPCON(Z)    DHR  410
80     XT=SIGN(XT,X)    DHR  420
      RETURN    DHR  430
      END    DHR  440

```

	SUBROUTINE DIMENS	DIM	10
C	VERSION 5.00 DIMENS 5/15/70	DIM	20
	COMMON /BLOCRC/ NRC,RC(12600)	DIM	30
	COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NDIM	DIM	40
	IARGS,VWXYZ(8),NERROR	DIM	50
	DIMENSION ARGS(100)	DIM	60
	EQUIVALENCE (ARGS(1),RC(12501))	DIM	70
	IF (NARGS.EQ.2) IF (KIND(1)+KIND(2)) 25,27,25	DIM	75
	K=10	DIM	80
10	CALL ERROR (K)	DIM	90
20	RETURN	DIM	100
25	K=20	DIM	110
	GO TO 10	DIM	120
27	IF (IARGS(1).GT.0.AND.IARGS(2).GT.0.AND.IARGS(1)*IARGS(2).LE.NRC)	DIM	130
	1GO TO 30	DIM	140
	K=3	DIM	150
	GO TO 10	DIM	160
30	NROW=IARGS(1)	DIM	170
	NCOL=IARGS(2)	DIM	180
	NRMAX=MIN0(NROW, NRMAX)	DIM	190
	GO TO 20	DIM	200
	END	DIM	210

SUBROUTINE DUMMYA DMA 10  
C VERSION 5.00 DUMMYA 5/15/70 DMA 20  
PRINT 10 DMA 30  
10 FORMAT(55H \* OMNITAB COMMAND DUMMYA IS NOT AVAILABLE AT THIS TIME) DMA 40  
RETURN DMA 50  
END DMA 60

SUBROUTINE DUMMYB DMB 10  
C VERSION 5.00 DUMMYB 5/15/70 DMB 20  
PRINT 10 DMB 30  
10 FORMAT(55H \* OMNITAB COMMAND DUMMYB IS NOT AVAILABLE AT THIS TIME) DMB 40  
RETURN DMB 50  
END DMB 60

SUBROUTINE DUMMYC DMC 10  
C VERSION 5.00 DUMMYC 5/15/70 DMC 20  
PRINT 10 DMC 30  
10 FORMAT(55H \* OMNITAB COMMAND DUMMYC IS NOT AVAILABLE AT THIS TIME) DMC 40  
RETURN DMC 50  
END DMC 60

SUBROUTINE DUMMYD DMD 10  
C VERSION 5.00 DUMMYD 5/15/70 DMD 20  
PRINT 10 DMD 30  
10 FORMAT(55H \* OMNITAB COMMAND DUMMYD IS NOT AVAILABLE AT THIS TIME) DMD 40  
RETURN DMD 50  
END DMD 60

SUBROUTINE DUMMYE DME 10  
C VERSION 5.00 DUMMYE 5/15/70 DME 20  
PRINT 10 DME 30  
10 FORMAT(55H \* OMNITAB COMMAND DUMMYE IS NOT AVAILABLE AT THIS TIME) DME 40  
RETURN DME 50  
END DME 60

SUBROUTINE DUMMYF DMF 10  
C VERSION 5.00 DUMMYF 5/15/70 DMF 20  
PRINT 10 DMF 30  
10 FORMAT(55H \* OMNITAB COMMAND DUMMYF IS NOT AVAILABLE AT THIS TIME) DMF 40  
RETURN DMF 50  
END DMF 60

```

C SUBROUTINE ERASE          ERA 10
C VERSION 5.00      ERASE      5/15/70    ERA 20
C                                         ERA 30
C ERASE COL (C), (C), (C), ETC.    ERA 40
C IF NO COLS SPECIFIED ALL OF WORKSHEET IS ERASED    ERA 45
C                                         ERA 50
C COMMON /BLOCRC/ NRC,RC(12600)    ERA 60
C COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NERA 70
1ARGS,VWXYZ(8),NERROR
DIMENSION ARGS(100)
EQUIVALENCE (ARGS(1),RC(12501))
IF (NARGS.EQ.0) GO TO 40
CALL CHKCOL (I)
IF (I.EQ.0) GO TO 20
I=20
CALL ERROR (I)
10 RETURN
20 IF (NERROR.NE.0.OR.NRMAX.EQ.0) GO TO 10
DO 30 I=1,NARGS
30 CALL VECTOR (0.,IARGS(I))
GO TO 10
C
C CLEAR ALL OF DIMENSIONED WORKSHEET.
C
40 IF (NERROR.NE.0) GO TO 10
NRMAX=NROW*NCOL
CALL VECTOR (0.,1)
NRMAX=0
GO TO 10
END

```

```

SUBROUTINE ERRINT (X,ERF,ERFC)                                ERT 10
C   VERSION 5.00      ERRINT  5/15/70                           ERT 20
    DOUBLE PRECISION AN,BN,CONS,C1,DN,ERF,ERFC,F,FN,FNM1,FNM2,FOUR,GN,ERT 25
1GNM1,GNM2,ONE,P,PREV,RNBC,SCF,SUM,TN,TOLER,TRRTPI,TWO,ULCF,ULPS,WNERT 30
2,X,Y,YSQ,FDEXP                                         ERT 40
COMMON/DCONL2/TRRTPI,NBC,NBM                               ERT 90
DATA ONE,TWO,FOUR,ULPS,CONS/1.D0,2.D0,4.D0,1.D0,.83D0/     ERT 100
RNBC=NBC                                                 ERT 120
TOLER=TWO**(-NBM)                                         ERT 130
IF (X) 20,10,20                                           ERT 140
10  ERF=0                                                 ERT 150
    ERFC=ONE                                              ERT 160
    RETURN                                                ERT 170
20  Y=ABS(X)                                              ERT 180
    YSQ=Y**2                                              ERT 190
    IF (Y-ULPS) 50,50,30                                 ERT 200
30  C1=TWO**((RNBC-ONE)/TWO)                            ERT 210
    ULCF=CONS*C1                                         ERT 220
    SCF=TWO**((C1**2-RNBC)                               ERT 230
    IF (Y-ULCF) 110,110,40                                 ERT 240
40  ERF=ONE                                              ERT 250
    ERFC=0                                                 ERT 260
    GO TO 80                                             ERT 270
50  SUM=0                                                 ERT 280
    DN=ONE                                              ERT 290
    TN=ONE                                              ERT 300
    P=TWO*YSQ                                         ERT 310
60  DN=DN+TWO                                         ERT 320
    TN=P*TN/DN                                         ERT 330
    SUM=TN+SUM                                         ERT 340
    IF (TN-TOLER) 70,60,60                               ERT 350
70  ERF=(SUM+ONE)*TRRTPI*Y*FDEXP(-YSQ)                 ERT 360
    ERFC=ONE-ERF                                         ERT 370
80  IF (X) 90,100,100                                 ERT 380
90  ERF=-ERF                                            ERT 390
    ERFC=TWO-ERFC                                         ERT 400
100 RETURN                                               ERT 410
110 FNM2=0                                              ERT 420
    GNM2=ONE                                         ERT 430
    FNM1=TWO*Y                                         ERT 440
    GNM1=TWO*YSQ+ONE                                  ERT 450
    PREV=FNM1/GNM1                                    ERT 460
    WN=ONE                                              ERT 470
    BN=GNM1+FOUR                                     ERT 480
120 AN=-WN*(WN+ONE)                                 ERT 490
    FN=BN*FNM1+AN*FNM2                               ERT 500
    GN=BN*GNM1+AN*GNM2                               ERT 510
    F=FN/GN                                         ERT 520
    IF (ABS(ONE-(F/PREV))-TOLER) 170,170,130       ERT 530
130 IF (PREV-F) 140,140,160                           ERT 540
140 IF (GN.LT.SCF) GO TO 150                         ERT 550
    FN=FN/SCF                                         ERT 560
    GN=GN/SCF                                         ERT 570
    FNM1=FNM1/SCF                                    ERT 580
    GNM1=GNM1/SCF                                    ERT 590
150 FNM2=FNM1                                         ERT 600
    GNM2=GNM1                                         ERT 610
    FNM1=FN                                           ERT 620
    GNM1=GN                                           ERT 630

```

	WN=WN+TWO	ERT 640
	BN=BN+FOUR	ERT 650
	PREV=F	ERT 660
	GO TO 120	ERT 670
160	F=PREV	ERT 680
170	ERFC=F*FDEXP (-YSQ)*TRRTPI /TWO	ERT 690
	ERF=ONE-ERFC	ERT 700
	GO TO 80	ERT 710
	END	ERT 720

```

SUBROUTINE ERROR(I)                                ERR 10
C      VERSION 5.00      ERROR      5/15/70          ERR 20
COMMON/BLOCRC/NRC,RC(12600)                      ERR 30
COMMON/BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,
1 NROW,NCOL,NARGS,VWXYZ(8),NERROR                ERR 40
DIMENSION ARGS(100)                               ERR 50
EQUIVALENCE (ARGS(1),RC(12501))                 ERR 60
COMMON/BLOCKC/KIO,INUNIT,ISCRAT,KBDOUT,KRDKN,LLIST
COMMON / BLOCKX / INDEX( 6, 8 ), LEVEL           ERR 70
COMMON/SPRV/NERCON,NERR,ISWERR                  ERR 80
DIMENSION IL( 2 )                                ERR 90
DATA IBL,IL(1),IL(2)/2H ,2H/I,2H/F/
ISCRUN=ISCRAT                                     ERR 100
C
C      IF    1 .LE. I .LE. 100, FATAL ERROR        ERR 110
C      IF 101 .LE. I .LE. 200, ARITHMETIC ERROR     ERR 120
C      IF 201 .LE. I .LE.      INFORMATIVE DIAGNOSTIC
C
C      NERR=NERR+1                                 ERR 130
7003 IF( I .GT. 100 ) GO TO 200                  ERR 140
NERROR=NERROR+1                                  ERR 150
WRITE( ISCRUN, 800 )                            ERR 160
800 FORMAT(/32H*** FATAL ERROR IN ABOVE COMMAND,52X)
IF(LLIST.NE.0) GO TO 710                      ERR 170
WRITE(ISCRUN,700)                                ERR 180
700 FORMAT(66H*** COMMAND WAS NOT LISTED BECAUSE NO LIST OR LIST 0 WASERR 260
1 IN EFFECT,18X)                               ERR 190
710 LLIST=3                                     ERR 200
      GO TO (801,802,803,804,805,806,807,808,809,810,811,812,813,814,815ERR 290
1,816,817,818,819,820,821,822,823,824,825,826,827,828,829,830), I ERR 300
801 WRITE(ISCRUN,1)                            ERR 310
1 FORMAT(29H*** NAME NOT FOUND IN LIBRARY,55X)  ERR 320
GO TO 900                                      ERR 330
802 WRITE(ISCRUN,2)                            ERR 340
2 FORMAT(28H*** ILLEGAL STATEMENT NUMBER,56X)  ERR 350
GO TO 900                                      ERR 360
803 WRITE(ISCRUN,3)                            ERR 370
3 FORMAT(28H*** ILLEGAL ARGUMENT ON CARD,56X)  ERR 380
GO TO 900                                      ERR 390
804 GO TO 900                                  ERR 400
805 WRITE(ISCRUN,5)                            ERR 410
5 FORMAT(38H*** COMMAND NOT ALLOWED IN REPEAT MODE,46X)  ERR 420
GO TO 900                                      ERR 430
806 WRITE(ISCRUN,6)                            ERR 440
6 FORMAT(74H*** STATEMENT NUMBER MAY NOT BEGIN ANY CARD BETWEEN BEGIERR 450
1N AND FINISH CARDS,10X)                      ERR 460
GO TO 900                                      ERR 470
807 WRITE(ISCRUN,7)                            ERR 480
7 FORMAT(23H*** ILLEGAL *STATEMENT*,61X)       ERR 490
GO TO 900                                      ERR 500
808 WRITE(ISCRUN,8)                            ERR 510
8 FORMAT(34H*** PHYSICAL CONSTANT NOT IN TABLE,50X)  ERR 520
GO TO 900                                      ERR 530
809 WRITE(ISCRUN,9)                            ERR 540
9 FORMAT(13H*** NRMAX = 0,71X)                 ERR 550
GO TO 900                                      ERR 560
810 WRITE(ISCRUN,10)   NARGS                  ERR 570
10 FORMAT(3H***,I4,34H IS AN ILLEGAL NUMBER OF ARGUMENTS,43X)  ERR 580
GO TO 900                                      ERR 590

```

811	WRITE(ISCRUN,11)	ERR 600
11	FORMAT(40H*** COLUMN NUMBER TOO BIG OR LESS THAN 1,44X)	ERR 610
	GO TO 900	ERR 620
812	WRITE(ISCRUN,12)	ERR 630
12	FORMAT(33H*** COMMAND STORAGE AREA OVERFLOW,51X)	ERR 640
	GO TO 900	ERR 650
813	WRITE(ISCRUN,13)	ERR 660
13	FORMAT(30H*** STATEMENT NUMBER NOT FOUND,54X)	ERR 670
	GO TO 900	ERR 680
814	WRITE(ISCRUN,14)	ERR 690
14	FORMAT(35H*** ILLEGAL OR NO FORMAT DESIGNATOR,49X)	ERR 700
	GO TO 900	ERR 710
815	WRITE(ISCRUN,15)	ERR 720
15	FORMAT(34H*** DIMENSIONED AREA EXCEEDS LIMIT,50X)	ERR 730
	GO TO 900	ERR 740
816	WRITE(ISCRUN,16)	ERR 750
16	FORMAT(27H*** ILLEGAL SIZE ROW NUMBER,57X)	ERR 760
	GO TO 900	ERR 770
817	WRITE(ISCRUN,17)	ERR 780
17	FORMAT(39H*** DEFINED MATRIX OVERFLOWS WORKSHEET,45X)	ERR 790
	GO TO 900	ERR 800
818	WRITE(ISCRUN,18)	ERR 810
18	FORMAT(36H*** INTEGER ARGUMENT LESS THAN -8191,48X)	ERR 820
	GO TO 900	ERR 830
819	WRITE(ISCRUN,19)	ERR 840
19	FORMAT(48H*** STORED PERFORM STATEMENT WILL EXECUTE ITSELF,36X)	ERR 850
	GO TO 900	ERR 860
820	WRITE(ISCRUN,20)	ERR 870
20	FORMAT(29H*** IMPROPER TYPE OF ARGUMENT,55X)	ERR 880
	GO TO 900	ERR 890
821	WRITE(ISCRUN,21)	ERR 900
21	FORMAT(26H*** COMMAND MUST BE STORED,58X)	ERR 910
	GO TO 900	ERR 920
822	WRITE(ISCRUN,22)	ERR 930
22	FORMAT(31H*** MATRIX IS (NEARLY) SINGULAR,53X)	ERR 940
	GO TO 900	ERR 950
823	WRITE(ISCRUN,23)	ERR 960
23	FORMAT(28H***INSUFFICIENT SCRATCH AREA,56X)	ERR 970
	GO TO 900	ERR 980
824	WRITE(ISCRUN,24)	ERR 990
24	FORMAT (49H*** DEGREE IS LARGER THAN NO. OF NON-ZERO WEIGHTS,35X)	ERR1000
	GO TO 900	ERR1010
825	WRITE(ISCRUN,25)	ERR1020
25	FORMAT(35H***NEGATIVE WEIGHTS MAY NOT BE USED,49X)	ERR1030
	GO TO 900	ERR1040
826	WRITE(ISCRUN,26)	ERR1050
26	FORMAT(51H***NUMBER OF COLUMNS IS GREATER THAN NUMBER OF ROWS,33X)	ERR1060
	GO TO 900	ERR1070
827	WRITE (ISCRUN,27)	ERR1080
27	FORMAT (19H***FORMAT NOT FOUND,65X)	ERR1090
	GO TO 900	ERR1100
C****	THE FOLLOWING CARDS ARE NEEDED ONLY FOR TAPE OPERATIONS	ERR1110
828	WRITE(ISCRUN,28)	ERR1120
28	FORMAT(47H* INCORRECT TAPE UNIT. COMMAND IS NOT EXECUTED.,37X)	ERR1130
	GO TO 900	ERR1140
C*****	*****	ERR1150
829	NSB = NARGS+1	ERR1160
	WRITE (ISCRUN,29) NSB	ERR1170
29	FORMAT (31H* NUMBER OF ARGUMENTS SHOULD BE,I2,51X)	ERR1180

830	GO TO 900	ERR1185
	WRITE(ISCRUN,30)	ERR1190
30	FORMAT(48H* AN INCREMENT COMMAND CAN NOT INCREMENT ITSELF.,36X)	ERR1200
900	IF( LEVEL .NE. 0 ) CALL RNDOWN	ERR1220
C	FORCE OUT OF REPEAT MODE IF FATAL ERROR	ERR1230
	IF( I .LE. 100 ) LEVEL = 0	ERR1240
	WRITE( ISCRUN, 901 )	ERR1250
901	FORMAT(84X)	ERR1260
	RETURN	ERR1270
200	IF(NERR.LE.NERCON.OR.LLIST.NE.3) GO TO 201	ERR1280
	IF(ISWERR.NE.0) RETURN	ERR1290
	ISWERR=1	ERR1300
	WRITE( ISCRUN, 9999)NERCON	ERR1310
9999	FORMAT(/1H*,I5,62H INFORMATIVE AND ARITHMETIC DIAGNOSTICS HAVE BEEN ENCOUNTERED.,16X/ 284H* ANY SUCH ADDITIONAL DIAGNOSTICS FOR THIS COMMAND OR REPEAT MODE ARE DISREGARDED. )	ERR1320 ERR1330 MERR1340 ERR1350
	RETURN	ERR1360
201	IF(I.GT.200) GO TO 400	ERR1370
C		ERR1380
C		ERR1390
C	ARITHMETIC TROUBLES, SET FLAGS	ERR1400
C		ERR1410
	CALL AERR(I-100)	ERR1420
250	RETURN	ERR1430
C		ERR1440
C	INFORMATIVE DIAGNOSTIC	ERR1450
C		ERR1460
400	IF( MOD( LLIST, 2 ) .EQ. 0 ) GO TO 250	ERR1470
	IF(LLIST.EQ.0) GO TO 250	ERR1475
	CALL INFERR(I)	ERR1480
	GO TO 900	ERR1490
	END	ERR1500

```

SUBROUTINE EXCHNG          EXC  10
C   VERSION 5.00      EXCHNG      5/15/70      EXC  20
C                                         EXC  30
C                                         EXC  40
C                                         EXC  50
C                                         EXC  60
C                                         EXC  70
C                                         EXC  80
C                                         EXC  90
C                                         EXC 100
C                                         EXC 110
C                                         EXC 120
C                                         EXC 130
C                                         EXC 140
C                                         EXC 150
C                                         EXC 160
C                                         EXC 170
C                                         EXC 180
C                                         EXC 190
C                                         EXC 200
C                                         EXC 210
C                                         EXC 220
C                                         EXC 230
C                                         EXC 240
C                                         EXC 250
C                                         EXC 260
C                                         EXC 270
C                                         EXC 280
C                                         EXC 290
C                                         EXC 300
C                                         EXC 310
C                                         EXC 320
C                                         EXC 330
C                                         EXC 340
C
C   EXCHANGE COL ++ WITH ++, COL ++ WITH ++, ETC.
C
C   COMMON /BLOCRC/ NRC,RC(12600)
C   COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NEXC
10    IARGS,VWXYZ(8),NERROR
C   DIMENSION ARGS(100)
C   EQUIVALENCE (ARGS(1),RC(12501))
C   IF (NARGS) 70,70,10
C   IF (NARGS.NE.(NARGS/2)*2) GO TO 70
C   DO 50 I=1,NARGS,2
C   CALL ADRESS (I,J)
C   IF (J) 60,80,20
C   CALL ADRESS (I+1,K)
C   IF (K) 60,80,30
C   IF (NERROR.NE.0) RETURN
C   DO 40 N=1,NRMAX
C   JJ=J+N-1
C   KK=K+N-1
C   WORK=RC(JJ)
C   RC(JJ)=RC(KK)
C   RC(KK)=WORK
C   CONTINUE
C   CONTINUE
C   GO TO 90
C   CALL ERROR (3)
C   GO TO 90
C   CALL ERROR (10)
C   GO TO 90
C   CALL ERROR (11)
C   RETURN
C   END

```

```

SUBROUTINE EXPAND (J,WHERE) EXD 10
C VERSION 5.00 EXPAND 5/15/70 EXD 20
COMMON /BLOCRC/ NRC,RC(12600) EXD 30
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NEXTD 40
IARGS,VWXYZ(8),NERROR EXD 50
DIMENSION ARGS(100) EXD 60
EQUIVALENCE (ARGS(1),RC(12501)) EXD 70
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG EXD 80
DIMENSION WHERE(1) EXD 90
C EXD 100
C THIS ROUTINE EXPANDS STORED COMMANDS FROM WHERE TO A USABLE EXD 110
C FORM IN ARGS, IARGS AND KIND. EXD 120
C EXD 130
II=0 EXD 140
I=0 EXD 150
JJJ=J EXD 160
C CONVERT ONLY FIRST ARGUMENT IF COMMAND IS INCREMENT OR RESTORE EXD 170
IF (L1.NE.14) GO TO 10 EXD 180
IF (L2.GE.6.AND.L2.LE.8) JJJ=2 EXD 190
10 II=II+1 EXD 200
20 I=I+1 EXD 210
IF (I.GE.JJJ) GO TO 80 EXD 220
T=WHERE(I) EXD 230
IF (T) 50,40,30 EXD 240
30 KIND(II)=0 EXD 250
IARGS(II)=T-8192. EXD 260
GO TO 10 EXD 270
40 KIND(II)=1 EXD 280
I=I+1 EXD 290
ARGS(II)=WHERE(I) EXD 300
GO TO 10 EXD 310
50 IF(T.EQ.(-1)) GO TO 100 EXD 320
CALL XPND (WHERE(I),K,ARGS(II),KND) EXD 330
IF (K.GE.0) GO TO 90 EXD 340
60 K=-K EXD 350
70 CALL ERROR (K) EXD 360
80 RETURN EXD 370
90 KIND(II)=KND EXD 380
IF (KND.EQ.0) IARGS(II)=ARGS(II) EXD 390
I=I+K EXD 400
GO TO 10 EXD 410
C EXD 420
C IF STORED VALUE = -1, THEN ARGS (INTEGER) ARE TO BE EXPANDED FROM EXD 430
C PREVIOUS ARG TO FOLLOWING WITH A MAXIMUM TOTAL OF 100 EXD 440
C EXD 450
100 I=I+1 EXD 460
C PICK UP NEXT ARG EXD 470
IU=WHERE(I) EXD 480
IF (KIND(II-1).NE.0.OR.I.GE.J) GO TO 190 EXD 490
IF (IU) 170,190,110 EXD 500
110 IU=IU-8192 EXD 510
120 K=IU-IARGS(II-1) EXD 520
NARGS=NARGS+IABS(K)-1 EXD 530
IF (NARGS.GT.100) GO TO 200 EXD 540
IF (K) 130,20,140 EXD 550
130 INC=-1 EXD 560
K=-K EXD 570
GO TO 150 EXD 580
140 INC=1 EXD 590

```

```

150 DO 160 IT=1,K EXD 600
      KIND(II)=0 EXD 610
      IARGS(II)=IARGS(II-1)+INC EXD 620
160 II=II+1 EXD 630
      GO TO 20 EXD 640
C EXD 650
C EXPAND FORM IARG *** ,,ARG,, EXD 660
C EXD 670
170 CALL XPND (WHERE(I),K,ARGS(II),KND) EXD 680
      IF (K.LT.0) GO TO 60 EXD 690
      I=I+K EXD 700
      IF (KND.EQ.0) GO TO 180 EXD 710
      K=20 EXD 720
      GO TO 70 EXD 730
180 IU=ARGS(II) EXD 740
      GO TO 120 EXD 750
190 CALL ERROR (211) EXD 760
      GO TO 10 EXD 770
200 K=10 EXD 780
      GO TO 70 EXD 790
      END EXD 800

```

```

SUBROUTINE EXPCON                                     EXN 10
C VERSION 5.00      EXPCON      5/15/70          EXN 20
C EXPCON SUBROUTINE S PEAVY      4/4/68          EXN 30
C COMMANDS                                         EXN 40
C L2=1: MVECDIAG                                EXN 50
C     MVECDIAG MATRIX IN R , C SIZE N , M PUT DIAGONAL IN C EXN 60
C     MVECDIAG MATRIX IN R , C SIZE N , M PUT DIAGONAL IN R , C EXN 70
C L2=2: MVECMMAT                                EXN 80
C     MVECMMAT MATRIX IN R , C SIZE N , M PUT ROW BY ROW EXN 90
C                                         AS A VECTOR IN C EXN 100
C     MVECMMAT MATRIX IN R , C SIZE N , M PUT ROW BY ROW AS A EXN 110
C                                         VECTOR IN R , C EXN 120
C L2=3: MMATVEC                                 EXN 130
C     MMATVEC  VECTOR C PUT AS ROW X ROW MATRIX IN R , C SIZE N , MEXN 135
C     MMATVEC  VECTOR R , C PUT AS ROW X ROW MATRIX IN R , C EXN 140
C                                         SIZE N X M EXN 145
COMMON /BLOCRC/ NRC,RC(12600)                      EXN 160
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NEXN 170
1ARGS,VWXYZ(8),NERROR                           EXN 180
DIMENSION ARGS(100)                             EXN 190
EQUIVALENCE (ARGS(1),RC(12501))                EXN 200
COMMON /SCRAT/ NS,NS2,A(13500)                  EXN 210
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG            EXN 220
IF (NARGS.EQ.5.OR.NARGS.EQ.6) GO TO 10          EXN 230
CALL ERROR (10)                                  EXN 240
RETURN                                            EXN 250
10 J=NARGS                                         EXN 260
KRR=0                                             EXN 270
CALL CKIND (J)                                    EXN 280
IF (J.EQ.0) IF (L2-2) 15,65,125                 EXN 285
CALL ERROR (3)                                    EXN 290
RETURN                                            EXN 300
C VEC DIAG ****                                EXN 310
15 IARGS(7)=MIN0(IARGS(3),IARGS(4))             EXN 320
IARGS(8)=1                                         EXN 330
IF (NARGS.EQ.6) GO TO 20                         EXN 340
IARGS(6)=IARGS(5)                               EXN 350
IARGS(5)=1                                         EXN 360
MKKR=226                                         EXN 370
20 IF (IARGS(5)+IARGS(7)-1.LE.NROW) GO TO 30    EXN 380
IARGS(7)=NROW-IARGS(5)+1                         EXN 390
KRR=MKKR                                         EXN 400
C ERROR 226: COLUMN NOT LONG ENOUGH TO STORE ALL ELEMENTS. ONLY NROWEXN 410
C WILL BE STORE                                     EXN 420
30 J=2                                             EXN 430
CALL MTXCHK (J)                                    EXN 440
IF (J.NE.0) CALL ERROR (17)                      EXN 450
IF (NERROR.NE.0) RETURN                          EXN 460
IF (KRR.NE.0) CALL ERROR (KRR)                  EXN 470
GO TO (40,80,140), L2                            EXN 480
40 IA=IARGS(1)                                     EXN 490
IB=IARGS(7)                                       EXN 500
DO 50 I=1,IB                                     EXN 510
A(I)=RC(IA)                                       EXN 520
50 IA=IA+NROW+1                                   EXN 530
IA=IARGS(5)                                       EXN 540
DO 60 I=1,IB                                     EXN 550
RC(IA)=A(I)                                       EXN 560
60 IA=IA+1                                       EXN 570

```

```

      RETURN                                EXN 580
C   65  VECTORIZE A MATRIX ***                  EXN 590
      IARGS(7)=IARGS(3)*IARGS(4)              EXN 600
      IARGS(8)=1                               EXN 610
      IF (NARGS.EQ.6) GO TO 70                 EXN 620
      IARGS(6)=IARGS(5)                         EXN 630
      IARGS(5)=1                               EXN 640
    70  MKKR=226                                EXN 650
      GO TO 20                                 EXN 690
    80  IB=IARGS(7)                                EXN 700
      IA=IARGS(1)                                EXN 710
      N=IARGS(3)                                EXN 720
      M=IARGS(4)                                EXN 730
      IC=1                                     EXN 740
      DO 100 I=1,N                            EXN 750
      IAA=IA                                     EXN 760
      DO 90 J=1,M                            EXN 770
      A(IC)=RC(IAA)                           EXN 780
      IF (IC.EQ.IB) GO TO 110                EXN 790
      IC=IC+1                                 EXN 800
    90  IAA=IAA+NROW                           EXN 810
    100 IA=IA+1                                EXN 820
    110 IA=IARGS(5)                                EXN 830
      DO 120 I=1,IB                            EXN 840
      RC(IA)=A(I)                             EXN 850
    120 IA=IA+1                                EXN 860
      RETURN                                  EXN 870
C   125 TAKE A COLUMN AND RESTORE IT TO A MATRIX OR ARRAY. EXN 880
    125 IARGS(8)=IARGS(NARGS)                  EXN 890
      IARGS(7)=IARGS(NARGS-1)                  EXN 900
      IARGS(6)=IARGS(NARGS-2)                  EXN 910
      IARGS(5)=IARGS(NARGS-3)                  EXN 920
      IF (NARGS.EQ.6) GO TO 130                EXN 930
      IARGS(2)=IARGS(1)                         EXN 940
      IARGS(1)=1                               EXN 950
    130 IARGS(3)=IARGS(7)*IARGS(8)             EXN 960
      IARGS(4)=1                               EXN 970
      IF (IARGS(1)+IARGS(3)-1.LE.NROW) GO TO 30 EXN 980
      IARGS(3)=NROW-IARGS(1)+1                 EXN 990
C   KRR=227                                EXN1000
C   227 ERROR:,NOT ENOUGH ELEMENTS IN COL TO RESTORE MATRIX OR ARRAY. EXN1010
C   ELEMENTS AVAILABLE WILL BE USED.          EXN1020
      GO TO 30                                 EXN1030
    140 IA=IARGS(1)                                EXN1040
      IB=IARGS(3)                                EXN1050
      DO 150 I=1,IB                            EXN1060
      A(I)=RC(IA)                             EXN1070
    150 IA=IA+1                                EXN1080
      IA=IARGS(5)                                EXN1090
      N=IARGS(7)                                EXN1100
      M=IARGS(8)                                EXN1110
      IC=1                                     EXN1120
      DO 170 I=1,N                            EXN1130
      IAA=IA                                     EXN1140
      DO 160 J=1,M                            EXN1150
      RC(IAA)=A(IC)                           EXN1160
      IF (IC.EQ.IB) RETURN                    EXN1170
      IC=IC+1                                 EXN1180
    160 IAA=IAA+NROW                           EXN1190
    170 IA=IA+1                                EXN1200
      RETURN                                  EXN1210
      END                                    EXN1220

```

```

SUBROUTINE EXTREM                                EXT 10
C      VERSION 5.00      EXTREM    5/15/70        EXT 20
COMMON /BLOCRC/ NRC,RC(12600)                   EXT 30
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NEXT 40
IARGS,VWXYZ(8),NERROR                         EXT 50
DIMENSION ARGS(100)                           EXT 60
EQUIVALENCE (ARGS(1),RC(12501))             EXT 70
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG          EXT 80
C
C      L2 = 4,5 MAX      L2 = 6,7 MIN          EXT 90
C
C      MAX OF ++ TO ++          EXT 120
C      MAX OF ++ TO ++, CORRESP ENTRY OF ++ TO ++, ++ TO ++, ETC.  EXT 130
C      LIKEWISE FOR MIN.          EXT 140
C
C      IF (NARGS.GT.0.AND.MOD(NARGS,2).EQ.0) GO TO 30          EXT 160
I=10
10     CALL ERROR (I)                            EXT 170
20     RETURN                                     EXT 190
30     CALL CHKCOLUMN (I)                        EXT 200
      IF (I.EQ.0) GO TO 40                      EXT 210
      I=20
      GO TO 10                                    EXT 230
40     IF (NERROR.NE.0) GO TO 20                EXT 240
      J=0
      IF (NRMAX-1) 50,110,60                     EXT 260
50     I=9
      GO TO 10                                    EXT 280
60     J=IARGS(1)                               EXT 290
      K=J+1
      L=K+NRMAX-2                  EXT 300
      IF (L2.GT.5) GO TO 80                  EXT 310
C
C      FIND MAXIMUM                           EXT 330
C
C      DO 70 I=K,L                            EXT 360
      IF (RC(J).LT.RC(I)) J=I
70     CONTINUE                                 EXT 370
      GO TO 100                                EXT 390
C
C      FIND MINIMUM                           EXT 400
C
80     DO 90 I=K,L                            EXT 430
      IF (RC(J).GT.RC(I)) J=I
90     CONTINUE                                 EXT 440
100    J=J-IARGS(1)                           EXT 450
110    DO 120 I=1,NARGS,2                    EXT 460
      K=IARGS(I)+J
120    CALL VECTOR (RC(K),IARGS(I+1))
      GO TO 20
      END                                     EXT 500
                                         EXT 510

```

```

C FUNCTION FCOS (X) FCO 10
C VERSION 5.00 FCOS 5/15/70 FCO 20
C
C THIS FUNCTION IS TO TRAP IF ARGUMENT IS GREATER THAN 3.3E7 IN FCO 30
C ABSOLUTE VALUE BEFORE SYSTEM DOES. RESULT SET = 0.0 FCO 40
C
C COMMON /CONSLB/ XTRIG,XEXP FCO 50
C IF (ABS(X).GT.XTRIG) GO TO 20 FCO 60
C FCOS=COS(X)
10 RETURN FCO 70
20 CALL ERROR (104) FCO 80
FCOS=0. FCO 90
GO TO 10 FCO 100
END FCO 110
FCO 120
FCO 130
FCO 140

```

```

C DOUBLE PRECISION FUNCTION FDCOS(X) FDC 10
C VERSION 5.00 FDCOS 5/15/70 FDC 20
C
C THIS FUNCTION IS TO TRAP IF ARGUMENT IS GREATER THAN 3.5016 IN FDC 30
C ABSOLUTE VALUE BEFORE SYSTEM DOES. RESULT SET =0 WITH DIAGNOSTIC. FDC 40
C
C FDC 50
C FDC 60
C
C DOUBLE PRECISION DSNCOS,DXEXP FDC 70
C COMMON /DCONLB/ DSNCOS,DXEXP FDC 80
C DOUBLE PRECISION X,DCOS FDC 90
C IF (DABS(X).GT.DSNCOS) GO TO 20 FDC 100
C FDCOS=DCOS(X)
10 RETURN FDC 110
20 CALL ERROR (104) FDC 120
FDCOS=0.D0 FDC 130
GO TO 10 FDC 140
END FDC 150
FDC 160

```

```

C DOUBLE PRECISION FUNCTION FDEXP (X) FDE 10
C VERSION 5.00 FDEXP 5/15/70 FDE 20
C
C THIS FUNCTION IS TO TRAP IF ARGUMENT IS GREATER THAN 704.DO FDE 30
C BEFORE SYSTEM DOES. RESULT IS SET = 0.0 AND DIAGNOSTIC IS PRINTED.FDE 40
C
C FDE 50
C FDE 60
C
C DOUBLE PRECISION DSNCOS,DXEXP FDE 70
C COMMON /DCONLB/ DSNCOS,DXEXP FDE 80
C DOUBLE PRECISION X,DEXP FDE 90
C IF (X.GT.DXEXP) GO TO 20 FDE 100
C FDEXP=DEXP(X)
10 RETURN FDE 110
20 CALL ERROR (102) FDE 120
FDEXP=0.0DO FDE 130
GO TO 10 FDE 140
END FDE 150
FDE 160

```

```

DOUBLE PRECISION FUNCTION FDLOG(X) FDL 10
C VERSION 5.00 FDLOG 5/15/70 FDL 20
C
C THIS FUNCITON IS TO TRAP ILLEGAL ARGUMENT FDL 30
C BEFORE SYSTEM DOES. RESULT SET 0.0 AND DIAGNOSTIC IS PRINTED FDL 40
C
C DOUBLE PRECISION X,DLOG FDL 50
IF (X.GT.0.D0) GO TO 10 FDL 60
CALL ERROR (101) FDL 70
FDLOG=0.D0 FDL 80
GO TO 20 FDL 90
10 FDLOG=DLOG(X) FDL 100
20 RETURN FDL 110
END FDL 120

```

```

FUNCTION FDPCON (X) FDP 10
C VERSION 5.00 FDPCON 5/15/70 FDP 20
C WRITTEN BY DAVID HOGBEN, SEL, NBS. 8/16/69. FDP 40
DOUBLE PRECISION X,D FDP 50
Y=X FDP 60
D=Y FDP 70
FDPCON=X+(X-D) FDP 80
RETURN FDP 90
END FDP 100

```

```

DOUBLE PRECISION FUNCTION FDSIN(X) FDS 10
C VERSION 5.00 FDSIN 5/15/70 FDS 20
C
C THIS FUNCITON IS TO TRAP IF ARGUMENT IS GREATER THAN 3.5D16 IN FDS 30
C ABSOLUTE VALUE BEFORE SYSTEM DOES. RESULT SET = 0.0 FDS 40
C
C DOUBLE PRECISION DSNCOS ,DXEXP FDS 50
COMMON /DCONLB/ DSNCOS ,DXEXP FDS 60
DOUBLE PRECISION X,DSIN FDS 70
IF (DABS(X).GT.DSNCOS) GO TO 20 FDS 80
FDSIN=DSIN(X) FDS 90
10 RETURN FDS 100
20 CALL ERROR (104) FDS 110
FDSIN=0.D0 FDS 120
GO TO 10 FDS 130
END FDS 140
FDS 150
FDS 160

```

```

C DOUBLE PRECISION FUNCTION FDSQRT (X) FDQ 10
C VERSION 5.00      FDSQRT      5/15/70 FDQ 20
C
C THIS FUNCTION TRAPS IF ARGUMENT IS NEGATIVE BEFORE SYSTEM DOES. FDQ 30
C RESULT SET = 0.0 AND DIAGNOSTIC IS PRINTED FDQ 40
C
C DOUBLE PRECISION X,DSQRT FDQ 60
IF (X.LT.0.D0) GO TO 10 FDQ 70
FDSQRT=DSQRT(X) FDQ 80
RETURN FDQ 90
10 CALL ERROR (101) FDQ 100
FDSQRT=0.D0 FDQ 110
RETURN FDQ 120
END FDQ 130
FDQ 140

```

```

C FUNCTION FEXP (X) FEX 10
C VERSION 5.00      FEXP       5/15/70 FEX 20
C
C THIS FUNCTION IS TO TRAP IF ARGUMENT IS GREATER THAN 88.0 BEFORE FEX 30
C SYSTEM DOES. RESULT SET = 0.0. FEX 40
C
C COMMON /CONSLB/ XTRIG,XEXP FEX 50
IF (X.GT.XEXP) GO TO 20 FEX 60
FEXP=EXP(X)
10 RETURN FEX 70
20 CALL ERROR (102) FEX 80
FEXP=0. FEX 90
GO TO 10 FEX 100
END FEX 110
FEX 120
FEX 130
FEX 140

```

```

C FUNCTION FEXP2 (B,E) FX2 10
C VERSION 5.00      FEXP2      5/15/70 FX2 20
C DATA IEXP/60/ FX2 30
C
C THIS FUNCTION IS INCLUDED TO CATCH EXPONENTIATION ERRORS BEFORE FX2 40
C THE SYSTEM DOES FX2 50
C
C IE=E FX2 60
IF (E.EQ.FLOAT(IE).AND.IE.LT.IEXP) GO TO 20 FX2 70
FEXP2=FEXP(E*FLOG(B))
10 RETURN FX2 80
20 FEXP2=B**IE FX2 90
GO TO 10 FX2 100
END FX2 110
FX2 120
FX2 130
FX2 140

```

```

SUBROUTINE FIXFLO          FIX 10
C VERSION 5.00   FIXFLO    5/15/70   FIX 20
COMMON /ABCDEF/ L(48)      FIX 30
COMMON /BLOCRC/ NRC,RC(12600) FIX 40
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NFIX 50
IARGS,VWXYZ(8),NERROR    FIX 60
DIMENSION ARGS(100)       FIX 70
EQUIVALENCE (ARGS(1),RC(12501)) FIX 80
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG FIX 90
COMMON/CONLB2/ER,ISIGD     FIX 95
COMMON /FMAT/ IFMTX(6),IOSWT,IFMTS(6),LHEAD(96) FIX 100
DIMENSION IB(3)            FIX 110
DATA IB(1),IB(2),IB(3)/2H8F,2H8E,2H1P/ FIX 120
C                                         FIX 130
C L2 = 3 FOR FIXED, L2 = 4 FOR FLOAT FIX 140
C L2=12 FOR FLEXIBLE               FIX 150
C                                         FIX 160
C IF (L2.NE.12) GO TO 5           FIX 170
C IOSWT=0                         FIX 180
C RETURN                           FIX 190
5 IF (L2.NE.4.OR.NARGS.NE.0) GO TO 8 FIX 192
I=6                               FIX 194
GO TO 50                          FIX 196
8 IF (NARGS.EQ.1) IF (KIND(1)) 30,40,30 FIX 198
I=10                             FIX 200
10 CALL ERROR (I)                 FIX 210
20 RETURN                         FIX 220
30 I=20                            FIX 230
GO TO 10                          FIX 240
40 I=IARGS(1)                     FIX 250
IF(I.GE.0.AND.I.LE.ISIGD) GO TO 50 FIX 260
I=ISIGD                          FIX 270
CALL ERROR (237)                 FIX 280
50 IOSWT=1                         FIX 290
IFMTX(5)=L(I+1)                  FIX 300
IF (L2.EQ.4) GO TO 60             FIX 310
C SET UP FIXED FORMAT             FIX 320
IFMTX(3)=IB(1)                   FIX 330
IFMTX(2)=L(45)                   FIX 340
RETURN                           FIX 350
C SET UP FLOATING FORMAT         FIX 360
60 IFMTX(3)=IB(2)                 FIX 370
IFMTX(2)=IB(3)                   FIX 380
RETURN                           FIX 390
END                             FIX 400

```

```

SUBROUTINE FLIP                               FLI 10
C      VERSION 5.00    FLIP      5/15/70          FLI 20
COMMON /BLOCRC/ NRC,RC(12600)                FLI 30
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NFLI 40
1ARGS,VWXYZ(8),NERROR                      FLI 50
DIMENSION ARGS(100)                         FLI 60
EQUIVALENCE (ARGS(1),RC(12501))            FLI 70
EQUIVALENCE (I,IARGS(100)), (J,IARGS(99)), (K,IARGS(98)), (KK,IARGFLI 80
1S(97)), (M,IARGS(96)), (MM,IARGS(95)), (MMM,IARGS(94)), (N,IARGS(9FLI 90
23)), (NN,IARGS(92)), (A,ARGS(1))          FLI 100
C                                         FLI 110
C      FLIP COL ++ INTO COL ++, ++ INTO ++, ETC.   FLI 120
C      IF NARGS = 0, FLIP THE ENTIRE ARRAY (WORKSHEET).   FLI 130
C                                         FLI 140
IF (NARGS.EQ.0) GO TO 40                     FLI 150
IF (MOD(NARGS,2).EQ.0) GO TO 30             FLI 160
I=10                                         FLI 170
10     CALL ERROR (I)                        FLI 180
20     RETURN                                FLI 190
30     CALL CHKCOL (I)                      FLI 200
IF (I.EQ.0) GO TO 40                         FLI 210
I=20                                         FLI 220
GO TO 10                                     FLI 230
40     IF (NERROR.NE.0) GO TO 20             FLI 240
IF (NRMAX-1) 50,20,60                         FLI 250
50     I=9                                     FLI 260
GO TO 10                                     FLI 270
60     KK=NRMAX-1                           FLI 280
K=KK/2                                       FLI 290
IF (NARGS.EQ.0) GO TO 90                     FLI 300
DO 80 I=1,NARGS,2                           FLI 310
M=IARGS(I)                                    FLI 320
N=IARGS(I+1)                                  FLI 330
MM=M+KK                                      FLI 340
NN=N+KK                                      FLI 350
MMM=M+K                                       FLI 360
DO 70 J=M,MMM                                FLI 370
A=RC(J)                                       FLI 380
RC(N)=RC(MM)                                 FLI 390
RC(NN)=A                                     FLI 400
N=N+1                                         FLI 410
MM=MM-1                                      FLI 420
70     NN=NN-1                                FLI 430
80     CONTINUE                                FLI 440
GO TO 20                                     FLI 450
C                                         FLI 460
C      FLIP ENTIRE ARRAY                      FLI 470
C                                         FLI 480
90     N=1                                     FLI 490
DO 110 I=1,NCOL                            FLI 500
M=N                                         FLI 510
MM=M+KK                                      FLI 520
DO 100 J=1,K                                FLI 530
A=RC(M)                                       FLI 540
RC(M)=RC(MM)                                 FLI 550
RC(MM)=A                                     FLI 560
M=M+1                                         FLI 570
100    MM=MM-1                                FLI 580
110    N=N+NROW                                FLI 590
GO TO 20                                     FLI 600
END                                         FLI 610

```

	FUNCTION FLOG (X)	FLE	10
C	VERSION 5.00 FLOG 5/15/70	FLE	20
C	FLOG CHECKS TO SEE IF ARGUMENT IS GREATER THAN 0, BEFORE USING	FLE	23
C	LIBRARY FUNCTION FOR NATURAL LOG.	FLE	25
C	IF X IS ZERO OR NEG., RESULT IS = 0 AND DIAGNOSTIC IS PRINTED.	FLE	27
	IF (X.GT.0.) GO TO 10	FLE	30
	CALL ERROR (101)	FLE	40
	FLOG=0.	FLE	50
	GO TO 20	FLE	60
10	FLOG=ALOG(X)	FLE	70
20	RETURN	FLE	80
	END	FLE	90

	FUNCTION FLOG10 (X)	FLT	10
C	VERSION 5.00 FLOG10 5/15/70	FLT	20
C		FLT	30
C	THIS FUNCTION CHECKS TO SEE IF X IS ZERO OR NEGATIVE BEFORE USING	FLT	40
C	LIBRARY ALOG10. INFORMATIVE DIAGNOSTIC IS PRINTED AND 0 RETURNED.	FLT	50
C		FLT	60
	IF (X.GT.0.0) GO TO 20	FLT	70
	CALL ERROR (101)	FLT	80
	FLOG10=0.0	FLT	90
10	RETURN	FLT	100
20	FLOG10=ALOG10(X)	FLT	110
	GO TO 10	FLT	120
	END	FLT	130

```

SUBROUTINE FNEC                                     FNE 10
C      VERSION 5.00      FNEC      5/15/70          FNE 20
C      THIS SUBROUTINE HANDLES MISC FUNCTION COMMANDS WITH TWO ARGUMENTS, FNE 30
C      THE FIRST IS (E) (A CONSTANT, OR COLUMN NUMBER) AND THE SECOND IS FNE 40
C      (C) A COLUMN NUMBER                                FNE 50
C      WRITTEN BY DAVID HOGBEN, SEL, NBS.   3/27/70.      FNE 60
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NFNE 70
IARGS,VWXYZ(8),NERROR                           FNE 80
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG             FNE 90
COMMON /BLOCRC/ NRC,RC(12600)                   FNE 100
DIMENSION ARGS(100)                            FNE 110
EQUIVALENCE (ARGS(1),RC(12501))              FNE 120
COMMON /SCRAT/ NS,NS2,A(13500)                 FNE 130
DOUBLE PRECISION Y,Z                          FNE 140
C      ***** COMMANDS *****                      FNE 150
C      ERROR (E), (C)                         L2=18    BY I. STEGUN, 3/25/70. FNE 160
C      CERF (E), (C)                         L2=19    BY I. STEGUN, 3/25/70. FNE 170
C      *****                                FNE 180
10     IF (NARGS.NE.2) CALL ERROR (10)           FNE 190
IF (NRMAX.EQ.0) CALL ERROR (9)                  FNE 200
CALL ADRESS (1,J1)                            FNE 210
IF (J1.LT.0) J1=-J1                           FNE 220
IF (J1.EQ.0) CALL ERROR (11)                  FNE 230
CALL ADRESS (2,J2)                            FNE 240
IF (J2.EQ.0) CALL ERROR (11)                  FNE 250
IF (J2.LT.0) CALL ERROR (20)                  FNE 260
IF (NERROR.NE.0) GO TO 70                     FNE 270
LL=L2-17                                      FNE 280
DO 60 I=1,NRMAX                               FNE 290
IF (KIND(1).EQ.1.AND.I.GT.1) GO TO 50        FNE 300
GO TO (20,30), LL                            FNE 310
20     CALL ERRINT (DBLE(RC(J1)),Y,Z)          FNE 320
X=FDPCON(Y)                                  FNE 330
GO TO 40                                      FNE 340
30     CALL ERRINT (DBLE(RC(J1)),Z,Y)          FNE 350
X=FDPCON(Y)                                  FNE 360
GO TO 40                                      FNE 370
40     J1=J1+1                                  FNE 380
50     RC(J2)=X                                FNE 390
60     J2=J2+1                                  FNE 400
70     RETURN                                    FNE 410
END                                         FNE 420

```

```

SUBROUTINE FNEIC          FNC 10
C VERSION 5.00   FNEIC    5/15/70      FNC 20
C THIS SUBROUTINE HANDLES INSTRUCTIONS OF THE FORM (E), (I), (C)  FNC 30
C WRITTEN BY DAVID HOGBEN, SEL, NBS.  4/21/70.      FNC 40
C ***** COMMON *****
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NFNC 60
IARGS,VWXYZ(8),NERROR      FNC 70
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG      FNC 80
COMMON /BLOCRC/ NRC,RC(12600)      FNC 90
DIMENSION ARGS(100)      FNC 100
EQUIVALENCE (ARGS(1),RC(12501))      FNC 110
COMMON /SCRAT/ NS,NS2,A(13500)      FNC 120
*****      FNC 130
C IF (NRMAX.EQ.0) CALL ERROR (9)      FNC 140
IF (NARGS.NE.3) CALL ERROR (10)      FNC 150
IF (KIND(2).NE.0) CALL ERROR (20)      FNC 160
CALL ADRESS (1,J)      FNC 170
IF (J.LT.0) J=-J      FNC 180
CALL ADRESS (3,K)      FNC 190
IF (J.EQ.0.OR.K.EQ.0) CALL ERROR (11)      FNC 200
IF (K.LT.0) CALL ERROR (20)      FNC 210
IF (NERROR.NE.0) RETURN      FNC 220
N=IARGS(2)      FNC 230
DO 10 I=1,NRMAX      FNC 240
C ROUND X EQUAL TO (E) TO (I) SIGNIFICANT DIGITS, PUT IN COLUMN (C) FNC 250
C SUBROUTINE TO ROUND. WRITTEN BY DAVID HOGBEN, SEL, NBS.  10/21/68. FNC 260
CALL DHRND (RC(J),N,RC(K))      FNC 270
IF (KIND(1).EQ.0) J=J+1      FNC 280
10 K=K+1      FNC 290
RETURN      FNC 300
END      FNC 310

```

```

SUBROUTINE FNKC                               FKC 10
C      VERSION 5.00    FNKC      5/15/70          FKC 20
C      SUBROUTINE TREATS INSTRUCTIONS OF THE FORM (K),(C)   FKC 30
C      WRITTEN BY DAVID HOGBEN, SEL, NBS. 4/22/70.          FKC 40
C      ***** COMMON *****                         FKC 50
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NFKC 60
IARGS,VWXYZ(8),NERROR                      FKC 70
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG          FKC 80
COMMON /BLOCRC/ NRC,RC(12600)                 FKC 90
DIMENSION ARGS(100)                         FKC 100
EQUIVALENCE (ARGS(1),RC(12501))            FKC 110
COMMON /SCRAT/ NS,NS2,A(13500)               FKC 120
*****                                     FKC 130
C      IF (L1.NE.24.OR.L2.NE.15) RETURN        FKC 140
CALL ERROR (235)                           FKC 150
IF (NRMAX.EQ.0) CALL ERROR (9)              FKC 160
IF (NARGS.NE.2) CALL ERROR (10)             FKC 170
CALL ADRESS (2,J)                         FKC 180
IF (J.EQ.0) CALL ERROR (11)                FKC 190
IF (J.LT.0) CALL ERROR (20)                FKC 200
IF (NERROR.NE.0) RETURN                   FKC 210
NST=ARGS(1)                                FKC 220
IF (NST.LE.0) NST=8192.0*ARGS(1)+0.1       FKC 230
IF (KIND(1).EQ.0) NST=IARGS(1)             FKC 240
NST=MOD(NST,8192)                          FKC 250
DO 10 I=1,NRMAX                            FKC 260
C      RNJBK SHOULD BE REPLACED BY MORE RELIABLE AND EFFICIENT GENERATOR.FKC 270
CALL RNJBK (RC(J),NST,NST)                  FKC 280
10 J=J+1                                    FKC 290
      RETURN                                  FKC 300
      END                                     FKC 310

```

```

C      SUBROUTINE FOURIA (Y,A,R,N,KAA)          FOU 10
      VERSION 5.00      FOURIA      5/15/70      FOU 20
      DOUBLE PRECISION Y(1),R(1),A,AA,AB,AC,BA,BB,AD  FOU 30
      DOUBLE PRECISION FDCOS,FDSIN      FOU 40
      M=N/2      FOU 50
      K=2*M      FOU 60
      L=0      FOU 70
      IF (N.EQ.K) GO TO 10      FOU 80
      L=1      FOU 90
10     AB=N      FOU 100
      AA=6.28318530717D0/AB      FOU 110
      A=0.0      FOU 120
      R(M)=.0      FOU 130
      AC=1.      FOU 140
      DO 20 I=1,N      FOU 150
      A=A+Y(I)      FOU 160
      R(M)=R(M)+AC*Y(I)      FOU 170
20     AC=-1.*AC      FOU 180
      A=A/AB      FOU 190
      R(M)=R(M)/AB      FOU 200
      J=M+L-1      FOU 210
      KA=M+1      FOU 220
      DO 40 K=1,J      FOU 230
      BA=Y(1)      FOU 240
      BB=0.0      FOU 250
      AC=K      FOU 260
      AC=AC*AA      FOU 270
      DO 30 I=2,N      FOU 280
      AD=I-1      FOU 290
      AD=AD*AC      FOU 300
      BA=BA+Y(I)*FDCOS(AD)      FOU 310
      BB=BB+Y(I)*FDSIN(AD)      FOU 320
      R(K)=2.*BA/AB      FOU 330
      R(KA)=2.*BB/AB      FOU 340
40     KA=KA+1      FOU 350
      IF (L.EQ.1) GO TO 50      FOU 360
      R(KA)=0.      FOU 370
50     RETURN      FOU 380
      END      FOU 390

```

	SUBROUTINE FPPT (V11,V12,P10,XA)	FPP	10
C	VERSION 5.0 FPPT 5/15/70.	FPP	20
	V1=V11	FPP	30
	V2=V12	FPP	40
	P0=P10	FPP	50
C	CALLS PROB AND THEN USES ISOLATE METHOD FOR SOLVING ITERATIVELY	FPP	60
	DIMENSION X(5), P(5)	FPP	70
	IF (V1-1.5) 10,10,20	FPP	80
C	USE STUDENT'S T	FPP	90
C	ONLY GOOD FOR P0=0.05	FPP	100
10	CALL TPCTPT (V2,XA)	FPP	110
	XA=XA**2.0	FPP	120
	GO TO 70	FPP	130
20	IF (V2-1.5) 30,30,40	FPP	140
C	ONLY GOOD FOR P0=0.05	FPP	150
C	SHOULD USE STUDENT'S T	FPP	160
30	XA = 225.0	FPP	170
	GO TO 70	FPP	180
C	TUKEY APPROXIMATION TO NORMAL PERCENT POINT	FPP	190
40	YP=-4.91*(P0**.14-(1.-P0)**.14)	FPP	200
C	AMS 55 APPROXIMATION 26.5.22	FPP	210
	H=2.0/(1.0/(V1-1.0)+1.0/(V2-1.0))	FPP	220
	XLMBDA=(YP**2-3.0)/6.0	FPP	230
	W=YP*FSQRT(H+XLMBDA)/H	FPP	240
	IF (V1-V2) 50,60,50	FPP	250
50	W=W-(1.0/(V1-1.0)-1.0/(V2-1.0))*(XLMBDA+0.833333-0.666667/H)	FPP	260
C	AMS 55 APPROXIMATION 26.6.16	FPP	270
60	XA=FEXP(2.*W)	FPP	280
70	XMIN=0.5*XA	FPP	290
	XMAX=2.0*XA	FPP	300
	CALL PROB (V1,V2,XMAX,Q)	FPP	310
	IF (Q.LE.P0) GO TO 80	FPP	320
	XA=1.9999*XMAX	FPP	330
	GO TO 70	FPP	340
80	CALL PROB (V1,V2,XMIN,Q)	FPP	350
	IF (P0.LE.Q) GO TO 90	FPP	360
	XA=0.5001*XMIN	FPP	370
	GO TO 70	FPP	380
90	X0=XA	FPP	390
	DO 140 I=1,5	FPP	400
	X(I)=XMIN+FLOAT(I-1)*(XMAX-XMIN)/4.	FPP	410
100	CALL PROB (V1,V2,X(I),P(I))	FPP	420
110	IF (P0-P(I)) 140,130,120	FPP	430
120	XMAX=X(I)	FPP	440
	XMIN=X(I-1)	FPP	450
	GO TO 150	FPP	460
130	XA=X(I)	FPP	470
	GO TO 160	FPP	480
140	CONTINUE	FPP	490
150	XA=(XMIN+XMAX)/2.	FPP	500
C	EXIT IF EITHER TOLERANCE IS SATISFIED * ABSOLUTE 5E-6, REL. 5E-7	FPP	510
	IF (ABS(X0-XA).GT.5.E-6.AND.ABS(X0-XA)/XA.GT.5.E-7) GO TO 90	FPP	520
160	RETURN	FPP	530
	END	FPP	540

```

SUBROUTINE FPROB                                FPR 10
C   VERSION 5.00      FPROB      5/15/70        FPR 20
C   WRITTEN BY S PEAVY    10/13/67        FPR 30
C   COMMAND IS AS FOLLOWING                      FPR 40
C     FPROBABILITY V1 $,V2 $, F   $, STORE Q IN COL ++
C     COMMON /BLOCRC/ NRC,RC(12600)          FPR 50
C     COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NFPR 70
1ARGS,VWXYZ(8),NERROR                         FPR 80
DIMENSION ARGS(100)                           FPR 90
EQUIVALENCE (ARGS(1),RC(12501))            FPR 100
IF (NARGS.NE.4) CALL ERROR (10)             FPR 110
IF (KIND(NARGS).NE.0) CALL ERROR (3)         FPR 120
I1=1                                         FPR 130
I2=1                                         FPR 140
I3=1                                         FPR 150
CALL ADRESS (1,IARGS(1))                     FPR 160
IF (IARGS(1)) 10,20,30                        FPR 170
10   I1=2                                     FPR 180
     V1=ARGS(1)                               FPR 190
     GO TO 40                                 FPR 200
20   CALL ERROR (11)                           FPR 210
30   L=IARGS(1)                               FPR 220
40   CALL ADRESS (2,IARGS(2))               FPR 230
     IF (IARGS(2)) 50,60,70                 FPR 240
50   I2=2                                     FPR 250
     V2=ARGS(2)                               FPR 260
     GO TO 80                                 FPR 270
60   CALL ERROR (11)                           FPR 280
70   M=IARGS(2)                               FPR 290
80   CALL ADRESS (3,IARGS(3))               FPR 300
     IF (IARGS(3)) 90,100,110              FPR 310
90   I3=2                                     FPR 320
     F=ARGS(3)                               FPR 330
     GO TO 120                               FPR 340
100  CALL ERROR (11)                           FPR 350
110  N=IARGS(3)                               FPR 360
120  CALL ADRESS (NARGS,K)                  FPR 370
     IF (K.LE.0) CALL ERROR (11)             FPR 380
     IF (I1+I2+I3.NE.6) GO TO 140           FPR 400
     CALL PROB (V1,V2,F,Q)                 FPR 410
     DO 130 I=1,NRMAX                      FPR 420
     RC(K)=Q                               FPR 430
130  K=K+1                                   FPR 440
     RETURN                                  FPR 450
140  DO 210 I=1,NRMAX                      FPR 460
     GO TO (150,160), I1                   FPR 470
150  V1=RC(L)                               FPR 480
     L=L+1                                   FPR 490
160  GO TO (170,180), I2                   FPR 500
170  V2=RC(M)                               FPR 510
     M=M+1                                   FPR 520
180  GO TO (190,200), I3                   FPR 530
190  F=RC(N)                               FPR 540
     N=N+1                                   FPR 550
200  CALL PROB (V1,V2,F,RC(K))            FPR 560
210  K=K+1                                   FPR 570
     RETURN                                  FPR 580
     END                                     FPR 590

```

```

SUBROUTINE FRDIST                               FRD  10
C      VERSION 5.00     FRDIST   5/15/70          FRD  20
COMMON/BLOCRC/NRC,RC(12600)                   FRD  30
COMMON/BLOCKD/IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,
1NARGS,VWXYZ(8),NERROR                      FRD  40
DIMENSION ARGS(100)                          FRD  50
EQUIVALENCE (ARGS(1),RC(12501))            FRD  70
COMMON/BLOCKE/NAME(4),L1,L2,ISRFLG           FRD  80
C      WRITTEN BY DAVID HOBGEN, SEL, NBS.    10/25/69.  FRD  90
C      ***** FORMS OF COMMAND *****          FRD 100
C      1 FREQUENCY OF COLUMN (C), PUT IN COLUMN (C)      FRD 110
C      2 FREQUENCY OF COL (C), USE (I) CELLS, PUT IN COLUMN (C)  FRD 120
C      3 FREQUENCY OF COL (C), USE (I) CELLS, OF LENGTH (A), PUT IN COL (C)  FRD 130
C      4 FREQUENCY OF COL (C), CELLS (I), LENGTH (A), START (A), IN COL (C)  FRD 140
C      5 FREQUENCY OF COL (C), LOWER (C), UPPER IN (C), FREQ IN (C)      FRD 150
C      7 FREQUENCY OF COL (C), CELLS (I), LENGTH (A), PUT IN (C), (C), (C)  FRD 160
C      8 FREQUENCY OF (C), CELLS (I), LENGTH (A), START (A), IN (C),(C),(C)  FRD 170
C      RESETS NRMAX TO NO. OF CELLS             FRD 180
C      *****
C      10 IF (NARGS.GT.1 .AND. NARGS.LT.8) GO TO 100        FRD 200
      CALL ERROR (10)                                FRD 210
      RETURN                                         FRD 220
      20 CALL ERROR (3)                                FRD 230
      RETURN                                         FRD 240
      30 CALL ERROR (11)                               FRD 250
      RETURN                                         FRD 260
      100 CALL ADRESS (1,J1)                           FRD 270
      IF (J1) 20,30,110                            FRD 280
      110 CALL ADRESS (NARGS,J2)                     FRD 290
      IF (J2) 20,30,120                            FRD 300
      120 NST = 1                                    FRD 310
      LIMIT = 0                                     FRD 320
      C      FORM (1)                                FRD 330
      IF (NARGS.EQ.2) GO TO 901                    FRD 340
      200 KN = IARGS(2)                             FRD 350
      C      FORM (3)                                FRD 360
      IF (NARGS.EQ.3) GO TO 902                    FRD 370
      IF (KIND(3).EQ.0) GO TO 500                FRD 380
      CELL = ARGS(3)                             FRD 390
      C      FORM (3)                                FRD 400
      IF (NARGS.EQ.4) GO TO 903                    FRD 410
      400 IF (KIND(4).EQ.0) GO TO 500                FRD 420
      NST = 0                                      FRD 430
      STRT = ARGS(4)                             FRD 440
      C      FORM (4)                                FRD 450
      IF (NARGS.EQ.5) GO TO 904                    FRD 460
      C      FORMS (5), (6), (7), (8)               FRD 470
      500 CALL ADRESS (NARGS-2,J3)                 FRD 480
      IF (J3) 20,30,510                            FRD 490
      510 CALL ADRESS (NARGS-1,J4)                 FRD 500
      IF (J4) 20,30,520                            FRD 510
      520 LIMIT = 1                                 FRD 520
      JJ = NARGS-3                                FRD 530
      GO TO (901,902,903,904), JJ                FRD 540
      901 KN = 0                                     FRD 550
      902 CELL = 0.0                                FRD 560
      903 STRT = 0.0                                FRD 570
      904 CALL FREQCY (RC(J1),RC(J2),NRMAX,KN,CELL,NST,STRT,LIMIT,
1      RC(J3),RC(J4))                         FRD 580
                                              FRD 590

```

NRMAX=KN  
RETURN  
END

FRD 600  
FRD 610  
FRD 620

C SUBROUTINE FREQCY (X,F,N,K,C,NSTART,START,LIMITS,XL,XU) FRE 10  
C VERSION 5.00 FREQCY 5/15/70 FRE 20  
C SUBROUTINE TO CONSTRUCT FREQUENCY DISTRIBUTION IN VECTOR F FOR FRE 30  
C VECTOR OF OBSERVATIONS X OF LENGTH N USING K CELLS OF LENGTH C. FRE 40  
C IF C=0.0, THEN C IS DETERMINED BY SUBROUTINE. IF BOTH K AND C=0.0, FRE 50  
C THEN BOTH K AND C ARE DETERMINED BY THE SUBROUTINE. FRE 60  
C IF NSTART=1, START DETERMINED. IF NSTART=0, START IS GIVEN. FRE 70  
C IF LIMITS=1, LOWER CELL BOUNDARIES ARE PUT IN XL AND UPPER IN XU. FRE 80  
C WRITTEN BY DAVID HOGBEN, SEL, NBS. 10/25/69. FRE 90  
DIMENSION X(1),F(1),XL(1),XU(1) FRE 100  
100 IF (K.GT.0) GO TO 200 FRE 110  
K = 1.5 + 3.3\*FLOG10(FLOAT(N)) FRE 120  
K = MAX0 (K,5) FRE 130  
200 XMIN=X(1) FRE 140  
XMAX=X(1) FRE 150  
DO 250 I=1,N FRE 160  
IF(X(I).LT.XMIN) XMIN=X(I) FRE 170  
IF(X(I).GT.XMAX) XMAX=X(I) FRE 180  
250 CONTINUE FRE 190  
IF (C) 300,260,300 FRE 200  
260 RANGE=XMAX-XMIN FRE 210  
C=RANGE/FLOAT(K-1) FRE 220  
300 IF (NSTART.EQ.0) GO TO 500 FRE 230  
START=XMIN-0.5\*C FRE 240  
500 DO 510 I=1,K FRE 250  
510 F(I)=0.0 FRE 260  
DO 520 I=1,N FRE 270  
J=(X(I)-START)/C+1.0 FRE 280  
520 F(J)=F(J)+1.0 FRE 290  
IF (LIMITS.EQ.0) RETURN FRE 300  
600 XL(1)=START FRE 310  
XU(1)=XL(1)+C FRE 320  
DO 610 I=2,K FRE 330  
XL(I)=XL(I-1)+C FRE 340  
610 XU(I)=XL(I)+C FRE 350  
RETURN FRE 360  
END FRE 370

```

FUNCTION FSIN (X) FSI .10
C VERSION 5.00 FSIN 5/15/70 FSI 20
C
C THIS FUNCTION IS TO TRAP IF ARGUMENT IS GREATER THAN 3.3E7 IN FSI 30
C ABSOLUTE VALUE BEFORE SYSTEM DOES. RESULT SET = 0.0 FSI 40
C
COMMON /CONSLB/ XTRIG,XEXP FSI 70
IF (ABS(X).GT.XTRIG) GO TO 20 FSI 80
FSIN=SIN(X) FSI 90
10 RETURN FSI 100
20 CALL ERROR (104) FSI 110
FSIN=0. FSI 120
GO TO 10 FSI 130
END FSI 140

```

```

FUNCTION FSQRT (X) FSQ 10
C VERSION 5.00 FSQRT 5/15/70 FSQ 20
C FSQRT CHECKS X FOR NEGATIVE VALUES. FSQ 23
C IF X.LT. ZERO, RESULT IS ZERO, AND DIAGNOSTIC IS PRINTED. FSQ 25
IF (X.LT.0.) GO TO 20 FSQ 30
FSQRT=SQRT(X) FSQ 40
10 RETURN FSQ 50
20 CALL ERROR (101) FSQ 60
FSQRT=0. FSQ 70
GO TO 10 FSQ 80
END FSQ 90

```

```

FUNCTION FTANH (X) FTA 10
C VERSION 5.00 FTANH 5/15/70 FTA 20
C
C SINCE TANH FUNCTION USES EXP FUNCTION FTANH CHECKS TO SEE IF THEFTA 40
C ABSOLUTE VALUE OF 2*X IS GREATER THAN XEXP (OR 88.3). IF THISFTA 50
C IS THE CASE, AND ERROR MESSAGE IS PRINTED AND FTANH=0. FTA 60
C
COMMON /CONSLB/ XTRIG,XEXP FTA 70
IF (ABS(2.*X).LE.XEXP) GO TO 20 FTA 80
FTANH=0.0 FTA 90
10 RETURN FTA 100
20 FTANH=TANH(X) FTA 110
GO TO 10 FTA 120
END FTA 130
FTA 140

```

```

SUBROUTINE FUNCT          FUN   10
C VERSION 5.00  FUNCT  5/15/70      FUN   20
COMMON /BLOCRC/NRC,RC(12600)      FUN   30
COMMON /BLOCKD/IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,      FUN   40
1 NARGS,VWXYZ(8),NERROR      FUN   50
DIMENSION ARGS(100)      FUN   60
EQUIVALENCE (ARGS(1),RC(12501))      FUN   70
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG      FUN   80
COMMON /CONSTS/ PI,E,HALFPI,DEG,RAD,XALOG      FUN   90
COMMON /SCRAT/ NS,NS2,A(13500)      FUN  100
DIMENSION II(4),KK(4)      FUN  110
EQUIVALENCE (II(1),I1),(II(2),I2),(II(3),I3),(II(4),I4)      FUN  120
C                                     FUN  130
C THIS SUBROUTINE HANDLES ALL TWO, THREE, AND FOUR ARGUMENT FUNCTION      FUN 140
C IF THE FIRST ARGUMENT IS A CONSTANT, THE FUNCION IS EVALUATED      FUN 150
C ONLY ONCE.      FUN 160
C                                     FUN 170
C L2= 1 SIN      L2= 2 COS      L2= 3 TAN      FUN 180
C L2= 4 COT      L2= 5 ARCSIN      L2= 6 ARCCOS      FUN 190
C L2= 7 ARCTAN      L2= 8 ARCCOT      L2= 9 SIND      FUN 200
C L2=10 COSD      L2=11 TAND      L2=12 COTD      FUN 210
C L2=13 ASIND      L2=14 ACOSD      L2=15 ATAND      FUN 220
C L2=16 ACOTD      L2=17 SQRT      L2=18 EXPONENT      FUN 230
C L2=19 NEGEXP      L2=20 LOGE      L2=21 LOGTEN      FUN 240
C L2=22 ANTILO      L2=23 SINH      L2=24 COSH      FUN 250
C L2=25 TANH      L2=26 COTH      L2=27 ASINH      FUN 260
C L2=28 ACOSH      L2=29 ATANH      L2=30 ACOTH      FUN 270
C L2=31 ABSOLU      L2=32 INTEGE      L2=33 FRACTI      FUN 280
C L2=34 SQUARE      FUN 290
C                                     FUN 300
IF(NARGS.LT.2.OR.NARGS.GT.4) CALL ERROR (10)      FUN 310
IF (NARGS.EQ.3) CALL ERROR (29)      FUN 315
DO 30 I=1,NARGS      FUN 320
KK(I)=1      FUN 330
CALL ADRESS (I,II(I))      FUN 340
IF(II(I)) 20,10,30      FUN 350
10 CALL ERROR (20)      FUN 360
GO TO 30      FUN 370
20 KK(I)=0      FUN 380
II(I)==II(I)      FUN 390
30 CONTINUE      FUN 400
IF(KK(NARGS).EQ.0) CALL ERROR (11)      FUN 410
IF(NRMAX.LE.0) CALL ERROR(9)      FUN 420
IF(NERROR.NE.0) RETURN      FUN 430
NR=NRMAX      FUN 440
IF(NARGS.EQ.4) GO TO 36      FUN 450
KK(4)=1      FUN 460
I4=I2      FUN 470
IF(NARGS.EQ.3) I4=I3      FUN 480
36 IL=I4      FUN 490
ILL=1      FUN 495
IF(KK(1).EQ.1) GO TO 40      FUN 500
IF(NARGS.EQ.2) NR=1      FUN 520
40 DO 1010 I=1,NR      FUN 530
GO TO (50,1005),ILL      FUN 535
50 X=RC(I1)      FUN 540
GO TO (110,120,130,140,150,160,170,180,190,200,
1 210,220,230,240,250,260,270,280,290,300,
2 310,320,330,340,350,360,370,380,390,400,      FUN 550
                                         FUN 560
                                         FUN 570

```

	3 410,420,430,440),L2	FUN 580
110	X=FSIN(X)	FUN 590
	GO TO 1000	FUN 600
120	X = FCOS(X)	FUN 610
	GO TO 1000	FUN 620
130	AX=FCOS(X)	FUN 630
	IF(AX.NE.0.0) GO TO 135	FUN 640
132	X = 0.0	FUN 650
	CALL ERROR (107)	FUN 660
	GO TO 1000	FUN 670
135	X = FSIN(X)/AX	FUN 680
	GO TO 1000	FUN 690
140	AX=FSIN(X)	FUN 700
	IF(AX.EQ.0.0) GO TO 132	FUN 710
	X = FCOS(X)/AX	FUN 720
	GO TO 1000	FUN 730
150	Y=X**2	FUN 740
	IF(Y.GT.(.5)) GO TO 151	FUN 750
	X=ATAN(X/FSQRT(1.-Y))	FUN 760
	GO TO 153	FUN 770
151	Z=X	FUN 780
	IF(X.NE.0.0) IF (ABS(X)-1.) 154,158,159	FUN 790
	X=HALFPI	FUN 800
152	IF(L2.EQ.5.OR.L2.EQ.13) X=HALFPI-X	FUN 810
153	IF(L2.GT.10) X=DEG*X	FUN 820
	GO TO 1000	FUN 830
154	YY=1.0	FUN 840
	DO 155 J=1,3	FUN 850
	Y=X**2	FUN 860
	IF(Y.LE.(.5)) GO TO 156	FUN 870
	YY=YY+YY	FUN 880
155	X=Y+Y-1.	FUN 890
	Y=X**2	FUN 900
156	X=SIGN(ATAN(FSQRT(1.-Y)/X)/YY,Z)	FUN 910
157	IF(Z.LT.0.0) X=PI+X	FUN 920
	GO TO 152	FUN 930
158	X=0.0	FUN 940
	GO TO 157	FUN 950
159	X=0.0	FUN 960
	CALL ERROR (103)	FUN 970
	GO TO 1000	FUN 980
160	GO TO 151	FUN 990
170	X=ATAN(X)	FUN1000
	GO TO 1000	FUN1010
180	IF(X.EQ.0.0) GO TO 132	FUN1020
	X=ATAN(1./X)	FUN1030
	GO TO 1000	FUN1040
190	X=FSIN(RAD*X)	FUN1050
	GO TO 1000	FUN1060
200	X=FCOS(RAD*X)	FUN1070
	GO TO 1000	FUN1080
210	X=X*RAD	FUN1090
	GO TO 130	FUN1100
220	X=X*RAD	FUN1110
	GO TO 140	FUN1120
230	GO TO 150	FUN1130
240	GO TO 151	FUN1140
250	X=DEG*ATAN(X)	FUN1150
	GO TO 1000	FUN1160

260	IF(X.EQ.0.0) GO TO 132	FUN1170
	X=DEG*ATAN(1.0/X)	FUN1180
	GO TO 1000	FUN1190
270	X=FSQRT(X)	FUN1200
	GO TO 1000	FUN1210
280	X=FEXP(X)	FUN1220
	GO TO 1000	FUN1230
290	X=FEXP(-X)	FUN1240
	GO TO 1000	FUN1250
300	X=FLOG(X)	FUN1260
	GO TO 1000	FUN1270
310	IF(X.GT.0.0) GO TO 315	FUN1280
	X=0.0	FUN1290
	CALL ERROR (101)	FUN1300
	GO TO 1000	FUN1310
315	X=FLOG10(X)	FUN1320
	GO TO 1000	FUN1330
320	IF(X.GT.XALOG) GO TO 325	FUN1340
	X=10.**X	FUN1350
	GO TO 1000	FUN1360
325	X=0.0	FUN1370
	CALL ERROR (102)	FUN1380
	GO TO 1000	FUN1390
330	AX=FTANH(X)	FUN1400
	GO TO 345	FUN1410
340	AX=1.0	FUN1420
345	Y=FEXP(X)	FUN1430
	X=.5*(Y+1./Y)*AX	FUN1440
	GO TO 1000	FUN1450
350	X=FTANH(X)	FUN1460
	GO TO 1000	FUN1470
360	Y=FTANH(X)	FUN1480
	IF(Y.EQ.0.0) GO TO 132	FUN1490
	X=1.0/Y	FUN1500
	GO TO 1000	FUN1510
370	X=SIGN(FLOG(ABS(X)+FSQRT(X**2+1.0)),X)	FUN1520
	GO TO 1000	FUN1530
380	X=FLOG(ABS(X)+FSQRT(X**2-1.0))	FUN1540
	GO TO 1000	FUN1550
390	IF(ABS(X).GE.1.0) GO TO 132	FUN1560
	X=.5*FLOG((1.+X)/(1.0-X))	FUN1570
	GO TO 1000	FUN1580
400	IF(ABS(X).LE.1.0) GO TO 132	FUN1590
	X=.5*FLOG((X+1.0)/(X-1.0))	FUN1600
	GO TO 1000	FUN1610
410	X=ABS(X)	FUN1620
	GO TO 1000	FUN1630
420	X=AINT(X)	FUN1640
	GO TO 1000	FUN1650
430	X=X-AINT(X)	FUN1660
	GO TO 1000	FUN1670
440	X=X*X	FUN1680
1000	XA=X	FUN1685
	IF(NARGS.EQ.2) GO TO 1007	FUN1690
1005	X=XA*RC(I2)+RC(I3)	FUN1695
1007	RC(I4)=X	FUN1700
	I1=I1+KK(1)	FUN1705
	I2=I2+KK(2)	FUN1710
	I3=I3+KK(3)	FUN1715

```
I4=I4+KK(4)          FUN1720
IF(I.EQ.1.AND.NARGS.NE.2.AND.KIND(1).EQ.1) ILL=2   FUN1722
1010 CONTINUE          FUN1725
IF(KIND(1).EQ.1.AND.NR.EQ.1) CALL VECTOR (RC(IL),IL) FUN1730
RETURN               FUN1740
END                  FUN1750
```

```

SUBROUTINE GENER          GEN 10
C VERSION 5.00   GENER 5/15/70          GEN 20
COMMON /BLOCRC/ NRC,RC(12600)          GEN 30
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NGEN 40
IARGS,VWXYZ(8),NERROR          GEN 50
DIMENSION ARGS(100)          GEN 60
EQUIVALENCE (ARGS(1),RC(12501))          GEN 70
C DELETE BLOCKE          GEN 80
C GENERATE          GEN 90
C     NARGS MUST BE .GE. 4 AND EVEN          GEN 100
IF (NARGS.GE.4.AND.MOD(NARGS,2).EQ.0) GO TO 10          GEN 110
CALL ERROR (10)          GEN 120
GO TO 90          GEN 130
C GET STORAGE COLUMN ADDRESS          GEN 140
10 CALL ADRESS (NARGS,J)          GEN 150
IF (J.GT.0) GO TO 20          GEN 160
CALL ERROR (3)          GEN 170
GO TO 90          GEN 180
20 IF (NERROR.NE.0) GO TO 90          GEN 190
C CONVERT INTEGERS TO FLOATING POINT          GEN 200
DO 30 I=2,NARGS          GEN 210
IF (KIND(I-1).EQ.0) ARGS(I-1)=IARGS(I-1)          GEN 220
30 CONTINUE          GEN 230
RC(J)=ARGS(1)          GEN 240
NDROW=J+NROW-1          GEN 250
DO 70 I=4,NARGS,2          GEN 260
IF (ARGS(I-3).GT.ARGS(I-1)) ARGS(I-2)=SIGN(ARGS(I-2),-1.)
S=SIGN(1.,ARGS(I-2))          GEN 270
ENDER=ARGS(I-1)-.01*ARGS(I-2)          GEN 280
40 J=J+1          GEN 290
RC(J)=RC(J-1)+ARGS(I-2)          GEN 300
IF (S*(RC(J)-ENDER)) 50,60,60          GEN 310
C     NOT DONE          GEN 320
50 IF (J.LT.NDROW) GO TO 40          GEN 330
C     EXCEEDED COLUMN LENGTH          GEN 340
CALL ERROR (201)          GEN 350
GO TO 80          GEN 360
C     PASSES GENERATE UPPER BOUND, SET IN UPPER BOUND          GEN 370
60 RC(J)=ARGS(I-1)          GEN 380
70 CONTINUE          GEN 390
80 NRMAX=MAX0(NRMAX,J-NDROW+NROW)          GEN 400
90 RETURN          GEN 410
END          GEN 420
                                     GEN 430

```

```

SUBROUTINE GQUAD                               QUA 10
C VERSION 5.00      GQUAD      5/15/70          QUA 20
C WRITTEN BY DAVID HOBGEN SEL, NBS.   8/18/69.    QUA 30
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NQUA 40
IARGS,VWXYZ(8),NERROR                         QUA 50
COMMON /BLOCRC/ NRC,RC(12600)                  QUA 60
DIMENSION ARGS(100)                           QUA 70
EQUIVALENCE (ARGS(1),RC(12501))              QUA 80
C DOUBLE PRECISION USED TO AVOID NOISE IN 8TH DIGIT.    QUA 100
C SLIGHT NOISE MAY BE LEFT DUE TO CONVERSION FROM DP TO SP    QUA 110
DOUBLE PRECISION C,B,BPA,BMA,DELGQ,STORE1,STORE2    QUA 120
C TAKEN FROM SYMBOLIC LISTING PAGE 251 (YELLOW COVER) JULY,1965    QUA 130
C A GOES TO C BECAUSE A DIMENSIONED             QUA 140
10 IF (NARGS.EQ.5) GO TO 20                   QUA 150
CALL ERROR (10)                                QUA 160
RETURN                                         QUA 170
C N MUST BE AN EXACT MULTIPLE OF 4 AND LESS THAN NROW    QUA 180
20 NGQ=INT(ARGS(1))*KIND(1)+IARGS(1)*(1-KIND(1))        QUA 190
IF (MOD(NGQ,4).EQ.0.AND.NGQ.GT.0) GO TO 50           QUA 200
30 CALL ERROR (3)                                QUA 210
RETURN                                         QUA 220
40 CALL ERROR (10)                                QUA 230
RETURN                                         QUA 240
50 IF (NGQ.GT.NROW) GO TO 30                   QUA 250
C RESET NRMAX IF NECESSARY                      QUA 260
NRMAX=MAX0(NGQ,NRMAX)                          QUA 270
CALL ADRESS (4,JPGQ)                           QUA 280
IF (JPGQ) 30,40,60                            QUA 290
60 JPGQ=JPGQ-1                                 QUA 300
CALL ADRESS (5,JWGQ)                           QUA 310
IF (JWGQ) 30,40,70                            QUA 320
70 JWGQ=JWGQ-1                                 QUA 330
IF (NERROR.NE.0) RETURN                         QUA 340
C=ARGS(2)*FLOAT(KIND(2))+FLOAT(IARGS(2)*(1-KIND(2)))  QUA 350
B=ARGS(3)*FLOAT(KIND(3))+FLOAT(IARGS(3)*(1-KIND(3)))  QUA 360
DELGQ=NGQ                                     QUA 370
DELGQ=4.D0*(B-C)/DELGQ                         QUA 380
DO 80 I=1,NGQ,4                                QUA 390
B=C+DELGQ                                     QUA 400
BPA=(B+C)/2.D0                                QUA 410
BMA=(B-C)/2.D0                                QUA 420
K1=I+JPGQ                                     QUA 430
K2=I+JWGQ                                     QUA 440
STORE1=-.861136311594053D0*BMA                QUA 450
STORE2=-.339981043584856D0*BMA                QUA 460
RC(K1)=FDPCON(STORE1+BPA)                     QUA 470
RC(K1+1)=FDPCON(STORE2+BPA)                   QUA 480
RC(K1+2)=FDPCON(BPA-STORE2)                   QUA 490
RC(K1+3)=FDPCON(BPA-STORE1)                   QUA 500
RC(K2)=FDPCON(.347854845137454D0*BMA)       QUA 510
RC(K2+1)=FDPCON(.652145154862546D0*BMA)     QUA 520
RC(K2+2)=RC(K2+1)                             QUA 530
RC(K2+3)=RC(K2)                               QUA 540
80 C=B                                         QUA 550
RETURN                                         QUA 560
END                                         QUA 570

```

```

SUBROUTINE HDIAG (A,N,IEGEN,U,COL,VECTOR,NROW,H)          HDI 10
C VERSION 5.00      HDIAG      5/15/70                  HDI 20
C DIMENSION A(NROW,1), H(54,54), U(54,54), X(54), IQ(54), COL(1), VEHDI 30
C ICTOR(NROW,1), IEGEN(2)                                HDI 40
C MIHDI3, FORTRAN II DIAGONALIZATION OF A REAL SYMMETRIC MATRIX BY HDI 50
C THE JACOBI METHOD.                                     HDI 60
C MAY 19, 1959                                         HDI 70
C CALLING SEQUENCE FOR DIAGONALIZATION                 HDI 80
C     CALL HDIAG( H, N, IEGEN, U, NR)                   HDI 90
C     WHERE H IS THE ARRAY TO BE DIAGONALIZED.          HDI 100
C N IS THE ORDER OF THE MATRIX, H.                      HDI 110
C
C IEGEN MUST BE SET UNEQUAL TO ZERO IF ONLY EIGENVALUES ARE HDI 120
C TO BE COMPUTED.                                      HDI 130
C IEGEN MUST BE SET EQUAL TO ZERO IF EIGENVALUES AND EIGENVECTORS HDI 140
C ARE TO BE COMPUTED.                                  HDI 150
C
C U IS THE UNITARY MATRIX USED FOR FORMATION OF THE EIGENVECTORS. HDI 160
C
C NR IS THE NUMBER OF ROTATIONS.                      HDI 170
C
C A DIMENSION STATEMENT MUST BE INSERTED IN THE SUBROUTINE. HDI 180
C DIMENSION H(N,N), U(N,N), X(N), IQ(N)                HDI 190
C
C
C THE SUBROUTINE OPERATES ONLY ON THE ELEMENTS OF H THAT ARE TO THE HDI 200
C RIGHT OF THE MAIN DIAGONAL.  THUS, ONLY A TRIANGULAR           HDI 210
C SECTION NEED BE STORED IN THE ARRAY H.                  HDI 220
C
C
C DO 10 I=1,N                                         HDI 230
C DO 10 J=1,N                                         HDI 240
C H(I,J)=A(I,J)                                       HDI 250
10 CONTINUE                                              HDI 260
IF (IEGEN(1)) 60,20,60                                 HDI 270
20 DO 50 I=1,N                                         HDI 280
DO 50 J=1,N                                         HDI 290
IF (I-J) 40,30,40                                     HDI 300
30 U(I,J)=1.0                                         HDI 310
GO TO 50                                              HDI 320
40 U(I,J)=0.                                         HDI 330
50 CONTINUE                                              HDI 340
C
60 NR=0                                                 HDI 350
IF (N-1) 440,440,70                                 HDI 360
C
C SCAN FOR LARGEST OFF DIAGONAL ELEMENT IN EACH ROW   HDI 370
C X(I) CONTAINS LARGEST ELEMENT IN ITH ROW            HDI 380
C IQ(I) HOLDS SECOND SUBSCRIPT DEFINING POSITION OF ELEMENT HDI 390
C
70 NM1=N-1                                             HDI 400
DO 90 I=1,NM1                                         HDI 410
X(I)=0.                                                 HDI 420
IPL1=I+1                                              HDI 430
DO 90 J=IPL1,N                                         HDI 440
IF (X(I)-ABS(H(I,J))) 80,80,90                         HDI 450
80 X(I)=ABS(H(I,J))                                    HDI 460
IQ(I)=J                                               HDI 470
90 CONTINUE                                              HDI 480

```

```

C SET INDICATOR FOR SHUT-OFF. RAP=2**-27, NR=NO. OF ROTATIONS      HDI 600
C HDTEST=1.0E37          HDI 610
C RAP=7.45058059E-9      HDI 620
C
C FIND MAXIMUM OF X(I) S FOR PIVOT ELEMENT AND                  HDI 650
C TEST FOR END OF PROBLEM                                     HDI 660
C
100 DO 130 I=1,NM11                                         HDI 670
    IF (I-1) 120,120,110                                     HDI 680
110 IF (XMAX-X(I)) 120,130,130                               HDI 690
120 XMAX=X(I)                                              HDI 700
    IPIV=I                                              HDI 710
    JPIV=IQ(I)                                             HDI 720
130 CONTINUE                                              HDI 730
C
C IS MAX. X(I) EQUAL TO ZERO, IF LESS THAN HDTEST, REVISE HDTEST   HDI 750
    IF (XMAX) 440,440,140                                     HDI 760
140 IF (HDTEST) 160,160,150                                 HDI 770
150 IF (XMAX-HDTEST) 160,160,190                            HDI 780
160 HDIMIN=ABS(H(1,1))                                     HDI 790
    DO 180 I=2,N                                           HDI 800
    IF (HDIMIN-ABS(H(I,I))) 180,180,170                     HDI 810
170 HDIMIN=ABS(H(I,I))                                     HDI 820
180 CONTINUE                                              HDI 830
C
C HDTEST=HDIMIN*RAP                                         HDI 840
C
C RETURN IF MAX.H(I,J)<LESS THAN(2**-27)ABSF(H(K,K)-MIN)       HDI 850
    IF (HDTEST-XMAX) 190,440,440                           HDI 860
190 NR=NR+1                                              HDI 870
C
C COMPUTE TANGENT, SINE AND COSINE,H(I,I),H(J,J)                 HDI 880
    TANG=SIGN(2.0,(H(IPIV,IPIV)-H(JPIV,JPIV)))*H(IPIV,JPIV)/(ABS(H(IPIV,IPIV)-H(JPIV,JPIV))+FSQRT((H(IPIV,IPIV)-H(JPIV,JPIV))**2+4.0*H(IPIV,IPIV)**2))   HDI 890
1V,IPIV)-H(JPIV,JPIV))+FSQRT((H(IPIV,IPIV)-H(JPIV,JPIV))**2+4.0*H(IPIV,IPIV)**2))   HDI 900
2PIV,JPIV)**2))                                              HDI 910
    COSINE=1./FSQRT(1.0+TANG**2)                           HDI 920
    SINE=TANG*COSINE                                         HDI 930
    HII=H(IPIV,IPIV)                                         HDI 940
    H(IPIV,IPIV)=COSINE**2*(HII+TANG*(2.*H(IPIV,JPIV)+TANG*H(JPIV,JPIV))**2+4.0*H(IPIV,IPIV)**2))   HDI 950
1))                                                               HDI 960
    H(JPIV,JPIV)=COSINE**2*(H(JPIV,JPIV)-TANG*(2.*H(IPIV,JPIV)-TANG*H(IPIV,IPIV))**2+4.0*H(IPIV,IPIV)**2))   HDI 970
1I))                                                               HDI 980
    H(IPIV,JPIV)=0.                                         HDI 990
C
C PSEUDO RANK THE EIGENVALUES                                     HDI 1000
C ADJUST SINE AND COS FOR COMPUTATION OF H(IK) AND U(IK)        HDI 1010
    IF (H(IPIV,IPIV)-H(JPIV,JPIV)) 200,210,210               HDI 1020
200 HTEMP=H(IPIV,IPIV)                                         HDI 1030
    H(IPIV,IPIV)=H(JPIV,JPIV)                               HDI 1040
    H(JPIV,JPIV)=HTEMP                                         HDI 1050
C
C RECOMPUTE SINE AND COS                                         HDI 1060
    HTEMP=SIGN(1.0,-SINE)*COSINE                           HDI 1070
    COSINE=ABS(SINE)                                         HDI 1080
    SINE=HTEMP                                              HDI 1090
210 CONTINUE                                              HDI 1100
C
C INSPECT THE IQS BETWEEN I+1 AND N-1 TO DETERMINE             HDI 1110
C WHETHER A NEW MAXIMUM VALUE SHOULD BE COMPUTED SINCE        HDI 1120

```

```

C THE PRESENT MAXIMUM IS IN THE I OR J ROW.          HDI1190
C
C DO 280 I=1,NM11                                  HDI1200
IF (I-IPIV) 230,280,220                            HDI1210
220 IF (I-JPIV) 230,280,230                            HDI1220
230 IF (IQ(I)-IPIV) 240,250,240                            HDI1230
240 IF (IQ(I)-JPIV) 280,250,280                            HDI1240
250 K=IQ(I)                                         HDI1250
      HTEMP=H(I,K)                                     HDI1260
      H(I,K)=0.                                       HDI1270
      IPL1=I+1                                       HDI1280
      X(I)=0.                                         HDI1290
C
C SEARCH IN DEPLETED ROW FOR NEW MAXIMUM          HDI1300
C
C DO 270 J=IPL1,N                                  HDI1310
IF (X(I)-ABS(H(I,J))) 260,260,270                HDI1320
260 X(I)=ABS(H(I,J))                                HDI1330
      IQ(I)=J                                         HDI1340
270 CONTINUE                                         HDI1350
      H(I,K)=HTEMP                                     HDI1360
280 CONTINUE                                         HDI1370
C
      X(IPIV)=0.                                      HDI1380
      X(JPIV)=0.                                      HDI1390
C
C CHANGE THE OTHER ELEMENTS OF H                  HDI1400
C
C DO 410 I=1,N                                  HDI1410
C
IF (I-IPIV) 290,410,330                            HDI1420
290 HTEMP=H(I,IPIV)                                HDI1430
      H(I,IPIV)=COSINE*HTEMP+SINE*H(I,JPIV)        HDI1440
      IF (X(I)-ABS(H(I,IPIV))) 300,310,310        HDI1450
300 X(I)=ABS(H(I,IPIV))                                HDI1460
      IQ(I)=IPIV                                     HDI1470
310 H(I,JPIV)=-SINE*HTEMP+COSINE*H(I,JPIV)        HDI1480
      IF (X(I)-ABS(H(I,JPIV))) 320,410,410        HDI1490
320 X(I)=ABS(H(I,JPIV))                                HDI1500
      IQ(I)=JPIV                                     HDI1510
      GO TO 410                                     HDI1520
C
330 IF (I-JPIV) 340,410,370                            HDI1530
340 HTEMP=H(IPIV,I)                                HDI1540
      H(IPIV,I)=COSINE*HTEMP+SINE*H(I,JPIV)        HDI1550
      IF (X(IPIV)-ABS(H(IPIV,I))) 350,360,360    HDI1560
350 X(IPIV)=ABS(H(IPIV,I))                                HDI1570
      IQ(IPIV)=I                                     HDI1580
360 H(I,JPIV)=-SINE*HTEMP+COSINE*H(I,JPIV)        HDI1590
      IF (X(I)-ABS(H(I,JPIV))) 320,410,410        HDI1600
C
370 HTEMP=H(IPIV,I)                                HDI1610
      H(IPIV,I)=COSINE*HTEMP+SINE*H(JPIV,I)        HDI1620
      IF (X(IPIV)-ABS(H(IPIV,I))) 380,390,390    HDI1630
380 X(IPIV)=ABS(H(IPIV,I))                                HDI1640
      IQ(IPIV)=I                                     HDI1650
390 H(JPIV,I)=-SINE*HTEMP+COSINE*H(JPIV,I)        HDI1660
      IF (X(JPIV)-ABS(H(JPIV,I))) 400,410,410    HDI1670
400 X(JPIV)=ABS(H(JPIV,I))                                HDI1680

```

	IQ(JPIV)=I	HDI1780
410	CONTINUE	HDI1790
C		HDI1800
C	TEST FOR COMPUTATION OF EIGENVECTORS	HDI1810
C		HDI1820
	IF (IEGEN(1)) 100,420,100	HDI1830
420	DO 430 I=1,N	HDI1840
	HTEMP=U(I,IPIV)	HDI1850
	U(I,IPIV)=COSINE*HTEMP+SINE*U(I,JPIV)	HDI1860
430	U(I,JPIV)=-SINE*HTEMP+COSINE*U(I,JPIV)	HDI1870
	GO TO 100	HDI1880
440	IF (IEGEN(2)-2) 450,470,450	HDI1890
450	DO 460 I=1,N	HDI1900
	COL(I)=H(I,I)	HDI1910
460	CONTINUE	HDI1920
	IF (IEGEN(2).NE.3) GO TO 490	HDI1930
470	DO 480 J=1,N	HDI1940
	DO 480 I=1,N	HDI1950
	VECTOR(I,J)=U(I,J)	HDI1960
480	CONTINUE	HDI1970
490	RETURN	HDI1980
	END	HDI1990

```

SUBROUTINE HEADS (LOC,NOO,IN,IO)          HEA 10
C VERSION 5.00     HEADS      5/15/70        HEA 20
C REWRITEN BY S PEAVY   8/8/69        HEA 30
C
C THIS SUBROUTINE INSERTS HEADINGS (IF AVAILABE) OVER THE COLUMNS    HEA 50
C WHEN NO FORMAT IS SPECIFIED        HEA 60
C
C LOC LOCATION WHERE COL NUMBERS ARE        HEA 80
C NOO NO OF COLUMN HEADINGS TO LOOK FOR. NOO LESS THAN OR = 8.    HEA 90
C IN IF IN =0 NEW HEADINGS        HEA 100
C           IF IN =1 PRINT OUT HEADINGS FROM RREVIOUS PAGE    HEA 110
C
C IF A HEADING EXISTS THE 12 CHARACTER HEADING WILL BE PRINTED.    HEA 120
C OTHERWISE THE HEADING COLUMN XXXX IS TO BE USED WHERE XXXX IS THE HEA 140
C NUMBER CONVERTED FOR DECIMAL PRINTOUT. THE HEADINGS ARE PRINTED    HEA 150
C OVER THE DATA WHICH IS IF FORMAT 1PBE15.6        HEA 160
C
C IO =0 PRINT HEADINGS        HEA 170
C IO NOT =0 DO NOT PRINT HEADINGS    HEA 180
C
C COMMON /ABCDEF/ L(48)        HEA 210
C COMMON /HEADER/ NOCARD(80),ITLE(60,6),LNCNT,IPRINT,NPAGE,IPUNCH    HEA 220
C COMMON /FMAT/ IFMTX(6),IOSWT,IFMTS(6),LHEAD(96)        HEA 230
C DIMENSION LOC(1)        HEA 240
C DIMENSION ICOLHD(7)        HEA 250
C DATA ICOLHD(1),ICOLHD(2),ICOLHD(3),ICOLHD(4),ICOLHD(5),ICOLHD(6),IHEA 260
1COLHD(7)/1HC,1HO,1HL,1HU,1HM,1HN,1H /
NO=NOO        HEA 270
IF (NO.GT.8) NO=8        HEA 280
IF (IN.NE.0) GO TO 80        HEA 300
IR=1        HEA 310
DO 70 I=1,NO        HEA 320
CALL PREPAK (5,IND,I,LOC(I),LHEAD(IR))        HEA 330
IF (IND.NE.0) GO TO 10        HEA 340
IR=IR+12        HEA 350
GO TO 70        HEA 360
10 DO 20 IS=1,7        HEA 370
LHEAD(IR)=ICOLHD(IS)        HEA 380
20 IR=IR+1        HEA 390
K=LOC(I)        HEA 400
KC=1000        HEA 410
KD=0        HEA 420
DO 60 IS=1,4        HEA 430
KA=K/KC        HEA 440
IF (KA.NE.0) GO TO 30        HEA 450
IF (KD.NE.0) GO TO 40        HEA 460
LHEAD(IR)=L(45)        HEA 470
GO TO 50        HEA 480
30 KD=1        HEA 490
40 KAP=KA+1        HEA 500
LHEAD(IR)=L(KAP)        HEA 510
50 IR=IR+1        HEA 520
K=K-KA*KC        HEA 530
60 KC=KC/10        HEA 540
LHEAD(IR)=L(45)        HEA 550
IR=IR+1        HEA 560
70 CONTINUE        HEA 570
80 IF (IO.NE.0) RETURN        HEA 580
IS=NO*12        HEA 590

```

```
WRITE (IPRINT,90) (LHEAD(I),I=1,IS)
```

```
HEA 600
```

```
RETURN
```

```
HEA 610
```

```
C
```

```
90 FORMAT (8(3X,12A1))
```

```
HEA 620
```

```
END
```

```
HEA 630
```

```
HEA 640
```

```

SUBROUTINE HISTGM          HIS 10
C   VERSION 5.00      HISTGM    5/15/70      HIS 20
C   WRITTEN BY DAVID HOBGEN, SEL, NBS.  10/24/69.      HIS 30
COMMON/ABCDEF/L(48)        HIS 40
COMMON/BLOCRC/NRC,RC(12600) HIS 50
COMMON/BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,
1 NROW,NCOL,NARGS,VWXYZ(8),NERROR      HIS 60
DIMENSION ARGS(100)        HIS 70
EQUIVALENCE( ARGS(1), RC(12501))      HIS 80
COMMON/BLOCKE/ NAME(4),L1,L2,ISRFLG      HIS 90
COMMON/HEADER/NOCARD(80),ITLE(60,6),LNCNT,IPRINT,NPAGE,IPUNCH      HIS 100
COMMON/SCRAT/NS,NS2,A(13500)      HIS 110
C   HISTOGRAM FOR MIDPOINTS IN COLUMN (C), FREQUENCIES IN COLUMN (C)      HIS 120
C   NHISTOGRAM (C), (C) L2=2 DOES NOT CALL NEW PAGE OR PRINT BLANK      HIS 130
C   LINE BETWEEN CELLS OR HEADING.      HIS 140
C   L2=L2-7      HIS 150
10  IF (NARGS.EQ.2) GO TO 20      HIS 160
CALL ERROR (10)      HIS 170
RETURN      HIS 180
20  CALL ADRESS (1,J1)      HIS 190
IF (J1) 30,40,50      HIS 200
30  CALL ERROR (3)      HIS 210
RETURN      HIS 220
40  CALL ERROR (11)      HIS 230
RETURN      HIS 240
50  CALL ADRESS (2,J2)      HIS 250
IF (J2) 30,40,60      HIS 260
60  IF (NRMAX.GT.0) GO TO 70      HIS 270
CALL ERROR (9)      HIS 280
RETURN      HIS 290
70  IF (NERROR.NE.0) RETURN      HIS 300
80  FORMAT (/ 25X,35HHISTOGRAM FOR FREQUENCIES IN COLUMN,I5,22H, MID-HIS 310
1POINTS IN COLUMN,I5//3X,10HMID-POINTS,7X,9HFREQUENCY/)      HIS 320
85  FORMAT (1X,14A1,2X,I5,3X, 95A1)      HIS 330
90  FORMAT (25X, 95A1)      HIS 340
100 CALL RFORMAT (RC(J1),NRMAX,8,NW1,NDEC1,13,A(1),A(1),0,0)      HIS 350
NBLANK = 15-NW1      HIS 360
IF (L2.EQ.2) GO TO 110      HIS 370
CALL PAGE (4)      HIS 380
WRITE (IPRINT,80) IARGS(1),IARGS(2)      HIS 390
110 LOC1=J1      HIS 400
LOC2=J2      HIS 410
DO 200 I=1,NRMAX      HIS 420
CALL RFORMAT (A(1),1,8,NW1,NDEC1,0,RC(LOC1),A(1),NBLANK,1)      HIS 430
LFREQ = RC(LOC2) + 0.001      HIS 440
IF (LFREQ.GT.0) GO TO 140      HIS 450
WRITE (IPRINT,85) (A(I1),I1=2,15),LFREQ      HIS 460
GO TO 150      HIS 470
140 I2END = MIN0 (LFREQ,95)      HIS 480
WRITE (IPRINT,85) (A(I1),I1=2,15),LFREQ,(L(40),I2=1,I2END)      HIS 490
IF (LFREQ.LE.95) GO TO 150      HIS 500
I3END = LFREQ-95      HIS 510
WRITE (IPRINT,90) (L(40),I3=1,I3END)      HIS 520
150 LOC1 = LOC1+1      HIS 530
200 LOC2 = LOC2+1      HIS 540
RETURN      HIS 550
END      HIS 560
                                HIS 570

```

```

SUBROUTINE IFS          IFS 10
C VERSION 5.00      IFS 20
COMMON /BLOCRC/ NRC,RC(12600)      IFS 30
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NIFS 40
IARGS,VWXYZ(8),NERROR           IFS 50
DIMENSION ARGS(100)             IFS 60
EQUIVALENCE (ARGS(1),RC(12501))   IFS 70
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG  IFS 80
COMMON /BLOCKX/ INDEX(6,8),LEVEL    IFS 90
DIMENSION II(3),K(3),NNN(7)        IFS 100
EQUIVALENCE (I1,II(1)), (I2,II(2)), (I3,II(3))  IFS 110
LOGICAL TWOARG                 IFS 120
C                                     IFS 130
C THIS COMMAND MAY APPEAR ONLY AS A STORED COMMAND.  IFS 140
C                                     IFS 150
C                                     IFS 160
C IFLT, IFEQ, IFGT, IFGE, IFNE, IFLE CORRESPOND TO L2 = 9, 14  IFS 170
C COMPARE   L2 = 15          IFS 175
C                                     IFS 180
C COMMANDS MAY HAVE 2 OR 3 ARGUMENTS (ONLY IFEQ AND IFNE MAY HAVE 3) IFS 190
C ANY ARGUMENT MAY BE OF ANY TYPE, COLUMN NUMBER OR CONSTANT.      IFS 200
C COMPARE MUST HAVE 3 ARGUMENTS          IFS 205
C                                     IFS 210
C IN COMPARE THE TEST IF FOR RELATIVE ERROR AND GOES          IFS 220
C                                     IFS 230
C     I  ARG1-ARG2  I      I      I          IFS 240
C     I  -----  I .LT. I  ARG3  I          IFS 250
C     I  ARG2  I      I      I          IFS 260
C                                     IFS 270
C IF ARG2 OR ARG1 IS 0., THEN ABSOLUTE ERROR WILL BE COMPUTED  IFS 272
C           ABS(ARG2-ARG1) .LT. ARG3 AND INFORMATIVE DIAGNOSTIC  IFS 273
C           WILL BE PRINTED.          IFS 278
C                                     IFS 280
C IF IFEQ AND IFNE CONTAIN A THRID ARGUMENT (TOLERANCE) ABSOLUTE  IFS 282
C ERROR WILL BE COMPUTED          IFS 284
C           ABS(ARG1-ARG2) .LT. ABS (ARG3)          IFS 286
C A GIVEN TOLERANCE IS IGNORED ON IFLT, IFLE, IFGT, IFGE          IFS 290
C EXAMPLES OF HOW COMMANDS READ.          IFS 300
C IFLT 8.32 LT EVERY ENTRY OF COL 34, CONDITION IS TRUE          IFS 310
C IFGE EACH ELEM COL 1 .GE. CORRESP. ELEM. COL 5, COND. IS TRUE  IFS 320
C IFEQ 2. .EQ. 5. CONDITION TRUE (USEFUL WHEN INCREMENTING ARGS. ) IFS 330
C                                     IFS 340
C IF CONDITION IS FALSE, NO ACTION IS TAKEN.          IFS 350
C IF CONDITION IS TRUE, THERE ARE TWO POSSIBILITIES..          IFS 360
C 1. IF THE TEST COMMAND IS THE LAST ONE IN THE REPEAT LOOP  IFS 370
C           CURRENTLY BEING EXECUTED, THE LOOP IS TERMINATED (DROPPED  IFS 380
C           BACK TO THE NEXT OUTER LEVEL IF MORE THAN ONE LEVEL DEEP). IFS 390
C 2. IF THE TEST COMMAND IS NOT THE LAST ONE, ALL THAT HAPPENS IS IFS 400
C           THAT THE REST OF THE LOOP IS NOT PERFORMED. THAT IS, IF THE IFS 410
C           LOOP COUNTER HAS NOT REACHED ITS UPPER LIMIT, IT IS ADVANCED IFS 420
C           ONE AND THE LOOP IS BEGUN FROM THE TOP AGAIN.          IFS 430
C                                     IFS 440
C IF (LEVEL.GT.0) GO TO 10          IFS 450
C CALL ERROR (21)                IFS 460
C GO TO 120                      IFS 470
10 IF(NARGS.EQ.2) IF(L2-15) 40,150,40          IFS 480
IF (NARGS.EQ.3) GO TO 30          IFS 490
CALL ERROR (10)                IFS 500
GO TO 120                      IFS 510

```

```

20 CALL ERROR (11) IFS 520
GO TO 120 IFS 530
30 IF(L2.EQ.10.OR.L2.EQ.13.OR.L2.EQ.15) GO TO 40 IFS 540
CALL ERROR (212) IFS 550
NARGS=2 IFS 560
40 DO 60 I=1,NARGS IFS 570
CALL ADDRESS (I,II(I)) IFS 580
IF (II(I)) 50,20,60 IFS 590
50 II(I)=-II(I) IFS 600
60 K(I)=1-KIND(I) IFS 610
IF(NRMAX.NE.0.OR.KIND(1) +KIND(2).EQ.2) IF (NERROR) 120,65,120 IFS 615
CALL ERROR (9) IFS 620
GO TO 120 IFS 630
65 NNN(4)=0 IFS 640
NNN(5)=0 IFS 650
NNN(6)=0 IFS 660
TWOARG=NARGS.EQ.2 IFS 670
DO 110 I=1,NRMAX IFS 680
IF (TWO ARG) IF (RC(I1)-RC(I2)) 75,77,80 IFS 685
C CHECK EQ,NE WITHIN BOUNDS IFS 690
T=ABS(RC(I1)-RC(I2)) IFS 692
IF(L2.NE.15) GO TO 66 IFS 694
IF(RC(I1).NE.0.0.AND.RC(I2).NE.0.0) GO TO 63 IFS 698
CALL ERROR (108) IFS 700
GO TO 66 IFS 702
63 T=ABS(T/RC(I2)) IFS 704
66 IF(T-ABS(RC(I3)))70,80,80 IFS 706
70 NNN(5)=NNN(5)+1 IFS 710
GO TO 90 IFS 720
C CHECK IFS WITHOUT BOUNDS IFS 730
75 NNN(4)=NNN(4)+1 IFS 740
GO TO 100 IFS 750
77 NNN(5)=NNN(5)+1 IFS 760
GO TO 100 IFS 770
80 NNN(6)=NNN(6)+1 IFS 780
90 I3=I3+K(3) IFS 790
100 I1=I1+K(1) IFS 800
110 I2=I2+K(2) IFS 810
NNN(1)=NNN(5)+NNN(6) IFS 820
NNN(2)=NNN(4)+NNN(6) IFS 830
NNN(7)=NNN(2) IFS 835
NNN(3)=NNN(4)+NNN(5) IFS 840
IF(NNN(L2-8).EQ.0) IF(INDEX(2,LEVEL)-INDEX(3,LEVEL)) 130,130,140 IFS 845
120 RETURN IFS 850
C IFS 860
C IF-COMMAND NOT AT END OF PERFORM LOOP, ADVANCE LOOP COUNT. IFS 870
C IFS 880
130 INDEX(2,LEVEL)=INDEX(3,LEVEL)+1 IFS 890
GO TO 120 IFS 900
C IFS 910
C IF-COMMAND IS AT END OF PERFORM LOOP, TERMINATE LOOP. IFS 920
C IFS 930
140 LEVEL=LEVEL-1 IFS 940
GO TO 120 IFS 950
150 CALL ERROR (10) IFS 960
RETURN IFS 970
END IFS 980

```

	SUBROUTINE INFERR(I)	INF	10
C	VERSION 5.00 INFERR 5/15/70	INF	20
C	INFORMATIVE DIAGNOSTICS 200 AND UP	INF	30
	COMMON/BLOCKC/KIO,INUNIT,ISCRAT,KBDOUT,KRDKNT,LLIST	INF	40
	COMMON/CONLB2/ER,ISIGD	INF	45
	ISCRUN=ISCRAT	INF	50
	WRITE(ISCRUN,500)	INF	60
500	FORMAT(/41H* INFORMATIVE DIAGNOSTIC IN ABOVE COMMAND,43X)	INF	70
	II=I-200	INF	80
	GO TO (401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412,	INF	90
1	413,414,415,416,417,418,419,420,	INF	100
2	421,422,423,424,425,426,427,428,429,430,431,432,433,434,435,	INF	105
3	436,437),II	INF	110
401	WRITE(ISCRUN,201)	INF	120
201	FORMAT(52H* TOO MUCH DATA IN SET, READ OR GENERATE, SPILL LOST,	INF	130
	132X)	INF	140
	GO TO 900	INF	150
402	WRITE(ISCRUN,202)	INF	160
202	FORMAT(61H* COMMAND NOT ALLOWED IN REPEAT MODE. EXECUTED BUT NOT SINF	170	
	1TORED,23X)	INF	180
	GO TO 900	INF	190
403	WRITE( ISCRUN, 203 )	INF	200
203	FORMAT(61H* VALUE REQUESTED IN SHORTEN, ACOALESCE OR AAVERAGE NOT	INF	210
	1FOUND,23X)	INF	220
	GO TO 900	INF	230
404	WRITE( ISCRUN, 204 )	INF	240
204	FORMAT(32H* BAD HEAD. COLUMN GT 50 OR NO /,52X)	INF	250
	GO TO 900	INF	260
405	WRITE( ISCRUN, 205 )	INF	270
205	FORMAT(68H* THIS COMMAND WAS NOT EXECUTED BECAUSE ITS MEANING WAS	INF	280
	1QUESTIONABLE,16X)	INF	290
	GO TO 900	INF	300
406	WRITE(ISCRUN,206)	INF	310
206	FORMAT (24H* F LESS THAN 0, SET = 0,60X)	INF	320
	GO TO 900	INF	330
407	WRITE(ISCRUN,207)	INF	340
207	FORMAT(24H* NU1 OR NU2 LESS THAN 1,60X)	INF	350
	GO TO 900	INF	360
408	WRITE(ISCRUN,208)	INF	370
208	FORMAT(33H* NU1 OR NU2 TRUNCATED TO INTEGER,51X)	INF	380
	GO TO 900	INF	390
409	WRITE(ISCRUN,209)	INF	400
209	FORMAT(34H* IMPROPER TITLE NUMBER, ASSUMED 1,50X)	INF	410
	GO TO 900	INF	420
410	WRITE(ISCRUN,210)	INF	430
210	FORMAT(54H* NO OF ROWS NOT = TO COLS. MATRIX USED LARGEST SQUARE,	INF	440
	1 20X)	INF	450
	GO TO 900	INF	460
411	WRITE(ISCRUN,211)	INF	470
211	FORMAT(52H* ASTERISK STRING IMPLYING ,THRU, INCORRECT, IGNORED,	INF	480
	1 32X)	INF	490
	GO TO 900	INF	500
412	WRITE(ISCRUN,212)	INF	510
212	FORMAT(43H* UNNECESSARY ARGUMENTS IN COMMAND IGNORED.,41X)	INF	520
	GO TO 900	INF	530
413	WRITE(ISCRUN,213)	INF	540
213	FORMAT(27H* PARTIAL STORAGE OF MATRIX,57X)	INF	550
	GO TO 900	INF	560
414	WRITE (ISCRUN,214)	INF	570

214	FORMAT(28H***INSUFFICIENT SCRATCH AREA,56X)	INF 580
	GO TO 900	INF 585
415	WRITE (ISCRUN,215)	INF 590
215	FORMAT (48H* NRMAX IS NOT LARGE ENOUGH TO ALLOW ITERATION ,36X)	INF 600
	GO TO 900	INF 610
416	WRITE (ISCRUN,216)	INF 620
216	FORMAT(68H* 1ST COLUMN OF ISETUP OR ISOLATE IS NOT MONOTONIC OR ISINF 1 CONSTANT. ,16X)	INF 630
	GO TO 900	INF 640
417	WRITE (ISCRUN,217)	INF 650
217	FORMAT (34H* ITERATION HAS FOUND NO VALUES. ,50X)	INF 670
	GO TO 900	INF 680
418	WRITE (ISCRUN,218)	INF 690
218	FORMAT(81H* WORKSHEET IS TOO SHORT TO ACCOMMODATE ALL THE VALUES GINF 1GENERATED BY THIS COMMAND. ,3X )	700
	GO TO 900	INF 710
419	WRITE (ISCRUN,219)	INF 720
219	FORMAT (30H* MAXMIN HAS FOUND NO EXTREMA. ,54X)	INF 740
	GO TO 900	INF 750
420	WRITE (ISCRUN,220)	INF 760
220	FORMAT (84H* MAXMIN HAS FOUND AND IGNORED A TRIAD OF X,S WITH AT LINF 1EAST TWO IDENTICAL VALUES. )	770
	GO TO 900	INF 780
421	WRITE (ISCRUN,221)	INF 790
221	FORMAT(59H* MORE THAN ONE ARGUMENT IN COMMAND. ONLY FIRST ONE IS UINF 1SED,25X)	810
	GO TO 900	INF 820
422	WRITE (ISCRUN,222)	INF 830
222	FORMAT (43H* FORMAT NOT FOUND. READABLE FORMAT IS USED, 41X)	INF 840
	GO TO 900	INF 850
423	WRITE (ISCRUN,223)	INF 860
223	FORMAT(38H* ONE,SOME OR ALL WEIGHTS ARE NEGATIVE,46X)	INF 870
	GO TO 900	INF 880
424	WRITE (ISCRUN,224)	INF 890
224	FORMAT(48H* ALL WEIGHTS ARE ZERO. COMMAND IS NOT EXECUTED,36X)	INF 900
	GO TO 900	INF 910
425	WRITE (ISCRUN,225)	INF 920
225	FORMAT(81H* ARG FOR BESIN,BESJN,BESKN GIVES A RESULT TOO LARGE/SMAINF 1LL. COMMAND NOT EXECUTED.,3X)	940
	GO TO 900	INF 950
426	WRITE (ISCRUN,226)	INF 960
226	FORMAT(73H* COLUMN NOT LONG ENOUGH TO STORE ALL ELEMENTS. ONLY NROINF 1W WILL BE STORED.,11X )	980
	GO TO 900	INF 990
427	WRITE (ISCRUN,227)	INF1000
227	FORMAT(78H* NOT ENOUGH DATA ON COL TO RESTORE MATRIX/ARRAY. DATA AINF1020 1AVAILABLE WILL BE USED.,6X)	1025
	GO TO 900	INF1030
428	WRITE (ISCRUN,228)	INF1040
228	FORMAT(84H* SUM OF SQRS DO NOT ADD UP-ABS. VALUE OF (TOTAL-ROW-COLINF1050 1-RES.)/TOTAL EXCEEDS 5.E-7 )	1050
	GO TO 900	INF1060
429	WRITE (ISCRUN,229)	INF1070
229	FORMAT(51H* MORE THAN 50 HEAD COLUMN COMMANDS HAVE BEEN USED.,33X)	INF1090
	GO TO 900	INF1100
430	WRITE (ISCRUN,230)	INF1110
230	FORMAT (72H* ATTEMPT TO PROMOTE FROM BELOW NRMAX. FIRST ARGUMENT IINF1120 1S RESET TO NRMAX., 12X)	1130
	GO TO 900	INF1140

431	WRITE (ISCRUN,231)	INF1150
231	FORMAT (53H* ATTEMPT TO DEMOTE OFF THE WORKSHEET. SPILL IS LOST., 131X)	INF1160 INF1170
	GO TO 900	INF1180
432	GO TO 900	INF1220
433	WRITE (ISCRUN,233)	INF1230
233	FORMAT(76H* NEGATIVE VALUE(S) WERE ENCOUNTERED BY PARTITION FUNCTI ON. ZEROES STORED. ,8X)	INF1240 INF1250
	GO TO 900	INF1260
434	WRITE (ISCRUN,234)	INF1270
234	FORMAT( 45H* NEGATIVE ABSOLUTE TEMPERATURES CONVERTED. ,39X)	INF1280
	GO TO 900	INF1290
435	WRITE (ISCRUN,235)	INF1300
235	FORMAT(76H* CAUTION, USE EXPERIMENTALLY ONLY. NOT OPTIMUM IN ORDE 1R TO MAKE IT MACHINE,10X/84H INDEPENDENT. REFERENCES - J.B. KRUS 2KAL,ACM,12,92. AND J.H. HALTON,SIAM REV.,12,1.)	INF1310 INF1320 INF1330
	GO TO 900	INF1340
436	WRITE (ISCRUN,236)	INF1350
236	FORMAT (78H* COMMAND IGNORED - S BEFORE COMMAND NAME MEANINGLESS I 1F NO STORAGE REQUESTED.)	INF1360 INF1370
	GO TO 900	INF1380
437	WRITE(ISCRUN,237) ISIGD	INF1390
237	FORMAT(63H* NUMBER OF SIGNIFICANT DIGITS AFTER DECIMAL PT HAS BEEN 1 SET TO ,I3,18X)	INF1400 INF1405
900	RETURN	INF1410
	END	INF1420

```

SUBROUTINE INPUT          INP   10
C VERSION 5.00      INPUT    5/15/70      INP   20
COMMON /BLOCKA/ MODE,M,KARD(83),KARG,ARG,ARG2,NEWCD(80),KRDEND      INP   30
COMMON /BLOCKB/ NSTMT,NSTMTX,NSTMTH,NCOM,LCOM,IOVFL,COM(2000)      INP   40
COMMON /BLOCKC/ KIO,INUNIT,ISCRAT,KBDOUT,KRDKNT,LLIST      INP   50
C
C THIS ROUTINE HANDLES THE READING OF INPUT RECORDS.      INP   60
C IF KIO = 0, INPUT IS CARD IMAGE FROM CARD READER OR TAPE.      INP   70
C IF KIO = 1, INPUT IS REAL-TIME FROM A KEYBOARD.      INP   80
C
C KRDKNT=KRDKNT+1      INP  100
IF (KIO.EQ.0) GO TO 20      INP  110
IF (MODE.EQ.3) GO TO 10      INP  120
WRITE (KBDOUT,30)      INP  130
GO TO 20      INP  140
10 WRITE (KBDOUT,40) NSTMT      INP  150
GO TO 20      INP  160
20 READ (INUNIT,50) NEWCD      INP  170
KARD(1)=0      INP  180
KARD(2)=0      INP  190
KARD(KRDEND+3)=46      INP  200
CALL OMCONV (NEWCD,KARD(3),KRDEND)      INP  210
RETURN      INP  220
C
30 FORMAT (9H READY      )      INP  230
40 FORMAT (9H READY      ,I3,3H / )      INP  240
50 FORMAT (80A1)      INP  250
END      INP  260
INP  270
INP  280

```

```

SUBROUTINE INTERP                                INT   10
C      VERSION 5.00     INTERP      5/15/70      INT   20
C      *
C      GENERAL FORM OF COMMAND IS                INT   30
C      INTERPOLATE X IN COL ++ Y IN COL 33 LENGTH=,, FOR THE FIRST    INT   40
C      ,, VALUES OF XP IN COL ++ USE ,, POINTS STORE IN COL 33      INT   50
C      *
C      COMMON /BLOCRC/ NRC,RC(12600)             INT   60
C      COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NINT 90
C      IARGS,VWXYZ(8),NERROR                   INT 100
C      COMMON /BLOCKC/ KIO,INUNIT,ISCRAT,KBDOUT,KRDKN,LLIST           INT 110
C      DIMENSION ARGS(100)                      INT 120
C      EQUIVALENCE (ARGS(1),RC(12501))          INT 130
C      COMMON /SCRAT/ NS,NS2,A(13500)            INT 140
C      *
C      CHECK TO SEE IF WE HAVE CORRECT NUMBER AND MODE OF ARGUMENTS  INT 150
C      *
C      IF (NARGS.NE.7) CALL ERROR (10)          INT 160
C      J=NARGS                                 INT 170
C      CALL CKIND (J)                          INT 180
C      IF (J.NE.0) CALL ERROR (3)              INT 190
C      *
C      CHECK NO. OF POINTS LESS THAN OR EQUAL TO NRMAX AND POSITIVE  INT 200
C      *
C      IF (IARGS(3).LT.0.OR.IARGS(4).LT.0) CALL ERROR (3)          INT 210
C      IF (IARGS(3).GT.NROW.OR.IARGS(4).GT.NRMAX) CALL ERROR (3)  INT 220
C      *
C      CHECK TO SEE IF WE HAVE MORE THAN TWO ENTRIES IN TABLE        INT 230
C      COMPUTE COLUMN ADDRESSES                  INT 240
C      *
C      IF (IARGS(3).LT.2) CALL ERROR (3)          INT 250
C      LXY=IARGS(3)                            INT 260
C      LXP=IARGS(4)                            INT 270
C      IARGS(3)=IARGS(5)                        INT 280
C      IARGS(4)=IARGS(7)                        INT 290
C      NARGS=4                                INT 300
C      CALL CHKCOL (J)                         INT 310
C      IF (J.NE.0) CALL ERROR (11)              INT 320
C      *
C      CHECK TO SEE IF WE EXCEED SCRATCH AREA  INT 330
C      CHECK FOR PREVIOUS ERRORS               INT 340
C      *
C      INDRV=0                                INT 350
C      IF (IARGS(6)-LXY) 20,20,10              INT 360
10     IARGS(6)=LXY                           INT 370
C      INDRV=1                                INT 380
C      *
C      CHECK TO SEE IF WE EXCEED SCRATCH AREA  INT 390
C      CHECK FOR PREVIOUS ERRORS               INT 400
C      *
C      INDRV=0                                INT 410
C      IF (IARGS(6)-LXY) 20,20,10              INT 420
C      IARGS(6)=LXY                           INT 430
C      INDRV=1                                INT 440
20     IF (IARGS(6)**2+3*IARGS(6)+LXP.LE.NS) GO TO 30  INT 450
C=1.0-4.0*FLOAT(NS-LXP)                     INT 460
C      I=FSQRT(C)                            INT 470
C      IARGS(6)=(-1+I)/2                      INT 480
C      INDRV=2                                INT 490
30     IA1=IARGS(1)                           INT 500
C      IA2=IARGS(2)                           INT 510
C      IA3=IARGS(3)                           INT 520
C      IA4=3*IARGS(6)+LXP+1                  INT 530
C      CALL INTRP (RC(IA1),RC(IA2),LXY,RC(IA3),A(1),LXP,IARGS(6),A(LXP+1))  INT 540
C      1,A(IA4),IND                           INT 550
C      STORE RESULTS                         INT 560
C      IA3=IARGS(4)                           INT 570
C      *
C      INDRV=0                                INT 580
C      IA3=IARGS(4)                           INT 590

```

```
DO 40 I=1,LXP           INT 600
RC(IA3)=A(I)             INT 610
IA3=IA3+1                INT 620
40 CONTINUE                INT 630
IF (INDRV.EQ.0.AND.IND.EQ.0) RETURN    INT 640
IF (INDRV-1) 70,50,60          INT 650
50 WRITE (ISCRAT,80)          INT 660
GO TO 70                  INT 670
60 WRITE (ISCRAT,90)          INT 680
70 IF (IND.EQ.0) RETURN      INT 690
WRITE (ISCRAT,100)          INT 700
RETURN                     INT 710
C
80 FORMAT (6X,20(1H+),39HORDER OF INTERPOLATION EQUALS LIST SIZE,19X)INT 730
90 FORMAT (6X,20(1H+),53HORDER OF INTERP WAS RESET DUE TO SIZE OF SCRINT 740
1ATCH AREA,5X)               INT 750
100 FORMAT (6X,20(1H+),42HEXTRAPOLATION DONE FOR MORE THAN ONE DELTA,1INT 760
16X)                         INT 770
END                         INT 780
```

```

SUBROUTINE INTRP (X,Y,NLIST,X1,RESULT,NX1,NORD,S,SA,IND)          INR 10
C VERSION 5.00      INTRP      5/15/70                           INR 20
C CALLING SEQUENCE                                         INR 30
C SUBROUTINE INTRP(X,Y,NLIST,X1,RESULT,NX1,NORD,S,SA,IND)          INR 40
C
C X  THE INDEPENDENT VALUE OF THE TABLE. MUST BE IN ASCENDING OR   INR 50
C     DESCENDING ORDER. NEED NOT BE EVENLY SPACED                  INR 60
C Y  THE DEPENDENT VALUE OF THE TABLE                            INR 70
C NLIST  LENGTH OF X OR Y                                     INR 80
C X1    VALUES TO BE INTERPOLATED                            INR 90
C RESULT RESULT FROM INTERPOLATION                         INR 100
C NX1    LENGTH OF X1 VECTOR                                INR 110
C NORD   ORDER OF INTERPOLATION                           INR 120
C S      SCRATCH AREA  S(3*NORD)                          INR 130
C SA     SCRATCH AREA  SA(NORD,NORD)                      INR 140
C IND    INDICATOR                                         INR 150
C           IND=0  EVERYTHING FINE                         INR 160
C           IND=2  EXTRAPOLATION AND MORE THEN ONE DELTA    INR 170
C
C
C DIMENSION X(1), Y(1), X1(1), RESULT(1), S(1), SA(NORD,NORD)      INR 180
C INDA=0                                         INR 190
C IND=0                                         INR 200
C NRD=NORD-1                                    INR 210
C NDIR=1                                         INR 220
C IF (X(1).GE.X(2)) NDIR=2                      INR 230
C I=1                                           INR 240
C DO 220 II=1,NX1                               INR 250
C IC=0                                         INR 260
C XA=X1(II)                                    INR 270
C GO TO (10,180), NDIR                         INR 280
C
10  IF (XA-X(1)) 20,160,30                     INR 290
C
20  IF (ABS(XA-X(1)).GT.ABS(X(1)-X(2))) INDA=2      INR 300
C IC=1                                         INR 310
C IA=1                                         INR 320
C GO TO 80                                      INR 330
C
30  DO 40 IA=I,NLIST                           INR 340
C     IF (X(IA)-XA) 40,170,60                   INR 350
C
40  CONTINUE                                     INR 360
C
50  IF (ABS(X(NLIST)-XA).GT.ABS(X(NLIST)-X(NLIST-1))) INDA=2      INR 370
C IE=1                                         INR 380
C IA=NLIST-NRD                                 INR 390
C IC=1                                         INR 400
C GO TO 80                                      INR 410
C
60  IA=IA-1                                     INR 420
C     IF (X(IA)-XA) 70,170,60                   INR 430
C
70  IF (IA+NRD.LE.NLIST) GO TO 80               INR 440
C     IC=1                                         INR 450
C     IA=NLIST-NRD                                 INR 460
C
80  IF (NRD.GT.1) GO TO 90                      INR 470
C     TEMP=(XA-X(IA))/(X(IA+1)-X(IA))          INR 480
C     RESULT(II)=Y(IA)+(Y(IA+1)-Y(IA))*TEMP      INR 490
C     GO TO 220                                     INR 500
C
90  IF (IC.NE.0) GO TO 100                      INR 510
C     IA=IA-NRD/2                                INR 520
C     IF (IA.LE.0) IA=1                           INR 530
C
100 NA=IA+NRD                                  INR 540
C     PROD=1.                                       INR 550
C     IZ=1                                         INR 560
C     IZA=NRD+1                                   INR 570
C

```

	DO 110 IB=IA,NA	INR 600
	S(IZ)=X(IB)	INR 610
	S(IZA)=XA-X(IB)	INR 620
	PROD=PROD*S(IZA)	INR 630
	IZ=IZ+1	INR 640
110	IZA=IZA+1	INR 650
	NB=NRD+1	INR 660
	DO 120 IAR=2,NB	INR 670
	DO 120 IBR=IAR,NB	INR 680
	SA(IBR-1,IAR-1)=S(IAR-1)-S(IBR)	INR 690
120	SA(IAR-1,IBR)=(-SA(IBR-1,IAR-1))	INR 700
	IZB=IZA	INR 710
	IZC=NORD+1	INR 720
	DO 140 IAR=1,NB	INR 730
	SUM=S(IZC)	INR 740
	DO 130 IBR=1,NRD	INR 750
130	SUM=SUM*SA(IBR,IAR)	INR 760
	S(IZA)=PROD/SUM	INR 770
	IZC=IZC+1	INR 780
140	IZA=IZA+1	INR 790
	R=0.0	INR 800
	IAX=IA	INR 810
	DO 150 IX=1,NORD	INR 820
	R=R+S(IZB)*Y(IAX)	INR 830
	IAX=IAX+1	INR 840
150	IZB=IZB+1	INR 850
	RESULT(II)=R	INR 860
	GO TO 220	INR 870
160	RESULT(II)=Y(1)	INR 880
	IA=1	INR 890
	GO TO 220	INR 900
170	RESULT(II)=Y(IA)	INR 910
	GO TO 220	INR 920
180	IF (XA-X(1)) 190,160,30	INR 930
190	DO 200 IA=I,NLIST	INR 940
	IF (XA-X(IA)) 200,170,210	INR 950
200	CONTINUE	INR 960
	GO TO 50	INR 970
210	IA=IA-1	INR 980
	IF (XA-X(IA)) 70,170,210	INR 990
220	I=IA	INR1000
	IND=IND+INDA	INR1010
	RETURN	INR1020
	END	INR1030

```

SUBROUTINE INVCHK (A,M,N,AINV,M1,Y,L2,ERR,IND)           INK 10
C VERSION 5.00      INVCHK      5/15/70                  INK 20
C INVCHK FOR OMNITAB UNIVAC 1108  S. PEAVY 5/24/67      INK 30
C THIS SUBROUTINE INVERTS A MATRIX AND PROVIDES ALL THE CHECKS DESCRIPTIVE
C IN PAC-1          INK 40
C                                     INK 50
C                                     INK 60
C A IS THE MATRIX TO BE INVERTED                      INK 70
C                                     INK 80
C M IS THE SIZE OF A AS DIMENSIONED IN THE CALLING PROGRAM A(M,M) INK 90
C                                     INK 100
C N IS THE SIZE OF A TO BE INVERTED                      INK 110
C     N LESS THAN OR =M-1                                INK 120
C                                     INK 130
C AINV WILL CONTAIN THE INVERTED MATRIX IF INVERSION IS OBTAINABLE INK 140
C                                     INK 150
C M1 IS THE SIZE OF AINV AS DIMENSIONED IN THE CALLING PROGRAM INK 160
C     AINV(M1,2*M1)      M1 MUST BE GREATER OR =N+1        INK 170
C     AINV MUST HAVE TWICE AS MANY COLUMNS AS ROWS          INK 180
C     A AND AINV CANNOT BE SAME OR EQUIVALENT              INK 190
C                                     INK 200
C ERR WILL CONTAIN THE 3 WAYS OF EVALUATING NORM CHECKS       INK 210
C     ERR IS A DIMENSIONED AS ERR(3)                      INK 220
C                                     INK 230
C IND IS AN INDICATOR                                     INK 240
C     IND=0 MATRIX INVERTED AND ERROR CHECKS MADE        INK 250
C     IND=1 MATRIX SINGULAR                               INK 260
C                                     INK 270
C COLUMN AINV(N+1,I)   I=1,...,N WILL CONTAIN THE ERROR BOUND OF INK 280
C THE SUM CHECKS+1.                                      INK 290
C                                     INK 300
C DIMENSION A(M,M), AINV(M1,M1), ERR(3), ANORM(2,3)        INK 310
C DIMENSION Y(N)                                         INK 320
C DATA ZERO/0.0/,ONE/1.0/                                INK 330
10  NA=N                                              INK 340
    DO 20 I=1,NA                                       INK 350
    DO 20 J=1,NA                                       INK 360
20  AINV(J,I)=A(J,I)                                 INK 370
    NB=NA                                             INK 380
    IF (L2.EQ.1) GO TO 40                           INK 390
    NB=NB+1                                           INK 400
    DO 30 I=1,NA                                       INK 410
    AINV(I,NA+1)=Y(I)                                 INK 420
30  AINV(NA+1,I)=ZERO                                INK 430
    AINV(NA+1,NA+1)=-ONE                            INK 440
    NA=NA+1                                           INK 450
40  DO 60 I=1,NA                                       INK 460
    SUM=ZERO                                         INK 470
    AINV(NA+1,I)=ZERO                                INK 480
    DO 50 J=1,NA                                       INK 490
50  SUM=SUM+AINV(I,J)                                INK 500
60  AINV(I,NA+1)=-SUM                                INK 510
    AINV(NA+1,NA+1)=ONE                             INK 520
    NB=NB+1                                           INK 530
    CALL SPINV (AINV,NB,M1,IND)                      INK 540
    IF (IND.NE.0) RETURN                            INK 550
    DO 140 K=1,2                                       INK 560
    DO 70 I=1,3                                       INK 570
70  ANORM(K,I)=ZERO                                INK 580
    DO 130 I=1,N                                       INK 590

```

```

SUM=ZERO           INK 600
DO 120 J=1,N      INK 610
GO TO (80,90), K INK 620
80 TEMP=ABS(AINV(I,J)) INK 630
GO TO 110         INK 640
90 TEMP=ZERO       INK 650
DO 100 L=1,N      INK 660
100 TEMP=TEMP+A(I,L)*AINV(L,J) INK 670
IF (I.EQ.J) TEMP=ONE-TEMP      INK 680
TEMP=ABS(TEMP)             INK 690
110 ANORM(K,1)=ANORM(K,1)+TEMP**2 INK 700
IF (ANORM(K,2).LT.TEMP) ANORM(K,2)=TEMP INK 710
120 SUM=SUM+TEMP          INK 720
IF (ANORM(K,3).LT.SUM) ANORM(K,3)=SUM INK 730
130 CONTINUE            INK 740
ANORM(K,1)=FSQRT(ANORM(K,1)) INK 750
140 ANORM(K,2)=FLOAT(N)*ANORM(K,2) INK 760
DO 150 K=1,3          INK 770
150 ERR(K)=(ANORM(1,K)*ANORM(2,K))/(1.-ANORM(2,K)) INK 780
RETURN              INK 790
END                 INK 800

```

```

SUBROUTINE INVCOR (A,M,N,AINV,M1,Y,L2,ERR,IND)           INC 10
C VERSION 5.00      INVCOR      5/15/70                 INC 20
C INVCOR FOR OMNITAB UNIVAC 1108 S. PEAVY 5/24/67       INC 30
C THIS SUBROUTINE INVERTS A MATRIX AND PROVIDES ALL THE CHECKS DESCRIPTIVE
C IN PAC-1                                              INC 40
C                                              INC 50
C                                              INC 60
C A IS THE MATRIX TO BE INVERTED                         INC 70
C                                              INC 80
C M IS THE SIZE OF A AS DIMENSIONED IN THE CALLING PROGRAM A(M,M) INC 90
C                                              INC 100
C N IS THE SIZE OF A TO BE INVERTED                      INC 110
C     N LESS THAN OR =M-1                               INC 120
C                                              INC 130
C AINV WILL CONTAIN THE INVERTED MATRIX IF INVERSION IS OBTAINABLE INC 140
C                                              INC 150
C M1 IS THE SIZE OF AINV AS DIMENSIONED IN THE CALLING PROGRAM INC 160
C     AINV(M1,2*M1)          M1 MUST BE GREATER OR =N+1      INC 170
C     AINV MUST HAVE TWICE AS MANY COLUMNS AS ROWS          INC 180
C     A AND AINV CANNOT BE SAME OR EQUIVALENT              INC 190
C                                              INC 200
C ERR WILL CONTAIN THE 3 WAYS OF EVALUATING NORM CHECKS    INC 210
C     ERR IS A DIMENSIONED AS ERR(3)                      INC 220
C                                              INC 230
C IND IS AN INDICATOR                                     INC 240
C     IND=0 MATRIX INVERTED AND ERROR CHECKS MADE        INC 250
C     IND=1 MATRIX SINGULAR                                INC 260
C                                              INC 270
C COLUMN AINV(N+1,I)   I=1,...,N WILL CONTAIN THE ERROR BOUND OF INC 280
C THE SUM CHECKS+1.                                         INC 290
C                                              INC 300
C DIMENSION A(M,M), AINV(M1,M1), ERR(3), ANORM(2,3)        INC 310
C DIMENSION Y(N)                                         INC 320
C DATA ZERO/0.0/,ONE/1.0/                                INC 330
10  NA=N                                              INC 340
    DO 20 I=1,NA                                         INC 350
    DO 20 J=1,NA                                         INC 360
20  AINV(J,I)=A(J,I)                                    INC 370
    NB=NA                                              INC 380
    IF (L2.EQ.1) GO TO 40                                INC 390
    NB=NB+1                                            INC 400
    DO 30 I=1,NA                                         INC 410
    AINV(I,NA+1)=Y(I)                                    INC 420
30  AINV(NA+1,I)=ZERO                                  INC 430
    AINV(NA+1,NA+1)=-ONE                                INC 440
    NA=NA+1                                            INC 450
40  DO 60 I=1,NA                                         INC 460
    SUM=ZERO                                           INC 470
    AINV(NA+1,I)=ZERO                                  INC 480
    DO 50 J=1,NA                                         INC 490
50  SUM=SUM+AINV(I,J)                                    INC 500
60  AINV(I,NA+1)=-SUM                                 INC 510
    AINV(NA+1,NA+1)=ONE                                INC 520
    NB=NB+1                                            INC 530
    CALL CSPINV (AINV,NB,M1,IND)                         INC 540
    IF (IND.NE.0) RETURN                                INC 550
    DO 140 K=1,2                                         INC 560
    DO 70 I=1,3                                         INC 570
70  ANORM(K,I)=ZERO                                 INC 580
    DO 130 I=1,N                                         INC 590

```

```

SUM=ZERO           INC 600
DO 120 J=1,N      INC 610
GO TO (80,90), K INC 620
80 TEMP=ABS(AINV(I,J)) INC 630
GO TO 110         INC 640
90 TEMP=ZERO       INC 650
DO 100 L=1,N      INC 660
100 TEMP=TEMP+A(I,L)*AINV(L,J) INC 670
IF (I.EQ.J) TEMP=ONE-TEMP     INC 680
TEMP=ABS(TEMP)        INC 690
110 ANORM(K,1)=ANORM(K,1)+TEMP**2 INC 700
IF (ANORM(K,2).LT.TEMP) ANORM(K,2)=TEMP INC 710
120 SUM=SUM+TEMP        INC 720
IF (ANORM(K,3).LT.SUM) ANORM(K,3)=SUM INC 730
130 CONTINUE          INC 740
ANORM(K,1)=FSQRT(ANORM(K,1)) INC 750
140 ANORM(K,2)=FLOAT(N)*ANORM(K,2) INC 760
DO 150 K=1,3        INC 770
150 ERR(K)=(ANORM(1,K)*ANORM(2,K))/(1.-ANORM(2,K)) INC 780
RETURN             INC 790
END                INC 800

```

```

SUBROUTINE INVERT                                INV   10
C VERSION 5.00      INVERT      5/15/70          INV   20
C MATRIX INVERSION, SOLUTION OF SYSTEM OF EQUATIONS    INV   30
C S PEAVY 5/22/67          INV   40
C MINVERT (+++,++) SIZE +++,++ STORE (+++,++)
C SOLVE   (+++,++,) SIZE +++,++ Y VECTOR +++ STORE +++
C LARGEST MATRIX TO BE INVERTED OR SYSTEM TO BE SOLVED IS 50    INV   50
C                                         INV   70
C                                         INV   90
C                                         INV  100
C L2=1  INVERT          INV  110
C L2=2  SOLVE          INV  120
COMMON /SCRAT/ NS,NS2,A(13500)                  INV 130
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG           INV 140
COMMON /BLOCRC/ NRC,RC(12600)                  INV 150
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NINV 160
IARGS,VWXYZ(8),NERROR                         INV 170
DIMENSION ARGS(100)                           INV 180
EQUIVALENCE (ARGS(1),RC(12501))             INV 190
COMMON /BLOCKC/ KIO,INUNIT,ISCRAT,KBDOUT,KRDKNT,LLIST  INV 200
DIMENSION ERR(3)                             INV 210
IF(NARGS.EQ.6) GO TO 10                      INV 220
CALL ERROR (10)                            INV 230
RETURN                                     INV 240
10 J=NARGS                                     INV 250
CALL CKIND (J)                               INV 260
IF (J.NE.0) GO TO 100                        INV 270
IF(IARGS(3)-IARGS(4)) 105,20,107            INV 275
20 NARGS=8                                     INV 320
KIND(7)=0                                    INV 330
KIND(8)=0                                    INV 340
IF (L2.EQ.1) GO TO 30                        INV 350
IARGS(9)=IARGS(6)                           INV 360
NARGS=9                                     INV 370
KIND(9)=0                                    INV 380
CALL ADRESS (NARGS,JE)                      INV 390
IF (JE.LE.0) GO TO 150                        INV 400
IARGS(6)=IARGS(5)                           INV 410
IARGS(5)=1                                    INV 420
IARGS(8)=1                                    INV 430
GO TO 40                                     INV 440
30 IARGS(8)=IARGS(3)                          INV 450
40 IARGS(7)=IARGS(3)                          INV 460
J=2                                         INV 470
CALL MTXCHK (J)                            INV 480
IF (J.NE.0) GO TO 140                        INV 490
JA=IARGS(1)                                 INV 500
JB=IARGS(5)                                 INV 510
IF (2*((IARGS(3)+2)**2).GT.NS) GO TO 120    INV 520
IF (NERROR.NE.0) RETURN                      INV 530
M1=IARGS(3)+1                               INV 540
IF (L2.EQ.2) M1=M1+1                         INV 550
CALL INVCHK (RC(JA),NROW,IARGS(3),A,M1,RC(JB),L2,ERR,IND)  INV 560
C CHECK TO SEE IF MATRIX WAS INVERTED. YES, IF IND=0    INV 570
IF (IND.NE.0) GO TO 130                      INV 580
IA=IARGS(3)                                 INV 590
IF (L2.EQ.2) GO TO 70                        INV 600
STORE INVERTED MATRIX                      INV 610
DO 60 I=1,IA                                INV 620
JC=JB                                      INV 630
JD=(I-1)*M1+1                              INV 640

```

```

DO 50 J=1,IA                         INV 650
RC(JC)=A(JD)                         INV 660
JC=JC+1                               INV 670
50   JD=JD+1                           INV 680
60   JB=JB+NROW                         INV 690
GO TO 90                               INV 700
C   STORE RESULTS OF SOLUTION          INV 710
70   JC=M1*IARGS(3)+1                  INV 720
DO 80 I=1,IA                           INV 730
RC(JE)=A(JC)                          INV 740
JC=JC+1                               INV 750
80   JE=JE+1                           INV 760
C   DETERMINE SMALLEST ERROR BOUND    INV 770
90   SERR=AMIN1(ERR(1),ERR(2),ERR(3))  INV 780
WRITE (ISCRAT,160) SERR               INV 790
RETURN                                 INV 800
100  CALL ERROR (3)                   INV 810
RETURN                                 INV 820
105  IARGS(4)=IARGS(3)                INV 830
GO TO 110                             INV 840
107  IARGS(3)=IARGS(4)                INV 850
110  CALL ERROR (210)                INV 860
C   PRINT ROW AND COLUMNS DO NOT AGREE,SIZE OF COLUMNS IS SET TO ROW  INV 870
GO TO 20                               INV 880
120  CALL ERROR (23)                 INV 890
C   PRINT MATRIX TOO LARGE TO INVERT  INV 900
RETURN                                 INV 910
130  CALL ERROR (22)                 INV 920
C   PRINT MATRIX IS SINGULAR OR NEAR SINGULAR-NO INVERSE      INV 930
RETURN                                 INV 940
140  IF (J.EQ.1) GO TO 150           INV 950
CALL ERROR (17)                         INV 960
RETURN                                 INV 970
150  CALL ERROR (11)                 INV 980
RETURN                                 INV 990
C
160  FORMAT (6X,20(1H+),43H SMALLEST ERROR BOUND ON INVERTED MATRIX IS,INV1010
1E8.1,7H      +++)                  INV1020
END                                     INV1030

```

```

SUBROUTINE ITERAT                               ITE 10
C      VERSION 5.00     ITERAT      5/15/70      ITE 20
COMMON /BLOCRC/ NRC,RC(12600)                  ITE 30
COMMON /SCRAT/ NS,NS2,A(13500)                 ITE 40
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NITE 50
1ARGS,VWXYZ(8),NERROR                         ITE 60
DIMENSION ARGS(100)                           ITE 70
EQUIVALENCE (ARGS(1),RC(12501))              ITE 80
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG          ITE 90
C      L2=1       ITERATE X IN ++, Y IN ++, DESIRED Y IN ++ START STORIN ITE 100
C      L2=2       ISETUP X IN ++, Y IN ++, DESIRED Y IN ++ STORE STARTIN ITE 110
C                      COLUMN ++
C                      IN COLUMN ++
C      L2=3       ISOLATE X IN ++ FOR Y IN COL ++=** STORE IN ++ AND ++ ITE 140
C      ISOLATE X IN ++ FOR Y IN COL ++=** USE ,, POINTS, STORE ITE 150
C                      IN ++ AND ++
C      ITERATE AND ISETUP USE THREE COLUMNS AFTER THE INDICATED STORE ITE 170
C      STORAGE FOR ITERATE AND ISETUP ARE NEW X ,AVERAGE BRACKETING X,ITE 180
C      AVERAGE BRACKETTING Y, AND SUCCESSFUL Y           ITE 190
C      CODE BY MRS. CARLA G. MESSINA NSRDS-NBS OCT 1967 ITE 200
INSERT=3                                         ITE 210
IF (NARGS-4) 10,40,80                          ITE 220
10 K=10                                         ITE 230
20 CALL ERROR (K)                            ITE 240
30 RETURN                                       ITE 250
40 IF (L2-2) 50,50,10                          ITE 260
50 NARGS=7                                       ITE 270
DO 60 I=5,7                                     ITE 280
KIND(I)=0                                       ITE 290
60 IARGS(I)=IARGS(I-1)+1                      ITE 300
IF (NROW-5) 70,180,180                         ITE 310
70 K=17                                         ITE 320
GO TO 20                                       ITE 330
80 IF (L2-3) 10,90,10                          ITE 340
90 IARGS(7)=IARGS(5)                           ITE 350
KIND(7)=KIND(5)                                ITE 360
IF (NARGS-6) 100,110,10                         ITE 370
100 NARGS=6                                      ITE 380
IARGS(6)=IARGS(5)                             ITE 390
KIND(6)=KIND(5)                                ITE 400
IARGS(5)=IARGS(4)                             ITE 410
KIND(5)=KIND(4)                                ITE 420
GO TO 140                                       ITE 430
110 IF (KIND(4)) 120,130,120                  ITE 440
120 K=3                                         ITE 450
GO TO 20                                       ITE 460
130 INSERT=IARGS(4)                            ITE 470
IARGS(4)=IARGS(5)                             ITE 480
140 IF (INSERT-1) 120,150,150                  ITE 490
150 IF (NROW-INSERT-2) 70,160,160             ITE 500
160 IF (KIND(3)) 170,120,170                  ITE 510
170 POINT=ARGS(3)                            ITE 520
IARGS(3)=IARGS(2)                             ITE 530
KIND(3)=0                                     ITE 540
180 CALL CHKCOL (J)                           ITE 550
IF (J) 120,190,120                         ITE 560
190 IF (NERROR.NE.0) GO TO 30                  ITE 570
IF (NRMAX-1) 200,210,220                     ITE 580
200 K=9                                         ITE 590

```

```

      GO TO 20
210   K=215
      GO TO 20
220   IF (L2-2) 300,230,230
230   I1=IARGS(1)-1
      DO 240 I=2,NRMAX
      K=I1+I
      IF (RC(K-1)-RC(K)) 250,240,270
240   CONTINUE
      GO TO 290
250   DO 260 I=2,NRMAX
      K=I1+I
      IF (RC(K-1)-RC(K)) 260,260,290
260   CONTINUE
      GO TO 300
270   DO 280 I=2,NRMAX
      K=I1+I
      IF (RC(K-1)-RC(K)) 290,280,280
280   CONTINUE
      GO TO 300
290   CALL ERROR (216)
300   DO 310 I=1,NRMAX
      I1=IARGS(1)+I-1
      I2=IARGS(2)+I-1
      I3=IARGS(3)+I-1
      A(I)=RC(I1)
      M=I+NRMAX
      A(M)=RC(I2)
      M=M+NRMAX
310   A(M)=RC(I3)
      M=0
      M1=0
      IOVFL=0
      IND2=INSERT+2
      DIV=INSERT+1
      IDIV=INSERT+1
      I1=IARGS(4)-1
      I2=IARGS(5)-1
      I3=IARGS(6)-1
      I4=IARGS(7)-1
      I5=IARGS(4)-1
      IF (L2-2) 500,500,320
C           ISOLATE
320   K1=NRMAX+1
      L1=2*NRMAX
      I5=IARGS(5)-1
      IF (POINT-A(K1)) 340,330,340
330   M=M+1
      I2=I2+1
      RC(I2)=A(1)
      M1=M1+1
      I3=I3+1
      RC(I3)=A(1)
340   K1=K1+1
      I=1
      DO 440 K=K1,L1
      I=I+1
      IF (POINT-A(K-1)) 350,440,360
350   IF (POINT-A(K)) 440,390,370
      ITE 600
      ITE 610
      ITE 620
      ITE 630
      ITE 640
      ITE 650
      ITE 660
      ITE 670
      ITE 680
      ITE 690
      ITE 700
      ITE 710
      ITE 720
      ITE 730
      ITE 740
      ITE 750
      ITE 760
      ITE 770
      ITE 780
      ITE 790
      ITE 800
      ITE 810
      ITE 820
      ITE 830
      ITE 840
      ITE 850
      ITE 860
      ITE 870
      ITE 880
      ITE 890
      ITE 900
      ITE 910
      ITE 920
      ITE 930
      ITE 940
      ITE 950
      ITE 960
      ITE 970
      ITE 980
      ITE 990
      ITE1000
      ITE1010
      ITE1020
      ITE1030
      ITE1040
      ITE1050
      ITE1060
      ITE1070
      ITE1080
      ITE1090
      ITE1100
      ITE1110
      ITE1120
      ITE1130
      ITE1140
      ITE1150
      ITE1160
      ITE1170
      ITE1180

```

```

360 IF (POINT-A(K)) 370,390,440 ITE1190
370 IF (NROW-M-IND2) 380,400,400 ITE1200
380 IOVFL=1 ITE1210
GO TO 430 ITE1220
390 A(I-1)=A(I) ITE1230
IF (NROW-M-1) 430,420,420 ITE1240
400 M=M+1 ITE1250
I2=I2+1 ITE1260
RC(I2)=A(I-1) ITE1270
DELT=(A(I)-A(I-1))/DIV ITE1280
DO 410 II=1,INSERT ITE1290
M=M+1 ITE1300
I2=I2+1 ITE1310
410 RC(I2)=RC(I2-1)+DELT ITE1320
420 M=M+1 ITE1330
I2=I2+1 ITE1340
RC(I2)=A(I) ITE1350
430 M1=M1+1 ITE1360
I3=I3+1 ITE1370
RC(I3)=(A(I-1)+A(I))/2.0 ITE1380
IF (NROW-M1) 470,470,440 ITE1390
440 CONTINUE ITE1400
IF (M) 480,480,450 ITE1410
450 NRMAX=M ITE1420
460 IF (IOVFL) 470,30,470 ITE1430
470 K=218 ITE1440
GO TO 20 ITE1450
480 K=217 ITE1460
DO 490 I=1,NRMAX ITE1470
I5=I5+1 ITE1480
490 RC(I5)=A(I) ITE1490
GO TO 20 ITE1500
500 K1=2*NRMAX+1 ITE1510
L1=3*NRMAX ITE1520
DO 510 K=1,NRMAX ITE1530
IF (A(L1)) 520,510,520 ITE1540
510 L1=L1-1 ITE1550
L1=3*NRMAX ITE1560
520 NEWY=L1-K1+1 ITE1570
IF (L2-2) 780,530,10 ITE1580
C ISETUP ITE1590
530 DO 740 K=K1,L1 ITE1600
I=1 ITE1610
L=NRMAX+1 ITE1620
IF (A(K)-A(I)) 590,540,590 ITE1630
540 IF (NROW-M-IND2) 550,560,560 ITE1640
550 IOVFL=1 ITE1650
GO TO 580 ITE1660
560 DO 570 II=1,IND2 ITE1670
M=M+1 ITE1680
I1=I1+1 ITE1690
570 RC(I1)=A(I) ITE1700
580 M1=M1+1 ITE1710
I2=I2+1 ITE1720
RC(I2)=A(I) ITE1730
I3=I3+1 ITE1740
RC(I3)=A(L) ITE1750
I4=I4+1 ITE1760
RC(I4)=A(K) ITE1770

```

```

      IF (NROW-M1) 470,470,590          ITE1780
590   DO 730 I=2,NRMAX                ITE1790
         L=NRMAX+I                      ITE1800
         IF (A(L-1)-A(K)) 600,730,610    ITE1810
600   IF (A(L)-A(K)) 730,620,670      ITE1820
610   IF (A(L)-A(K)) 670,620,730      ITE1830
620   IF (NROW-M-IND2) 630,640,640      ITE1840
630   IOVFL=1                          ITE1850
      GO TO 660                        ITE1860
640   DO 650 J=1,IND2                  ITE1870
      M=M+1                            ITE1880
      I1=I1+1                          ITE1890
650   RC(I1)=A(I)                      ITE1900
660   M1=M1+1                          ITE1910
      I2=I2+1                          ITE1920
      RC(I2)=A(I)                      ITE1930
      I3=I3+1                          ITE1940
      RC(I3)=A(L)                      ITE1950
      GO TO 720                        ITE1960
670   IF (NROW-M-IND2) 680,690,690      ITE1970
680   IOVFL=1                          ITE1980
      GO TO 710                        ITE1990
690   DELT=(A(I)-A(I-1))/DIV          ITE2000
      M=M+1                            ITE2010
      I1=I1+1                          ITE2020
      RC(I1)=A(I-1)                    ITE2030
      DO 700 J=1,INSERT                ITE2040
      M=M+1                            ITE2050
      I1=I1+1                          ITE2060
700   RC(I1)=RC(I1-1)+DELT          ITE2070
      M=M+1                            ITE2080
      I1=I1+1                          ITE2090
      RC(I1)=A(I)                      ITE2100
710   M1=M1+1                          ITE2110
      I2=I2+1                          ITE2120
      RC(I2)=(A(I)+A(I-1))/2.0        ITE2130
      I3=I3+1                          ITE2140
      RC(I3)=(A(L)+A(L-1))/2.0        ITE2150
720   I4=I4+1                          ITE2160
      RC(I4)=A(K)                      ITE2170
      IF (NROW-M1) 470,470,740        ITE2180
730   CONTINUE                         ITE2190
740   CONTINUE                         ITE2200
750   IF (M) 480,480,760              ITE2210
760   IF (M-NEWY) 770,770,450        ITE2220
770   NRMAX=NEWY                      ITE2230
      GO TO 460                        ITE2240
C                                     ITERATE
780   II=IND2*(NRMAX/IND2)            ITE2250
      IF (II) 210,210,790            ITE2260
790   DO 960 K=K1,L1                  ITE2270
      DO 950 K3=1,II,IND2            ITE2280
      DO 830 J=1,1DIV                ITE2290
      I=K3+J                          ITE2300
      L=NRMAX+I                      ITE2310
      IF (A(L-1)-A(K)) 810,800,820    ITE2320
800   I=I-1                            ITE2330
      L=L-1                            ITE2340
      GO TO 840                        ITE2350
                                     ITE2360

```

810	IF (A(L)-A(K)) 830,840,890	ITE2370
820	IF (A(L)-A(K)) 890,840,830	ITE2380
830	CONTINUE	ITE2390
	GO TO 950	ITE2400
840	IF (NROW-M-IND2) 850,860,860	ITE2410
850	I0VFL=1	ITE2420
	GO TO 880	ITE2430
860	DO 870 J=1,IND2	ITE2440
	M=M+1	ITE2450
	I1=I1+1	ITE2460
870	RC(I1)=A(I)	ITE2470
880	M1=M1+1	ITE2480
	I2=I2+1	ITE2490
	RC(I2)=A(I)	ITE2500
	I3=I3+1	ITE2510
	RC(I3)=A(L)	ITE2520
	GO TO 940	ITE2530
890	IF (NROW-M-IND2) 900,910,910	ITE2540
900	I0VFL=1	ITE2550
	GO TO 930	ITE2560
910	DELT=(A(I)-A(I-1))/DIV	ITE2570
	M=M+1	ITE2580
	I1=I1+1	ITE2590
	RC(I1)=A(I-1)	ITE2600
	DO 920 J=1,INSERT	ITE2610
	M=M+1	ITE2620
	I1=I1+1	ITE2630
920	RC(I1)=RC(I1-1)+DELT	ITE2640
	M=M+1	ITE2650
	I1=I1+1	ITE2660
	RC(I1)=A(I)	ITE2670
930	M1=M1+1	ITE2680
	I2=I2+1	ITE2690
	RC(I2)=(A(I)+A(I-1))/2.0	ITE2700
	I3=I3+1	ITE2710
	RC(I3)=(A(L)+A(L-1))/2.0	ITE2720
940	I4=I4+1	ITE2730
	RC(I4)=A(K)	ITE2740
	IF (NROW-M1) 470,470,960	ITE2750
950	CONTINUE	ITE2760
960	CONTINUE	ITE2770
	GO TO 750	ITE2780
	END	ITE2790

C	BLOCK DATA	LBCONS	LBC	10
C	VERSION 5.00	LBCONS	LBC	20
C	BLOCK DATA		LBC	30
C	THESE CONSTANTS MAY HAVE TO BE CHANGED FOR OTHER COMPUTERS OR		LBC	40
C	LIBRARY ROUTINES		LBC	50
C	DSNCOS IS USED BY DOUBLE PRECISION SIN AND COS FUNCTIONS IN ORDER		LBC	60
C	TO TRAP IF ARGUMENT BECOMES TOO LARGE		LBC	70
C	XTRIG IS USED BY FSIN,FCOS FUNCTIONS IN ORDER TO TRAP IF ARGUMENT	LBC	80	
C	BECOMES TOO LARGE	LBC	90	
C	XEXP IS USED BY FEXP FUNCION IN ORDER TO TRAP IF ARGUMENT	LBC	100	
C	BECOMES TOO LARGE	LBC	110	
C	DXEXP IS USED BY FDEXP FUNCTION IN ORDER TO TRAP IF ARGUMENT	LBC	120	
C	BECOMES TOO LARGE	LBC	130	
C	ER IS USED BY SUBROUTINE CSPINV TO CHECK ON A COMPUTER ZERO	LBC	140	
C	NBC IS USED BY SUBROUTINE ERRINT AND IS THE NUMBER OF BINARY	LBC	150	
C	BITS IN THE CHARACTERISTIC OF A DOUBLE PRECISION NUMBER	LBC	160	
C	NBM IS USED BY SUBROUTINE ERRINT AND IS THE NUMBER OF BINARY	LBC	170	
C	BITS IN THE MANTISSA OF A DOUBLE PRECISION NUMBER	LBC	180	
C	TRRTPI IS USED BY SUBROUTINE ERRINT AND IS THE VALUE 2.0/SQRT(PI)	LBC	190	
C	ISIGD IS USED BY SUBROUTINE FIXFLO AND INFERR, NO. OF SIGNIFICANT	LBC	210	
C	DIGITS I/O SUB. CAN PRINT AFTER DECIMAL POINT ISIGD=8	LBC	220	
C	DOUBLE PRECISION DSNCOS,DXEXP,TRRTPI	LBC	230	
C	COMMON/CONSLB/XTRIG,XEXP	LBC	240	
C	COMMON/CONLB2/ER,ISIGD	LBC	250	
C	COMMON/CONSTS/PI,E,HALFPI,DEG,RAD,XALOG	LBC	260	
C	COMMON/DCONLB/DSNCOS,DXEXP	LBC	270	
C	COMMON/DCONL2/TRRTPI,NBC,NBM	LBC	280	
C	DATA DSNCOS/3.5D16/,DXEXP/704.0D0/	LBC	290	
C	DATA NBC/11/,NBM/60/,TRRTPI/1.128379167095512574D0/	LBC	300	
C	DATA XTRIG/3.3E7/,XEXP/88.0 /	LBC	310	
C	DATA ER/1.E-8/,ISIGD/8/	LBC	320	
C	THIS BLOCK DEFINES CONSTANTS TO BE USED THROUGHOUT OMNITAB	LBC	330	
C	WHOSE VALUE (ACCURACY) WILL HAVE TO BE CHANGED FOR OTHER	LBC	340	
C	COMPUTERS	LBC	350	
C	PI=3.14159265 (VALUE OF PI)	LBC	360	
C	E, 2.7182818 (BASE OF NATURAL LOGS)	LBC	370	
C	HALFPI=1.5707963 (VALUE OF PI/2)	LBC	380	
C	DEG= 57.2957795 (NUMBER OF DEGREES IN ONE RADIAN)	LBC	390	
C	RAD=.0174532925 (NUMBER OF RADIANS IN ONE DEGREE)	LBC	400	
C	XALOG= 38. (EXPONENT BOUND)	LBC	410	
C	DATA PI,E,HALFPI,DEG,RAD,XALOG/	LBC	420	
C	1 3.14159265,2.7182818,1.5707963,57.2957795,.0174532925,38./	LBC	430	
C	END	LBC	440	
C		LBC	450	
C		LBC	460	
C		LBC	470	
C		LBC	480	
C		LBC	490	
C		LBC	500	
C		LBC	510	
C		LBC	520	
C		LBC	530	
C		LBC	540	
C		LBC	550	
C		LBC	560	

```

SUBROUTINE LIST (K) LIS 10
C VERSION 5.00 LIST 5/15/70 LIS 20
C WRITTEN BY R VARNER 3/14/68 LIS 30
C K=0 COMMAND IS LIST LIS 40
C K=1 COMMAND IS NOLIST LIS 50
COMMON /BLOCKC/ KIO,INUNIT,ISCRAT,KBDOUT,KRDKNT,LLIST LIS 60
COMMON /BLOCRC/ NRC,RC(12600) LIS 70
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NLIS 80
IARGS,VWXYZ(8),NERROR LIS 90
DIMENSION ARGS(100) LIS 100
EQUIVALENCE (ARGS(1),RC(12501)) LIS 110
IF (K.EQ.0) GO TO 20 LIS 120
C NO LIST OR NOLIST LIS 130
IARGS(1)=0 LIS 140
10 IF (NERROR.EQ.0) LLIST=IARGS(1) LIS 150
WRITE (ISCRAT,30) IARGS(1)
RETURN LIS 170
C LIS 180
C LIST (WITH NO ARGUMENTS) = LIST 3 LIS 190
C LIST 0 = NO LISTING LIS 200
C LIST 1 = LIST ONLY INFORMATIVE DIAGS. LIS 210
C LIST 2 = LIST ONLY ARITH. ERR LIS 220
C LIST 3 = LIST BOTH TYPES OF ERRORS LIS 230
C LIST=4 SUPPRESS BOTH ARITHMETIC ERRORS AND INFORMATIVE DIAGNOSTICS LIS 235
C IF A FATAL ERROR OCCURS, LLIST IS SET TO AND KEPT AT 3 LIS 240
20 IF(NARGS.EQ.0.OR.IARGS(1).LT.0.OR.IARGS(1).GT.4) IARGS(1)=3 LIS 250
GO TO 10 LIS 260
C LIS 270
30 FORMAT (1H,,I1,82X) LIS 280
END LIS 290

```

```

FUNCTION LOCATE (L) LOC 10
C VERSION 5.00 LOCATE 5/15/70 LOC 20
COMMON /BLOCKB/ NSTMT,NSTMTH,NCOM,LCOM,IOVFL,COM(2000) LOC 30
C LOC 40
C THIS FUNCTION SEARCHES THE LIST OF STORED COMMANDS TO SEE IF ONE LOC 50
C WITH STATEMENT NUMBER L EXISTS. IF IT DOES, RETURN ITS LOCATION. LOC 60
C IF IT DOESN'T EXIST, RETURN NEGATIVE THE LOCATION OF THE NEXT LOC 70
C HIGHER STATEMENT NUMBER. LOC 80
C LOC 90
I=1 LOC 100
AL=L LOC 105
10 IF (COM(I)-AL) 20,30,40 LOC 110
20 I=I+IFIX(COM(I+1))
GO TO 10 LOC 130
30 LOCATE=I LOC 140
GO TO 50 LOC 150
40 LOCATE=-I LOC 160
50 RETURN LOC 170
END LOC 180

```

C BLOCK DATA LOOKTB LOT 10  
 C VERSION 5.00 LOOKTB 5/15/70 LOT 20  
 BLOCK DATA LOT 30  
 COMMON/ICODE/ NIR,NID,NIRD,LIR,LID,LIRD LOT 40  
 COMMON/CODE/IALPH (6),NALPH (5), ID(9,3),  
 1 IR(300,4),IRD(30,6) LOT 50  
 C \*\*\*\* THE FOLLOWING CARDS ARE NEEDED ONLY FOR TAPE OPERATIONS LOT 60  
 COMMON/ICODTP/NITP, LITP LOT 70  
 COMMON/CODETP/ITP(10,4) LOT 80  
 COMMON/CODETP/ITP(10,4) LOT 90  
 \*\*\*\*\*LOT 100  
 C ADD SUB SUBTRA MULT MULTIP DIV DIVIDE RAISE LOT 110  
 C  
 DATA IR(1,1),IR(1,2),IR(1,3),IR(1,4),IR(2,1),IR(2,2),IR(2,3),  
 1 IR(2,4),IR(3,1),IR(3,2),IR(3,3),IR(3,4),IR(4,1),IR(4,2),IR(4,3),  
 2 IR(4,4),IR(5,1),IR(5,2),IR(5,3),IR(5,4),IR(6,1),IR(6,2),IR(6,3),  
 3 IR(6,4),IR(7,1),IR(7,2),IR(7,3),IR(7,4),IR(8,1),IR(8,2),IR(8,3),  
 4 IR(8,4) /  
 5 841,0,11,1,14420, 0,11,2,14420,15067,11,2, LOT 130  
 6 10056,14580,11,3,10056,14839,11,3,3181, 0,11,4,  
 7 3181,6674,11,4,13158,13986,11,5/ LOT 140  
 C  
 DATA IR( 9,1),IR( 9,2),IR( 9,3),IR( 9,4),IR(10,1),IR(10,2),  
 1 IR(10,3),IR(10,4),IR(11,1),IR(11,2),IR(11,3),IR(11,4),IR(12,1),  
 2 IR(12,2),IR(12,3),IR(12,4),IR(13,1),IR(13,2),IR(13,3),IR(13,4),  
 3 IR(14,1),IR(14,2),IR(14,3),IR(14,4),IR(15,1),IR(15,2),IR(15,3),  
 4 IR(15,4),IR(16,1),IR(16,2),IR(16,3),IR(16,4),IR(17,1),IR(17,2),  
 5 IR(17,3),IR(17,4),IR(18,1),IR(18,2),IR(18,3),IR(18,4),IR(19,1),  
 6 IR(19,2),IR(19,3),IR(19,4),IR(20,1),IR(20,2),IR(20,3),IR(20,4),  
 7 IR(21,1),IR(21,2),IR(21,3),IR(21,4),IR(22,1),IR(22,2),IR(22,3),  
 8 IR(22,4),IR(23,1),IR(23,2),IR(23,3),IR(23,4),IR(24,1),IR(24,2),  
 9 IR(24,3),IR(24,4),IR(25,1),IR(25,2),IR(25,3),IR(25,4) /  
 A 14108, 0,12,1, 2611, 0,12,2,14621, 0,12, 3, LOT 290  
 B 2612, 0,12, 4, 0, 0, 0, 1251,10206,12, 5, LOT 300  
 C 0, 0, 0, 0, 825,13851,12, 6, 0, 0, 0, 0, 0, LOT 310  
 D 1270,10206,12, 7, 0, 0, 0, 825,14580,12, 8, LOT 320  
 E 14108, 2916,12, 9, 2611, 2916,12,10,14621, 2916,12,11, LOT 330  
 F 2612, 2916,12,12, 1251,10314,12,13/ LOT 340  
 DATA IR(26,1),IR(26,2),IR(26,3),IR(26,4),IR(27,1),IR(27,2),  
 1 IR(27,3),IR(27,4),IR(28,1),IR(28,2),IR(28,3),IR(28,4),IR(29,1),  
 2 IR(29,2),IR(29,3),IR(29,4),IR(30,1),IR(30,2),IR(30,3),IR(30,4),  
 3 IR(31,1),IR(31,2),IR(31,3),IR(31,4),IR(32,1),IR(32,2),IR(32,3),  
 4 IR(32,4),IR(33,1),IR(33,2),IR(33,3),IR(33,4),IR(34,1),IR(34,2),  
 5 IR(34,3),IR(34,4),IR(35,1),IR(35,2),IR(35,3),IR(35,4),IR(36,1),  
 6 IR(36,2),IR(36,3),IR(36,4),IR(37,1),IR(37,2),IR(37,3),IR(37,4),  
 7 IR(38,1),IR(38,2),IR(38,3),IR(38,4),IR(39,1),IR(39,2),IR(39,3),  
 8 IR(39,4),IR(40,1),IR(40,2),IR(40,3),IR(40,4),IR(41,1),IR(41,2),  
 9 IR(41,3),IR(41,4),IR(42,1),IR(42,2),IR(42,3),IR(42,4),IR(43,1),  
 A IR(43,2),IR(43,3),IR(43,4),IR(44,1),IR(44,2),IR(44,3),IR(44,4),  
 B IR(45,1),IR(45,2),IR(45,3),IR(45,4)/  
 C 825,13959,12,14, 1270,10314,12,15, 825,14688,12,16, LOT 450  
 D 14328,14580,12,17, 14328,14580,12,17, LOT 460  
 E 4309, 0,12,18, 4309,11318,12,18,10348, 4309,12,19, LOT 470  
 C  
 DATA IR(46,1),IR(46,2),IR(46,3),IR(46,4),IR(47,1),IR(47,2),  
 1 IR(47,3),IR(47,4),IR(48,1),IR(48,2),IR(48,3),IR(48,4),IR(49,1),  
 2 IR(49,2),IR(49,3),IR(49,4),IR(50,1),IR(50,2),IR(50,3),IR(50,4),  
 3 IR(51,1),IR(51,2),IR(51,3),IR(51,4),IR(52,1),IR(52,2),IR(52,3),  
 4 IR(52,4),IR(53,1),IR(53,2),IR(53,3),IR(53,4),IR(54,1),IR(54,2),  
 5 IR(54,3),IR(54,4),IR(55,1),IR(55,2),IR(55,3),IR(55,4),IR(56,1),  
 6 IR(56,2),IR(56,3),IR(56,4),IR(57,1),IR(57,2),IR(57,3),IR(57,4),  
 7 IR(58,1),IR(58,2),IR(58,3),IR(58,4),IR(59,1),IR(59,2),IR(59,3),  
 8 IR(59,4),IR(60,1),IR(60,2),IR(60,3),IR(60,4),IR(61,1),IR(61,2),  
 9 IR(61,3),IR(61,4),IR(62,1),IR(62,2),IR(62,3),IR(62,4),IR(63,1),  
 A IR(63,2),IR(63,3),IR(63,4),IR(64,1),IR(64,2),IR(64,3),IR(64,4),  
 B IR(65,1),IR(65,2),IR(65,3),IR(65,4),IR(66,1),IR(66,2),IR(66,3),  
 C IR(66,4),IR(67,1),IR(67,2),IR(67,3),IR(67,4),IR(68,1),IR(68,2),  
 D IR(68,3),IR(68,4),IR(69,1),IR(69,2),IR(69,3),IR(69,4),IR(70,1),  
 E IR(70,2),IR(70,3),IR(70,4),IR(71,1),IR(71,2),IR(71,3),IR(71,4),  
 F IR(72,1),IR(72,2),IR(72,3),IR(72,4),IR(73,1),IR(73,2),IR(73,3),  
 G IR(73,4),IR(74,1),IR(74,2),IR(74,3),IR(74,4),IR(75,1),IR(75,2),  
 H IR(75,3),IR(75,4),IR(76,1),IR(76,2),IR(76,3),IR(76,4),IR(77,1),  
 I IR(77,2),IR(77,3),IR(77,4),IR(78,1),IR(78,2),IR(78,3),IR(78,4),  
 J IR(79,1),IR(79,2),IR(79,3),IR(79,4),IR(80,1),IR(80,2),IR(80,3),  
 K IR(80,4),IR(81,1),IR(81,2),IR(81,3),IR(81,4),IR(82,1),IR(82,2),  
 L IR(82,3),IR(82,4),IR(83,1),IR(83,2),IR(83,3),IR(83,4),IR(84,1),  
 M IR(84,2),IR(84,3),IR(84,4),IR(85,1),IR(85,2),IR(85,3),IR(85,4),  
 N IR(86,1),IR(86,2),IR(86,3),IR(86,4),IR(87,1),IR(87,2),IR(87,3),  
 O IR(87,4),IR(88,1),IR(88,2),IR(88,3),IR(88,4),IR(89,1),IR(89,2),  
 P IR(89,3),IR(89,4),IR(90,1),IR(90,2),IR(90,3),IR(90,4),IR(91,1),  
 Q IR(91,2),IR(91,3),IR(91,4),IR(92,1),IR(92,2),IR(92,3),IR(92,4),  
 R IR(93,1),IR(93,2),IR(93,3),IR(93,4),IR(94,1),IR(94,2),IR(94,3),  
 S IR(94,4),IR(95,1),IR(95,2),IR(95,3),IR(95,4),IR(96,1),IR(96,2),  
 T IR(96,3),IR(96,4),IR(97,1),IR(97,2),IR(97,3),IR(97,4),IR(98,1),  
 U IR(98,2),IR(98,3),IR(98,4),IR(99,1),IR(99,2),IR(99,3),IR(99,4),  
 V IR(100,1),IR(100,2),IR(100,3),IR(100,4),IR(101,1),IR(101,2),IR(101,3),  
 W IR(101,4),IR(102,1),IR(102,2),IR(102,3),IR(102,4),IR(103,1),IR(103,2),  
 X IR(103,3),IR(103,4),IR(104,1),IR(104,2),IR(104,3),IR(104,4),IR(105,1),  
 Y IR(105,2),IR(105,3),IR(105,4),IR(106,1),IR(106,2),IR(106,3),IR(106,4),  
 Z IR(107,1),IR(107,2),IR(107,3),IR(107,4),IR(108,1),IR(108,2),IR(108,3),  
 AA IR(108,4),IR(109,1),IR(109,2),IR(109,3),IR(109,4),IR(110,1),IR(110,2),  
 BB IR(110,3),IR(110,4),IR(111,1),IR(111,2),IR(111,3),IR(111,4),IR(112,1),  
 CC IR(112,2),IR(112,3),IR(112,4),IR(113,1),IR(113,2),IR(113,3),IR(113,4),  
 DD IR(114,1),IR(114,2),IR(114,3),IR(114,4),IR(115,1),IR(115,2),IR(115,3),  
 EE IR(115,4),IR(116,1),IR(116,2),IR(116,3),IR(116,4),IR(117,1),IR(117,2),  
 FF IR(117,3),IR(117,4),IR(118,1),IR(118,2),IR(118,3),IR(118,4),IR(119,1),  
 GG IR(119,2),IR(119,3),IR(119,4),IR(120,1),IR(120,2),IR(120,3),IR(120,4),  
 HH IR(121,1),IR(121,2),IR(121,3),IR(121,4),IR(122,1),IR(122,2),IR(122,3),  
 II IR(122,4),IR(123,1),IR(123,2),IR(123,3),IR(123,4),IR(124,1),IR(124,2),  
 JJ IR(124,3),IR(124,4),IR(125,1),IR(125,2),IR(125,3),IR(125,4),IR(126,1),  
 KK IR(126,2),IR(126,3),IR(126,4),IR(127,1),IR(127,2),IR(127,3),IR(127,4),  
 LL IR(128,1),IR(128,2),IR(128,3),IR(128,4),IR(129,1),IR(129,2),IR(129,3),  
 MM IR(129,4),IR(130,1),IR(130,2),IR(130,3),IR(130,4),IR(131,1),IR(131,2),  
 NN IR(131,3),IR(131,4),IR(132,1),IR(132,2),IR(132,3),IR(132,4),IR(133,1),  
 OO IR(133,2),IR(133,3),IR(133,4),IR(134,1),IR(134,2),IR(134,3),IR(134,4),  
 PP IR(135,1),IR(135,2),IR(135,3),IR(135,4),IR(136,1),IR(136,2),IR(136,3),  
 QQ IR(136,4),IR(137,1),IR(137,2),IR(137,3),IR(137,4),IR(138,1),IR(138,2),  
 RR IR(138,3),IR(138,4),IR(139,1),IR(139,2),IR(139,3),IR(139,4),IR(140,1),  
 SS IR(140,2),IR(140,3),IR(140,4),IR(141,1),IR(141,2),IR(141,3),IR(141,4),  
 TT IR(142,1),IR(142,2),IR(142,3),IR(142,4),IR(143,1),IR(143,2),IR(143,3),  
 UU IR(143,4),IR(144,1),IR(144,2),IR(144,3),IR(144,4),IR(145,1),IR(145,2),  
 VV IR(145,3),IR(145,4),IR(146,1),IR(146,2),IR(146,3),IR(146,4),IR(147,1),  
 WW IR(147,2),IR(147,3),IR(147,4),IR(148,1),IR(148,2),IR(148,3),IR(148,4),  
 XX IR(149,1),IR(149,2),IR(149,3),IR(149,4),IR(150,1),IR(150,2),IR(150,3),  
 YY IR(150,4),IR(151,1),IR(151,2),IR(151,3),IR(151,4),IR(152,1),IR(152,2),  
 ZZ IR(152,3),IR(152,4),IR(153,1),IR(153,2),IR(153,3),IR(153,4),IR(154,1),  
 AA IR(154,2),IR(154,3),IR(154,4),IR(155,1),IR(155,2),IR(155,3),IR(155,4),  
 BB IR(156,1),IR(156,2),IR(156,3),IR(156,4),IR(157,1),IR(157,2),IR(157,3),  
 CC IR(157,4),IR(158,1),IR(158,2),IR(158,3),IR(158,4),IR(159,1),IR(159,2),  
 DD IR(159,3),IR(159,4),IR(160,1),IR(160,2),IR(160,3),IR(160,4),IR(161,1),  
 EE IR(161,2),IR(161,3),IR(161,4),IR(162,1),IR(162,2),IR(162,3),IR(162,4),  
 FF IR(163,1),IR(163,2),IR(163,3),IR(163,4),IR(164,1),IR(164,2),IR(164,3),  
 GG IR(164,4),IR(165,1),IR(165,2),IR(165,3),IR(165,4),IR(166,1),IR(166,2),  
 HH IR(166,3),IR(166,4),IR(167,1),IR(167,2),IR(167,3),IR(167,4),IR(168,1),  
 II IR(168,2),IR(168,3),IR(168,4),IR(169,1),IR(169,2),IR(169,3),IR(169,4),  
 JJ IR(170,1),IR(170,2),IR(170,3),IR(170,4),IR(171,1),IR(171,2),IR(171,3),  
 KK IR(171,4),IR(172,1),IR(172,2),IR(172,3),IR(172,4),IR(173,1),IR(173,2),  
 LL IR(173,3),IR(173,4),IR(174,1),IR(174,2),IR(174,3),IR(174,4),IR(175,1),  
 MM IR(175,2),IR(175,3),IR(175,4),IR(176,1),IR(176,2),IR(176,3),IR(176,4),  
 PP IR(177,1),IR(177,2),IR(177,3),IR(177,4),IR(178,1),IR(178,2),IR(178,3),  
 QQ IR(178,4),IR(179,1),IR(179,2),IR(179,3),IR(179,4),IR(180,1),IR(180,2),  
 RR IR(180,3),IR(180,4),IR(181,1),IR(181,2),IR(181,3),IR(181,4),IR(182,1),  
 SS IR(182,2),IR(182,3),IR(182,4),IR(183,1),IR(183,2),IR(183,3),IR(183,4),  
 TT IR(184,1),IR(184,2),IR(184,3),IR(184,4),IR(185,1),IR(185,2),IR(185,3),  
 UU IR(185,4),IR(186,1),IR(186,2),IR(186,3),IR(186,4),IR(187,1),IR(187,2),  
 VV IR(187,3),IR(187,4),IR(188,1),IR(188,2),IR(188,3),IR(188,4),IR(189,1),  
 WW IR(189,2),IR(189,3),IR(189,4),IR(190,1),IR(190,2),IR(190,3),IR(190,4),  
 XX IR(191,1),IR(191,2),IR(191,3),IR(191,4),IR(192,1),IR(192,2),IR(192,3),  
 YY IR(192,4),IR(193,1),IR(193,2),IR(193,3),IR(193,4),IR(194,1),IR(194,2),  
 ZZ IR(194,3),IR(194,4),IR(195,1),IR(195,2),IR(195,3),IR(195,4),IR(196,1),  
 AA IR(196,2),IR(196,3),IR(196,4),IR(197,1),IR(197,2),IR(197,3),IR(197,4),  
 BB IR(198,1),IR(198,2),IR(198,3),IR(198,4),IR(199,1),IR(199,2),IR(199,3),  
 CC IR(199,4),IR(200,1),IR(200,2),IR(200,3),IR(200,4),IR(201,1),IR(201,2),  
 DD IR(201,3),IR(201,4),IR(202,1),IR(202,2),IR(202,3),IR(202,4),IR(203,1),  
 EE IR(203,2),IR(203,3),IR(203,4),IR(204,1),IR(204,2),IR(204,3),IR(204,4),  
 FF IR(205,1),IR(205,2),IR(205,3),IR(205,4),IR(206,1),IR(206,2),IR(206,3),  
 GG IR(206,4),IR(207,1),IR(207,2),IR(207,3),IR(207,4),IR(208,1),IR(208,2),  
 HH IR(208,3),IR(208,4),IR(209,1),IR(209,2),IR(209,3),IR(209,4),IR(210,1),  
 II IR(210,2),IR(210,3),IR(210,4),IR(211,1),IR(211,2),IR(211,3),IR(211,4),  
 JJ IR(212,1),IR(212,2),IR(212,3),IR(212,4),IR(213,1),IR(213,2),IR(213,3),  
 KK IR(213,4),IR(214,1),IR(214,2),IR(214,3),IR(214,4),IR(215,1),IR(215,2),  
 LL IR(215,3),IR(215,4),IR(216,1),IR(216,2),IR(216,3),IR(216,4),IR(217,1),  
 MM IR(217,2),IR(217,3),IR(217,4),IR(218,1),IR(218,2),IR(218,3),IR(218,4),  
 PP IR(219,1),IR(219,2),IR(219,3),IR(219,4),IR(220,1),IR(220,2),IR(220,3),  
 QQ IR(220,4),IR(221,1),IR(221,2),IR(221,3),IR(221,4),IR(222,1),IR(222,2),  
 RR IR(222,3),IR(222,4),IR(223,1),IR(223,2),IR(223,3),IR(223,4),IR(224,1),  
 SS IR(224,2),IR(224,3),IR(224,4),IR(225,1),IR(225,2),IR(225,3),IR(225,4),  
 TT IR(226,1),IR(226,2),IR(226,3),IR(226,4),IR(227,1),IR(227,2),IR(227,3),  
 UU IR(227,4),IR(228,1),IR(228,2),IR(228,3),IR(228,4),IR(229,1),IR(229,2),  
 VV IR(229,3),IR(229,4),IR(230,1),IR(230,2),IR(230,3),IR(230,4),IR(231,1),  
 WW IR(231,2),IR(231,3),IR(231,4),IR(232,1),IR(232,2),IR(232,3),IR(232,4),  
 XX IR(233,1),IR(233,2),IR(233,3),IR(233,4),IR(234,1),IR(234,2),IR(234,3),  
 YY IR(234,4),IR(235,1),IR(235,2),IR(235,3),IR(235,4),IR(236,1),IR(236,2),  
 ZZ IR(236,3),IR(236,4),IR(237,1),IR(237,2),IR(237,3),IR(237,4),IR(238,1),  
 AA IR(238,2),IR(238,3),IR(238,4),IR(239,1),IR(239,2),IR(239,3),IR(239,4),  
 BB IR(240,1),IR(240,2),IR(240,3),IR(240,4),IR(241,1),IR(241,2),IR(241,3),  
 CC IR(241,4),IR(242,1),IR(242,2),IR(242,3),IR(242,4),IR(243,1),IR(243,2),  
 DD IR(243,3),IR(243,4),IR(244,1),IR(244,2),IR(244,3),IR(244,4),IR(245,1),  
 EE IR(245,2),IR(245,3),IR(245,4),IR(246,1),IR(246,2),IR(246,3),IR(246,4),  
 FF IR(247,1),IR(247,2),IR(247,3),IR(247,4),IR(248,1),IR(248,2),IR(248,3),  
 GG IR(248,4),IR(249,1),IR(249,2),IR(249,3),IR(249,4),IR(250,1),IR(250,2),  
 HH IR(250,3),IR(250,4),IR(251,1),IR(251,2),IR(251,3),IR(251,4),IR(252,1),  
 II IR(252,2),IR(252,3),IR(252,4),IR(253,1),IR(253,2),IR(253,3),IR(253,4),  
 JJ IR(254,1),IR(254,2),IR(254,3),IR(254,4),IR(255,1),IR(255,2),IR(255,3),  
 KK IR(255,4),IR(256,1),IR(256,2),IR(256,3),IR(256,4),IR(257,1),IR(257,2),  
 LL IR(257,3),IR(257,4),IR(258,1),IR(258,2),IR(258,3),IR(258,4),IR(259,1),  
 MM IR(259,2),IR(259,3),IR(259,4),IR(260,1),IR(260,2),IR(260,3),IR(260,4),  
 PP IR(261,1),IR(261,2),IR(261,3),IR(261,4),IR(262,1),IR(262,2),IR(262,3),  
 QQ IR(262,4),IR(263,1),IR(263,2),IR(263,3),IR(263,4),IR(264,1),IR(264,2),  
 RR IR(264,3),IR(264,4),IR(265,1),IR(265,2),IR(265,3),IR(265,4),IR(266,1),  
 SS IR(266,2),IR(266,3),IR(266,4),IR(267,1),IR(267,2),IR(267,3),IR(267,4),  
 TT IR(268,1),IR(268,2),IR(268,3),IR(268,4),IR(269,1),IR(269,2),IR(269,3),  
 UU IR(269,4),IR(270,1),IR(270,2),IR(270,3),IR(270,4),IR(271,1),IR(271,2),  
 VV IR(271,3),IR(271,4),IR(272,1),IR(272,2),IR(272,3),IR(272,4),IR(273,1),  
 WW IR(273,2),IR(273,3),IR(273,4),IR(274,1),IR(274,2),IR(274,3),IR(274,4),  
 XX IR(275,1),IR(275,2),IR(275,3),IR(275,4),IR(276,1),IR(276,2),IR(276,3),  
 YY IR(276,4),IR(277,1),IR(277,2),IR(277,3),IR(277,4),IR(278,1),IR(278,2),  
 ZZ IR(278,3),IR(278,4),IR(279,1),IR(279,2),IR(279,3),IR(279,4),IR(280,1),  
 AA IR(280,2),IR(280,3),IR(280,4),IR(281,1),IR(281,2),IR(281,3),IR(281,4),  
 BB IR(282,1),IR(282,2),IR(282,3),IR(282,4),IR(283,1),IR(283,2),IR(283,3),  
 CC IR(283,4),IR(284,1),IR(284,2),IR(284,3),IR(284,4),IR(285,1),IR(285,2),  
 DD IR(285,3),IR(285,4),IR(286,1),IR(286,2),IR(286,3),IR(286,4),IR(287,1),  
 EE IR(287,2),IR(287,3),IR(287,4),IR(288,1),IR(288,2),IR(288,3),IR(288,4),  
 FF IR(289,1),IR(289,2),IR(289,3),IR(289,4),IR(290,1),IR(290,2),IR(290,3),  
 GG IR(290,4),IR(291,1),IR(291,2),IR(291,3),IR(291,4),IR(292,1),IR(292,2),  
 HH IR(292,3),IR(292,4),IR(293,1),IR(293,2),IR(293,3),IR(293,4),IR(294,1),  
 II IR(294,2),IR(294,3),IR(294,4),IR(295,1),IR(295,2),IR(295,3),IR(295,4),  
 JJ IR(296,1),IR(296,2),IR(296,3),IR(296,4),IR(297,1),IR(297,2),IR(297,3),  
 KK IR(297,4),IR(298,1),IR(298,2),IR(298,3),IR(298,4),IR(299,1),IR(299,2),  
 LL IR(299,3),IR(299,4),IR(300,1),IR(300,2),IR(300,3),IR(300,4),IR(301,1),  
 MM IR(301,2),IR(301,3),IR(301,4),IR(302,1),IR(302,2),IR(302,3),IR(302,4),  
 PP IR(303,1),IR(303,2),IR(303,3),IR(303,4),IR(304,1),IR(304,2),IR(304,3),  
 QQ IR(304,4),IR(305,1),IR(305,2),IR(305,3),IR(305,4),IR(306,1),IR(306,2),  
 RR IR(306,3),IR(306,4),IR(307,1),IR(307,2),IR(307,3),IR(307,4),IR(308,1),  
 SS IR(308,2),IR(308,3),IR(308,4),IR(309,1),IR(309,2),IR(309,3),IR(309,4),  
 TT IR(310,1),IR(310,2),IR(310,3),IR(310,4),IR(311,1),IR(311,2),IR(311,3),  
 UU IR(311,4),IR(312,1),IR(312,2),IR(312,3),IR(312,4),IR(313,1),IR(313,2),  
 VV IR(313,3),IR(313,4),IR(314,1),IR(314,2),IR(314,3),IR(314,4),IR(315,1),  
 WW IR(315,2),IR(315,3),IR(315,4),IR(316,1),IR(316,2),IR(316,3),IR(3



```

C          DATA      IR(84,1),IR(84,2),IR(84,3),IR(84,4),IR(85,1),IR(85,2),    LOT1190
C          1 IR(85,3),IR(85,4)/    LOT1200
C          2 9849, 9288,17, 1, 9964, 7079,17, 2/    LOT1210
C          MADD     MSUB     MTRANS ATRANS AADD     ASUB     AMULT   AMULTI ASUBTR    LOT1220
C          MSCALA   ADIVID  ADIV    ARAISE    LOT1230
C          DATA      IR(86,1),IR(86,2),IR(86,3),IR(86,4),IR(87,1),IR(87,2),    LOT1240
C          6           IR(87,3),IR(87,4),IR(88,1),IR(88,2),IR(88,3),IR(88,4),    LOT1250
C          7 IR(89,1),IR(89,2),IR(89,3),IR(89,4),IR(90,1),IR(90,2),IR(90,3),    LOT1260
C          8 IR(90,4),IR(91,1),IR(91,2),IR(91,3),IR(91,4),IR(92,1),IR(92,2),    LOT1270
C          9 IR(92,3),IR(92,4),IR(93,1),IR(93,2),IR(93,3),IR(93,4),IR(94,1),    LOT1280
C          A IR(94,2),IR(94,3),IR(94,4),IR(95,1),IR(95,2),IR(95,3),IR(95,4),    LOT1290
C          B IR(96,1),IR(96,2),IR(96,3),IR(96,4),IR(97,1),IR(97,2),IR(97,3),    LOT1300
C          C IR(97,4),IR(98,1),IR(98,2),IR(98,3),IR(98,4)/    LOT1310
C          D 9508, 2916,18, 1,10011, 1458,18, 2,10035, 1126,18, 3,    LOT1320
C          E 1287, 1126,18, 3, 760, 2916,18, 4, 1263, 1458,18, 5,    LOT1330
C          F 1101, 9288,18, 6, 1101, 9297,18, 6, 1263, 2016,18, 5,    LOT1340
C          G 9993, 1054,18, 6, 846,16285,18, 7, 846,16038,18, 7,    LOT1350
C          H 1216, 7079,18, 8/    LOT1360
C          DATA      IR( 99,1),IR( 99,2),IR( 99,3),IR( 99,4),IR(100,1),    LOT1370
C          1 IR(100,2),IR(100,3),IR(100,4),IR(101,1),IR(101,2),IR(101,3),    LOT1380
C          2 IR(101,4),IR(102,1),IR(102,2),IR(102,3),IR(102,4),IR(103,1),    LOT1390
C          3 IR(103,2),IR(103,3),IR(103,4),IR(104,1),IR(104,2),IR(104,3),    LOT1400
C          4 IR(104,4)/    LOT1410
C          5     0,     0, 0, 0,     0,     0, 0, 0,     0,     0, 0, 0,    LOT1420
C          6     0,     0, 0, 0,     0,     0, 0, 0,     0,     0, 0, 0/    LOT1430
C          PARSUM PARPRO RMS      AVERAG SUM    LOT1440
C          DATA      IR(105,1),IR(105,2),IR(105,3),IR(105,4),IR(106,1),    LOT1450
C          1 IR(106,2),IR(106,3),IR(106,4),IR(107,1),IR(107,2),IR(107,3),    LOT1460
C          2 IR(107,4),IR(108,1),IR(108,2),IR(108,3),IR(108,4),IR(109,1),    LOT1470
C          3 IR(109,2),IR(109,3),IR(109,4)/    LOT1480
C          4 11709,14431,20, 1,11709,12165,20, 2,13492,     0,20, 3,    LOT1490
C          5 1328,13156,20, 4,14431,     0,20, 5/    LOT1500
C          ROWSUM PRODUC DEFINE MAX      MAXIMU MIN      MINIMU SORT      ORDER    LOT1510
C          ERASE EXCHAN FLIP      CHANGE HIERAR LIST      NULL    LOT1520
C          DATA      IR(110,1),IR(110,2),IR(110,3),IR(110,4),IR(111,1),    LOT1530
C          1 IR(111,2),IR(111,3),IR(111,4),IR(112,1),IR(112,2),IR(112,3),    LOT1540
C          2 IR(112,4),IR(113,1),IR(113,2),IR(113,3),IR(113,4),IR(114,1),    LOT1550
C          3 IR(114,2),IR(114,3),IR(114,4),IR(115,1),IR(115,2),IR(115,3),    LOT1560
C          4 IR(115,4),IR(116,1),IR(116,2),IR(116,3),IR(116,4),IR(117,1),    LOT1570
C          5 IR(117,2),IR(117,3),IR(117,4),IR(118,1),IR(118,2),IR(118,3),    LOT1580
C          6 IR(118,4),IR(119,1),IR(119,2),IR(119,3),IR(119,4),IR(120,1),    LOT1590
C          7 IR(120,2),IR(120,3),IR(120,4),IR(121,1),IR(121,2),IR(121,3),    LOT1600
C          8 IR(121,4),IR(122,1),IR(122,2),IR(122,3),IR(122,4),IR(123,1),    LOT1610
C          9 IR(123,2),IR(123,3),IR(123,4),IR(124,1),IR(124,2),IR(124,3),    LOT1620
C          A IR(124,4),IR(125,1),IR(125,2),IR(125,3),IR(125,4),IR(126,1),    LOT1630
C          B IR(126,2),IR(126,3),IR(126,4)/    LOT1640
C          C 13550,14431,21, 1,12165, 3486,21, 2, 3057, 6944,21, 3,    LOT1650
C          D 9528,     0,21, 5, 9528, 6933,21, 5, 9734,     0,21, 6,    LOT1660
C          E 9734, 6933,21, 6,14274,14580,21, 8,11425, 4131,21, 9,    LOT1670

```

F 4132,13986,21,10, 4296, 5873,21,11, 4707,11664,21,12, LOT1780  
 G 2404,10400,21,13, 6080,13167,21,14, 9010,14580,21,15, LOT1790  
 H 0, 0, 0, 0,10785, 8748,21,17/ LOT1800  
 C  
 C POLYFI SPOLYF FIT SFIT SOLVE MORTHO LOT1810  
 C  
 DATA IR(127,1),IR(127,2),IR(127,3),IR(127,4),IR(128,1), LOT1840  
 1 IR(128,2),IR(128,3),IR(128,4),IR(129,1),IR(129,2),IR(129,3), LOT1850  
 2 IR(129,4),IR(130,1),IR(130,2),IR(130,3),IR(130,4),IR(131,1), LOT1860  
 3 IR(131,2),IR(131,3),IR(131,4),IR(132,1),IR(132,2),IR(132,3), LOT1870  
 4 IR(132,4)/ LOT1880  
 5 12081,18396,22, 1,14298, 9429,22, 2, 4637, 0,22, 3, LOT1890  
 6 14022,14580,22, 4,14268,16173,16, 2, 9900,14811,22, 5/ LOT1900  
 C  
 C COUNT SHORTE EXPAND DUPLIC MOVE DIM AMOVE LOT1910  
 C MMOVE PROMOT DEMOTE DIMENS LOT1920  
 C  
 DATA IR(133,1),IR(133,2),IR(133,3),IR(133,4),IR(134,1), LOT1930  
 1 IR(134,2),IR(134,3),IR(134,4),IR(135,1),IR(135,2),IR(135,3), LOT1940  
 2 IR(135,4),IR(136,1),IR(136,2),IR(136,3),IR(136,4),IR(137,1), LOT1950  
 3 IR(137,2),IR(137,3),IR(137,4),IR(138,1),IR(138,2),IR(138,3), LOT1960  
 4 IR(138,4),IR(139,1),IR(139,2),IR(139,3),IR(139,4),IR(140,1), LOT1970  
 5 IR(140,2),IR(140,3),IR(140,4),IR(141,1),IR(141,2),IR(141,3), LOT1980  
 6 IR(141,4),IR(142,1),IR(142,2),IR(142,3),IR(142,4),IR(143,1), LOT1990  
 7 IR(143,2),IR(143,3),IR(143,4),IR(144,1),IR(144,2),IR(144,3), LOT2000  
 8 IR(144,4),IR(145,1),IR(145,2),IR(145,3),IR(145,4)/ LOT2010  
 9 0, 0, 0, 0, 0, 0, 2613,10746,23, 2, LOT2020  
 A 14082,13667,23, 3, 4309, 1111,23, 4, 3499, 8994,23, 5, LOT2030  
 B 9904, 3645,23, 6, 3172, 0,23,12, 1095,16173,23, 6, LOT2040  
 C 9843,16173,23, 6,12165, 9902,23,10, 3064,11480,23,11, LOT2050  
 D 3172, 4042,23,12/ LOT2060  
 C  
 C STATIS SSTATI RANKS ACCURA LOT2070  
 C  
 DATA IR(146,1),IR(146,2),IR(146,3),IR(146,4),IR(147,1), LOT2080  
 1 IR(147,2),IR(147,3),IR(147,4),IR(148,1),IR(148,2),IR(148,3), LOT2090  
 2 IR(148,4),IR(149,1),IR(149,2),IR(149,3),IR(149,4),IR(150,1), LOT2100  
 3 IR(150,2),IR(150,3),IR(150,4)/ LOT2110  
 4 14392,14842,24, 1,14384, 1278,24, 2,13163, 8532,24, 3, LOT2120  
 5 813,15796,11, 6, 0, 0, 0, 0/ LOT2130  
 C  
 C SELECT SEARCH CENSOR LOT2140  
 C  
 DATA IR(151,1),IR(151,2),IR(151,3),IR(151,4),IR(152,1), LOT2150  
 1 IR(152,2),IR(152,3),IR(152,4),IR(153,1),IR(153,2),IR(153,3), LOT2160  
 2 IR(153,4)/ LOT2170  
 3 13998, 3746,25, 1,13987,13211,25, 2, 2336,14274,25, 3/ LOT2180  
 C  
 C MVECDI MVECMA MMATV LOT2190  
 C  
 DATA IR(154,1),IR(154,2),IR(154,3),IR(154,4),IR(155,1), LOT2200  
 1 IR(155,2),IR(155,3),IR(155,4),IR(156,1),IR(156,2),IR(156,3), LOT2210  
 2 IR(156,4),IR(157,1),IR(157,2),IR(157,3),IR(157,4),IR(158,1), LOT2220  
 3 IR(158,2),IR(158,3),IR(158,4),IR(159,1),IR(159,2),IR(159,3), LOT2230  
 4 IR(159,4)/ LOT2240  
 5 10076, 2304,26, 1, 0, 0, 0, 0,10076, 2539,26, 2, LOT2250  
 6 0, 0, 0, 0, 9829,15179,26, 3, 0, 0, 0, 0/ LOT2260  
 C  
 C MKRONE MTRIAN MEIGEN LOT2270

C DATA IR(160,1),IR(160,2),IR(160,3),IR(160,4),IR(161,1),  
 1 IR(161,2),IR(161,3),IR(161,4),IR(162,1),IR(162,2),IR(162,3),  
 2 IR(162,4)/  
 3 9792,11318,17, 3,10035, 6602,17, 4, 9621, 5252,17, 5/ LOT2370  
 C INTERP LOT2380  
 C  
 C DATA IR(163,1),IR(163,2),IR(163,3),IR(163,4)/  
 1 6959, 4147,25, 4/ LOT2390  
 C MPROPE APROPE SMPROP SAPROP  
 C DATA IR(164,1),IR(164,2),IR(164,3),IR(164,4),IR(165,1),  
 1 IR(165,2),IR(165,3),IR(165,4),IR(166,1),IR(166,2),IR(166,3),  
 2 IR(166,4),IR(167,1),IR(167,2),IR(167,3),IR(167,4)/  
 3 9927,11372,27, 1, 1179,11372,27, 2,  
 4 14218,13543,27, 3,13894,13543,27, 4/ LOT2400  
 C ITERATE ISETUP ISOLATE  
 C  
 C DATA IR(168,1),IR(168,2),IR(168,3),IR(168,4),IR(169,1),  
 1 IR(169,2),IR(169,3),IR(169,4),IR(170,1),IR(170,2),IR(170,3),  
 2 IR(170,4)/  
 3 7106,13169,28, 1, 7079,15163,28, 2, 7089, 8795,28, 3/ LOT2410  
 C EXTREMA SEPARATE INSERT MAXMIN  
 C  
 C DATA IR(171,1),IR(171,2),IR(171,3),IR(171,4),IR(172,1),  
 1 IR(172,2),IR(172,3),IR(172,4),IR(173,1),IR(173,2),IR(173,3),  
 2 IR(173,4),IR(174,1),IR(174,2),IR(174,3),IR(174,4)/  
 3 4313,13270,29, 4,14002, 1216,29, 2, 6958, 4151,29, 3,  
 4 9528, 9734,29, 4/ LOT2420  
 C LAGUER NORMLA HERMIT UCHEBY TCHEBY LEGEND  
 C  
 C DATA IR(175,1),IR(175,2),IR(175,3),IR(175,4),IR(176,1),  
 1 IR(176,2),IR(176,3),IR(176,4),IR(177,1),IR(177,2),IR(177,3),  
 2 IR(177,4),IR(178,1),IR(178,2),IR(178,3),IR(178,4),IR(179,1),  
 3 IR(179,2),IR(179,3),IR(179,4),IR(180,1),IR(180,2),IR(180,3),  
 4 IR(180,4)/  
 5 8782,15462,19, 2,10629, 9802,19, 1, 5985, 9740,19, 3,  
 6 15398, 3724,19, 4,14669, 3724,19, 6, 8890, 4027,19, 5/ LOT2430  
 C BJZERO BJONE BYZERO BYONE BIZERO BIONE BKZERO BKONE  
 C EXIZER EXIONE EXKZER EXKONE KBIZER KBIONE KBKZER KBKONE  
 C  
 C DATA IR(181,1),IR(181,2),IR(181,3),IR(181,4),IR(182,1),  
 1 IR(182,2),IR(182,3),IR(182,4),IR(183,1),IR(183,2),IR(183,3),  
 2 IR(183,4),IR(184,1),IR(184,2),IR(184,3),IR(184,4),IR(185,1),  
 3 IR(185,2),IR(185,3),IR(185,4),IR(186,1),IR(186,2),IR(186,3),  
 4 IR(186,4),IR(187,1),IR(187,2),IR(187,3),IR(187,4),IR(188,1),  
 5 IR(188,2),IR(188,3),IR(188,4),IR(189,1),IR(189,2),IR(189,3),  
 6 IR(189,4),IR(190,1),IR(190,2),IR(190,3),IR(190,4),IR(191,1),  
 7 IR(191,2),IR(191,3),IR(191,4),IR(192,1),IR(192,2),IR(192,3),  
 8 IR(192,4),IR(193,1),IR(193,2),IR(193,3),IR(193,4),IR(194,1),  
 9 IR(194,2),IR(194,3),IR(194,4),IR(195,1),IR(195,2),IR(195,3),  
 A IR(195,4),IR(196,1),IR(196,2),IR(196,3),IR(196,4)/  
 B 1754, 4146,30, 1, 1743,10341,30, 2, 2159, 4146,30, 3,  
 C 2148,10341,30, 4, 1727, 4146,30, 5, 1716,10341,30, 6, LOT2440

D	1781, 4146,30, 7, 1770,10341,30, 8, 4302,19107,30, 9,	LOT2960
E	4302,11318,30,10, 4304,19107,30,11, 4304,11318,30,12,	LOT2970
F	8082,19107,30,13, 8082,11318,30,14, 8084,19107,30,15,	LOT2980
G	8084,11318,30,16/	LOT2990
C		LOT3000
C	KEXIZE KEXION KEXKZE KEXKON CIZERO CIONE CKZERO CKONE	LOT3010
C	CEIZER CEIONE CEKZER CEKONE INTJO BESJN HARMON BESIN	LOT3020
C	BESKN	LOT3030
C		LOT3040
	DATA IR(197,1),IR(197,2),IR(197,3),IR(197,4),IR(198,1),	LOT3050
1	IR(198,2),IR(198,3),IR(198,4),IR(199,1),IR(199,2),IR(199,3),	LOT3060
2	IR(199,4),IR(200,1),IR(200,2),IR(200,3),IR(200,4),IR(201,1),	LOT3070
3	IR(201,2),IR(201,3),IR(201,4),IR(202,1),IR(202,2),IR(202,3),	LOT3080
4	IR(202,4),IR(203,1),IR(203,2),IR(203,3),IR(203,4),IR(204,1),	LOT3090
5	IR(204,2),IR(204,3),IR(204,4),IR(205,1),IR(205,2),IR(205,3),	LOT3100
6	IR(205,4),IR(206,1),IR(206,2),IR(206,3),IR(206,4),IR(207,1),	LOT3110
7	IR(207,2),IR(207,3),IR(207,4),IR(208,1),IR(208,2),IR(208,3),	LOT3120
8	IR(208,4),IR(209,1),IR(209,2),IR(209,3),IR(209,4),IR(210,1),	LOT3130
9	IR(210,2),IR(210,3),IR(210,4),IR(211,1),IR(211,2),IR(211,3),	LOT3140
A	IR(211,4),IR(212,1),IR(212,2),IR(212,3),IR(212,4),IR(213,1),	LOT3150
B	IR(213,2),IR(213,3),IR(213,4)/	LOT3160
C	8178, 7268,30,17, 8178, 6980,30,18, 8178, 8726,30,19,	LOT3170
D	8178, 8438,30,20, 2456, 4146,30,21, 2445,10341,30,22,	LOT3180
E	2510, 4146,30,23, 2499,10341,30,24, 2331,19107,30,25,	LOT3190
F	2331,11318,30,26, 2333,19107,30,27, 2333,11318,30,28,	LOT3200
G	6959, 7695,30,29, 1612, 7668,30,32, 5877, 9896,30,37,	LOT3210
H	1612, 6939,30,38, 1612, 8397,30,39/	LOT3220
C		LOT3230
C	TWOWAY	LOT3240
C		LOT3250
	DATA IR(214,1),IR(214,2),IR(214,3),IR(214,4)/	LOT3260
1	15216,16819,24,6/	LOT3270
C		LOT3280
C	FLEXIB	LOT3290
C		LOT3300
	DATA IR(215,1),IR(215,2),IR(215,3),IR(215,4)/	LOT3310
1	4703,17741,13,12/	LOT3320
C		LOT3330
C	SQUARE	LOT3340
C		LOT3350
	DATA IR(216,1),IR(216,2),IR(216,3),IR(216,4)/	LOT3360
1	14331, 1220,12,34/	LOT3370
C		LOT3380
C	ACOALE AAVERA	LOT3390
C		LOT3400
	DATA IR(217,1),IR(217,2),IR(217,3),IR(217,4),IR(218,1),	LOT3410
1	IR(218,2),IR(218,3),IR(218,4)/	LOT3420
2	825, 1058,18, 9, 778, 4132,18,10/	LOT3430
C		LOT3440
C	MATCH	LOT3450
C		LOT3460
	DATA IR(219,1),IR(219,2),IR(219,3),IR(219,4)/	LOT3470
1	9524,2403,25,5/	LOT3480
C		LOT3490
C	HISTOG NHISTO FREQUE	LOT3500
C		LOT3510
	DATA IR(220,1),IR(220,2),IR(220,3),IR(220,4),IR(221,1),	LOT3520
1	IR(221,2),IR(221,3),IR(221,4),IR(222,1),IR(222,2),IR(222,3),	LOT3530
2	IR(222,4)/	LOT3540

C 3 6094,14992,24, 8,10431,14406,24, 9, 4865,12965,24,10/ LOT3550  
 C CORREL SCORRE LOT3560  
 C DATA IR(223,1),IR(223,2),IR(223,3),IR(223,4),IR(224,1), LOT3570  
 1 IR(224,2),IR(224,3),IR(224,4)/ LOT3580  
 2 2610,13269,24,11,13947,13613,24,12/ LOT3590  
 C COMPARE,ONEWAY,SONEWAY,ERROR,CERF,STWOWAY LOT3600  
 C DATA IR(225,1),IR(225,2),IR(225,3),IR(225,4),IR(226,1),IR(226,2), LOT3610  
 1 IR(226,3),IR(226,4),IR(227,1),IR(227,2),IR(227,3),IR(227,4), LOT3620  
 2 IR(228,1),IR(228,2),IR(228,3),IR(228,4),IR(229,1),IR(229,2), LOT3630  
 3 IR(229,3),IR(229,4),IR(230,1),IR(230,2),IR(230,3),IR(230,4)/ LOT3640  
 4 2605,11709,14,15,11318,16819,24,13,14270, 4267,24,14, LOT3650  
 5 4149,11421,21,18, 2340, 4374,21,19,14414,11557,24, 7/ LOT3660  
 C CTOF,FTOC,ATOMIC,MOLWT,EINSTEIN,PFTRANS,PFATOMIC,PARTFUNCT,BOLDISTLOT3720  
 DATA IR(231,1),IR(231,2),IR(231,3),IR(231,4),IR(232,1),IR(232,2),LOT3730  
 1 IR(232,3),IR(232,4),IR(233,1),IR(233,2),IR(233,3),IR(233,4),LOT3740  
 2 IR(234,1),IR(234,2),IR(234,3),IR(234,4),IR(235,1),IR(235,2),LOT3750  
 3 IR(235,3),IR(235,4),IR(236,1),IR(236,2),IR(236,3),IR(236,4),LOT3760  
 4 IR(237,1),IR(237,2),IR(237,3),IR(237,4),IR(238,1),IR(238,2),LOT3770  
 5 IR(238,3),IR(238,4),IR(239,1),IR(239,2),IR(239,3),IR(239,4)/LOT3780  
 6 2742, 4374,31, 1, 4929, 2187,31, 2, 1284, 9723,31, 3, 9894,17307LOT3790  
 7 ,31, 4, 3902,14396,31,5,11846,13163,31, 6,11827,14998,31, 7,11709LOT3800  
 8 ,14763,31, 8, 1875, 3178,31, 9/ LOT3810  
 C ROUND LOT3820  
 C DATA IR(240,1),IR(240,2),IR(240,3),IR(240,4)/13548,10314,13,14/ LOT3830  
 C COMPLEX ARITHMETIC - CADD, CSUBTRACT, CMULTIPLY, CDIVIDE, LOT3840  
 C CRECTANGULAR, CPOLAR LOT3850  
 C DATA IR(241,1),IR(241,2),IR(241,3),IR(241,4),IR(242,1),IR(242,2),LOT3900  
 1 IR(242,3),IR(242,4),IR(243,1),IR(243,2),IR(243,3),IR(243,4),LOT3910  
 2 IR(244,1),IR(244,2),IR(244,3),IR(244,4),IR(245,1),IR(245,2),LOT3920  
 3 IR(245,3),IR(245,4),IR(246,1),IR(246,2),IR(246,3),IR(246,4)/LOT3930  
 4 2218, 2916,32, 1, 2721, 2016,32, 2, 2559, 9297,32, 3, LOT3940  
 5 2304,16285,32, 4, 2678, 2728,32, 5, 2634, 8793,32, 6/ LOT3950  
 C \*\*\*\*\*LOT3960  
 C \* \*LOT3970  
 C \* USED THUR IR(246,4):AVAILABLE IR(247,1) THRU IR(300,4) \*LOT3990  
 C \* \*LOT4000  
 C \*\*\*\*\*LOT4010  
 C RESET PRINT PUNCH READ ABRIDG APRINT MPRINT LOT4020  
 C DATA ID(1,1),ID(1,2),ID(1,3),ID(2,1),ID(2,2),ID(2,3),ID(3,1), LOT4030  
 1 ID(3,2),ID(3,3),ID(4,1),ID(4,2),ID(4,3),ID(5,1),ID(5,2),ID(5,3), LOT4040  
 2 ID(6,1),ID(6,2),ID(6,3),ID(7,1),ID(7,2),ID(7,3)/ LOT4050  
 3 13276, 4185,1,12159,10746,2,12245, 2403,3,13258, 2916,5, LOT4060  
 4 801, 6676,6, 1179, 6959,4, 9927, 6959,7/ LOT4070  
 C NPRINT LOT4080  
 C DATA ID(8,1),ID(8,2),ID(8,3)/ LOT4090  
 C \*\*\*\*\*LOT4100  
 C \* \*LOT4110  
 C \* \*LOT4120  
 C \*\*\*\*\*LOT4130

1 10656, 6959,8/ LOT4140  
 C  
 C A B C D E F LOT4150  
 C  
 C DATA IALPH(1),IALPH(2),IALPH(3),IALPH(4),IALPH(5),IALPH(6)/ LOT4160  
 1 729, 1458, 2187, 2916, 3645, 4374/ LOT4170  
 C  
 C V W X Y Z LOT4180  
 C  
 C DATA NALPH(1),NALPH(2),NALPH(3),NALPH(4),NALPH(5)/ LOT4190  
 1 16038,16767,17496,18225,18954/ LOT4200  
 C  
 C NO LIST CLOSE UP NEW PAGE LOT4210  
 C M(XX,) M(X,AX)=M(X,X) M(XAX,) M(AD) M(DA) M(AV) M(V,A) LOT4220  
 C  
 C DATA IRD( 1,1),IRD( 1,2),IRD( 1,3),IRD( 1,4),IRD( 1,5), LOT4230  
 1 IRD( 1,6),IRD( 2,1),IRD( 2,2),IRD( 2,3),IRD( 2,4),IRD( 2,5), LOT4240  
 2 IRD( 2,6),IRD( 3,1),IRD( 3,2),IRD( 3,3),IRD( 3,4),IRD( 3,5), LOT4250  
 3 IRD( 3,6),IRD( 4,1),IRD( 4,2),IRD( 4,3),IRD( 4,4),IRD( 4,5), LOT4260  
 4 IRD( 4,6),IRD( 5,1),IRD( 5,2),IRD( 5,3),IRD( 5,4),IRD( 5,5), LOT4270  
 5 IRD( 5,6),IRD( 6,1),IRD( 6,2),IRD( 6,3),IRD( 6,4),IRD( 6,5), LOT4280  
 6 IRD( 6,6),IRD( 7,1),IRD( 7,2),IRD( 7,3),IRD( 7,4),IRD( 7,5), LOT4290  
 7 IRD( 7,6),IRD( 8,1),IRD( 8,2),IRD( 8,3),IRD( 8,4),IRD( 8,5), LOT4300  
 8 IRD( 8,6),IRD( 9,1),IRD( 9,2),IRD( 9,3),IRD( 9,4),IRD( 9,5), LOT4310  
 9 IRD( 9,6),IRD(10,1),IRD(10,2),IRD(10,3),IRD(10,4),IRD(10,5), LOT4320  
 A IRD(10,6)/ LOT4330  
 B 10611, 0, 9010,14580,21,16, 2526,13986,15741, 0,23, 1, LOT4340  
 C 10364, 0,11698, 3645,13, 8, 9477, 0,18144, 0,51, 1, LOT4350  
 D 9477, 0,17496, 0,51, 2, 9477, 0,17547, 0,51, 3, LOT4360  
 E 9477, 0, 837, 0,52, 1, 9477, 0,2943, 0,52, 2, LOT4370  
 F 9477, 0, 1323, 0,53, 1, 9477, 0,16038, 0,53, 2/ LOT4380  
 C  
 C DUMMY A B C D LOT4390  
 C  
 C DATA IRD(11,1),IRD(11,2),IRD(11,3),IRD(11,4),IRD(11,5), LOT4400  
 1 IRD(11,6),IRD(12,1),IRD(12,2),IRD(12,3),IRD(12,4),IRD(12,5), LOT4410  
 3 IRD(12,6),IRD(13,1),IRD(13,2),IRD(13,3),IRD(13,4),IRD(13,5), LOT4420  
 3 IRD(13,6),IRD(14,1),IRD(14,2),IRD(14,3),IRD(14,4),IRD(14,5), LOT4430  
 4 IRD(14,6),IRD(15,1),IRD(15,2),IRD(15,3),IRD(15,4),IRD(15,5), LOT4440  
 5 IRD(15,6)/ LOT4450  
 6 0, 0, 0, 0, 0, 3496,10152, 729, 0,54, 2, LOT4460  
 7 3496,10152, 1458, 0,54, 3, 3496,10152,2187, 0,54, 4, LOT4470  
 8 3496,10152, 2916, 0,54, 5/ LOT4480  
 C  
 C ROW SUM F PROBAB LOT4490  
 C  
 C DATA IRD(16,1),IRD(16,2),IRD(16,3),IRD(16,4),IRD(16,5), LOT4500  
 1 IRD(16,6),IRD(17,1),IRD(17,2),IRD(17,3),IRD(17,4),IRD(17,5), LOT4510  
 2 IRD(17,6)/ LOT4520  
 3 13550, 0,14431, 0,21,1, 4374, 0,12165, 1487,24,5/ LOT4530  
 C  
 C ELLIPT FIRST ELLIPT SECOND ZEROS BJZERO ZEROS BJONE LOT4540  
 C STRUVE ZERO STRUVE ONE LOT4550  
 C  
 C DATA IRD(18,1),IRD(18,2),IRD(18,3),IRD(18,4),IRD(18,5), LOT4560  
 1 IRD(18,6),IRD(19,1),IRD(19,2),IRD(19,3),IRD(19,4),IRD(19,5), LOT4570  
 2 IRD(19,6),IRD(20,1),IRD(20,2),IRD(20,3),IRD(20,4),IRD(20,5), LOT4580  
 3 IRD(20,6),IRD(21,1),IRD(21,2),IRD(21,3),IRD(21,4),IRD(21,5), LOT4590  
 4 IRD(21,6),IRD(22,1),IRD(22,2),IRD(22,3),IRD(22,4),IRD(22,5), LOT4600

```

5 IRD(22,6),IRD(23,1),IRD(23,2),IRD(23,3),IRD(23,4),IRD(23,5),    LOT4730
6 IRD(23,6)/    LOT4740
7 3981, 7013, 4635,14391,30,30, 3981, 7013,13989,11317,30,31,    LOT4750
8 19107,11448, 1754, 4146,30,33,19107,11448, 1743,10341,30,34,    LOT4760
9 14409,15908,19107,10935,30,35,14409,15908,11318, 0,30,36/    LOT4770
C
C
C
      DATA      IRD(24,1),IRD(24,2),IRD(24,3),IRD(24,4),IRD(24,5),    LOT4810
1 IRD(24,6)/    LOT4820
2 0, 0, 0, 0, 0/    LOT4830
C
C
      PAGE PLOT    LOT4840
C
      DATA      IRD(25,1),IRD(25,2),IRD(25,3),IRD(25,4),IRD(25,5),    LOT4870
1 IRD(25,6)/    LOT4880
2 11698, 3645,12003,14580,13, 6/    LOT4890
C
C
      GAUSS QUADRATURE    LOT4900
C
      DATA      IRD(26,1),IRD(26,2),IRD(26,3),IRD(26,4),IRD(26,5),    LOT4930
1 IRD(26,6)/    LOT4940
2 5151,14364,12961, 3403,24, 4/    LOT4950
C
C
      DUMMY E F UNIFOR RANDOM    LOT4960
C
      DATA      ITP(1,1),ITP(1,2),ITP(1,3),ITP(1,4),ITP(2,1),    LOT4990
1 ITP(2,2),ITP(2,3),ITP(2,4),ITP(3,1),ITP(3,2),ITP(3,3),    LOT5000
2 ITP(3,4),ITP(4,1),ITP(4,2),ITP(4,3),ITP(4,4),ITP(5,1),    LOT5010
3 ITP(5,2),ITP(5,3),ITP(5,4),ITP(6,1),ITP(6,2),ITP(6,3),    LOT5015
4 ITP(6,4),ITP(7,1),ITP(7,2),ITP(7,3),ITP(7,4),ITP(8,1),    LOT5020
5 ITP(8,2),ITP(8,3),ITP(8,4),ITP(9,1),ITP(9,2),ITP(9,3),    LOT5030
6 ITP(9,4)/    ****LOT5040
*
* USED THRU ITP(9,4): AVAILABLE ITP(10,1) THRU ITP(10,4)    *LOT5050
*
***** THE FOLLOWING CARD IS NEEDED ONLY FOR TAPE OPERATIONS    *LOT5060
*****
* READ TAPE,CREAD TAPE,WRITE TAPE,SETTAPE,CSET TAPE,ENDFILE TAPE,    LOT5110
* REWIND TAPE, SKIP TAPE,BACKSPACE TAPE    LOT5120
*****
      DATA      ITP(1,1),ITP(1,2),ITP(1,3),ITP(1,4),ITP(2,1),    LOT5140
1 ITP(2,2),ITP(2,3),ITP(2,4),ITP(3,1),ITP(3,2),ITP(3,3),    LOT5150
2 ITP(3,4),ITP(4,1),ITP(4,2),ITP(4,3),ITP(4,4),ITP(5,1),    LOT5160
3 ITP(5,2),ITP(5,3),ITP(5,4),ITP(6,1),ITP(6,2),ITP(6,3),    LOT5170
4 ITP(6,4),ITP(7,1),ITP(7,2),ITP(7,3),ITP(7,4),ITP(8,1),    LOT5180
5 ITP(8,2),ITP(8,3),ITP(8,4),ITP(9,1),ITP(9,2),ITP(9,3),    LOT5190
6 ITP(9,4)/    LOT5200
7 13258, 2916,45,1, 2678, 837,46,1,17262,14715,47,1,    LOT5210
8 14006, 0,48,1, 2705,14580,49,1, 4027, 4629,50,1,    LOT5220
9 13280, 6943,50,2,14157,11664,50,3, 1488, 8548,50,4/    LOT5230
*****
***** USED THRU ITP(9,4): AVAILABLE ITP(10,1) THRU ITP(10,4)*****LOT5280
*****
***** END*****LOT5300

```

```

SUBROUTINE LOOKUP                                LOU 10
C VERSION 5.00      LOOKUP      5/15/70          LOU 20
C WRITTEN BY S PEAVY    3/14/68          LOU 30
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG          LOU 40
COMMON /ICODE/ NIR,NID,NIRD,LIR,LID,LIRD        LOU 50
COMMON /CODE/ IALPH(6),NALPH(5),ID(9,3),IR(300,4),IRD(30,6) LOU 60
C   THE FOLLOWING CARDS ARE NEEDED ONLY FOR THE OPERATIONS LOU 70
COMMON /ICODTP/ NITP,LITP                      LOU 80
COMMON /CODETP/ ITP(10,4)                      LOU 90
COMMON /TAPE/ NAME4(2),NTPCT,IPUNCP,INUNIP,L1TP        LOU 100
C THIS SUBROUTINE CHECKS TO SEE IF FIRST WORD AND SOMETIMES SECOND LOU 102
C WORD ON COMMAND CARD IS A LEGITIMATE COMMAND          LOU 104
C   IF COMMAND IS FOUND L1 AND L2 ARE ASSIGNED A VALUE      LOU 106
C   IF COMMAND IS NOT FOUND IN THE DICTIONARY L1 IS SET EQUAL TO ZERO LOU 108
C ****
C L1=0                                              LOU 120
DO 10 I=1,NIR                                     LOU 130
IF (NAME(1).NE.IR(I,1).OR.NAME(2).NE.IR(I,2)) GO TO 10 LOU 140
L1=IR(I,3)                                         LOU 150
L2=IR(I,4)                                         LOU 160
C   THE FOLLOWING CARDS ARE NEEDED ONLY FOR TAPE OPERATIONS LOU 170
IF (NAME(1).NE.IR(50,1)) RETURN                  LOU 180
C   NAME(3)=TAP   NAME(4)=E                      LOU 190
IF (NAME(3).EQ.14623.AND.NAME(4).EQ.3645) GO TO 40 LOU 200
C ****
C RETURN                                            LOU 220
10 CONTINUE                                         LOU 230
DO 20 I=1,NID                                     LOU 240
IF (NAME(1).NE.ID(I,1).OR.NAME(2).NE.ID(I,2)) GO TO 20 LOU 250
L1=ID(I,3)                                         LOU 260
GO TO 80                                           LOU 270
20 CONTINUE                                         LOU 280
DO 30 I=1,NIRD                                    LOU 290
IF (NAME(1).NE.IRD(I,1).OR.NAME(2).NE.IRD(I,2).OR.NAME(3).NE.IRD(I,3).OR.NAME(4).NE.IRD(I,4)) GO TO 30 LOU 300
L1=IRD(I,5)                                         LOU 310
L2=IRD(I,6)                                         LOU 320
RETURN                                            LOU 330
30 CONTINUE                                         LOU 340
C   THE FOLLOWING CARDS ARE NEEDED ONLY FOR TAPE OPERATIONS LOU 350
40 DO 50 I=1,NITP                                 LOU 360
IF (NAME(1).NE.ITP(I,1).OR.NAME(2).NE.ITP(I,2)) GO TO 50 LOU 370
L1=ITP(I,3)                                         LOU 380
L2=ITP(I,4)                                         LOU 390
GO TO 60                                           LOU 400
50 CONTINUE                                         LOU 410
RETURN                                            LOU 420
60 IF (L1.GT.47) RETURN                          LOU 430
DO 70 I=1,6                                         LOU 440
IF (NAME4(1).NE.IALPH(I)) GO TO 70                LOU 450
L2=I+1                                             LOU 460
RETURN                                            LOU 470
70 CONTINUE                                         LOU 480
C ****
C RETURN                                            LOU 490
C   THE FOLLOWING CARDS ARE NEEDED ONLY FOR TAPE OPERATIONS LOU 500
80 IF (L1.NE.5) GO TO 90                          LOU 510
C   NAME(3)=TAP   NAME(4)=E                      LOU 520
IF (NAME(3).EQ.14623.AND.NAME(4).EQ.3645) GO TO 40 LOU 530
C   ****
C ****

```

C THIS CARD WAS 500 L2=1 LOU 560  
90 L2=1 LOU 570  
C \*\*\*\* LOU 580  
IF (L1.EQ.1) GO TO 110 LOU 590  
DO 100 I=1,6 LOU 600  
IF (NAME(3).NE.IALPH(I)) GO TO 100 LOU 610  
L2=I+1 LOU 620  
RETURN LOU 630  
100 CONTINUE LOU 640  
IF (L1.NE.2) RETURN LOU 641  
IF (L2.NE.1) RETURN LOU 642  
IF (NAME(3).NE.10631.AND.NAME(4).NE.3645) RETURN LOU 643  
C THE COMMAND IS PRINT NOTE LOU 644  
L1=13 LOU 645  
L2=13 LOU 646  
RETURN LOU 650  
110 DO 120 I=1,5 LOU 660  
IF (NAME(3).NE.NALPH(I)) GO TO 120 LOU 670  
L2=I+2 LOU 680  
RETURN LOU 690  
120 CONTINUE LOU 700  
RETURN LOU 710  
END LOU 720

```

SUBROUTINE MATRIX                                MAT 10
C      VERSION 5.00      MATRIX      5/15/70      MAT 20
C
C      L2=1  ADD MATRICES A+B    MADD A(,) N,M, TO B(,) N,M AND S ORE IN C(MAT 40
C      L2=2  SUB MATRICES A-B    MSUB A(,) N,M FROM B(,)N,M AND STORE IN C(MAT 60
C      L2=3  TRANPOSE MATRIX MTRANS A(,) N,M AND STORE IN C(,)      MAT 80
C          TRANPOSE ARRAY ATRANS A(,) N,M AND STORE IN C(,)      MAT 90
C      L2=4  ARRAY ADD        AADD      MAT 100
C      L2=5  ARRAY SUBTRACT   ASUB      MAT 110
C      L2=6  ARRAY MULTIPLY   AMULT      MAT 120
C      L2=7  ARRAY DIVIDE     ADIV      MAT 130
C      L2=8  ARRAY RAISE     ARAISE      MAT 140
C
C          GENERAL FORMS FOR ARRAY OPERATIONS      MAT 150
C          A(,) N,M B(,) N,K STORE IN C(,) ARRAY BY ARRAY      MAT 160
C          A(,) N,M B(,)      STORE IN C(,) ARRAY BY ARRAY      MAT 170
C          A(,) N,M K      STORE IN C(,) ARRAY BY COLUMN      MAT 180
C          A(,) N,M X      STORE IN C(,) ARRAY BY CONSTANT      MAT 200
C
C          COMMON /BLOCRC/ NRC,RC(12600)      MAT 230
C          COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NMAT 240
C          IARGS,VWXYZ(8),NERROR      MAT 250
C          DIMENSION ARGS(100)      MAT 260
C          EQUIVALENCE (ARGS(1),RC(12501))      MAT 270
C          COMMON /SCRAT/ NS,NS2,A(13500)      MAT 280
C          COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG      MAT 290
C
C          CHECK TO SEE IF WE HAVE CORRECT NUMBER OF ARGUMENTS      MAT 310
C
C          NP=NARGS      MAT 320
C          IF (L2-3) 10,20,30      MAT 330
10         IF (NARGS.NE.8.AND.NARGS.NE.10) GO TO 40      MAT 340
          GO TO 50      MAT 350
20         IF (NARGS.NE.6) GO TO 40      MAT 360
          GO TO 50      MAT 370
30         IF(NARGS.LT.7.OR.NARGS.GT.10.OR.NARGS.EQ.9) GO TO 40      MAT 380
          GO TO 50      MAT 390
40         CALL ERROR (10)      MAT 400
          RETURN      MAT 410
C
C          CHECK TO SEE IF ALL ARGUMENTS ARE INTEGERS      MAT 420
C
C          IF(L2.GT.3.AND.NARGS.EQ.7) GO TO 70      MAT 430
50
60         J=NARGS      MAT 440
          CALL CKIND (J)      MAT 450
          IF (J.EQ.0) GO TO 80      MAT 460
          CALL ERROR (3)      MAT 470
          GO TO 80      MAT 480
70         ISAVE=KIND(NARGS)      MAT 490
          KIND(NARGS)=KIND(NARGS-2)      MAT 500
          KIND(NARGS-2)=KIND(NARGS-1)      MAT 510
          KIND(NARGS-1)=ISAVE      MAT 520
          NARGS=NARGS-1      MAT 530
          GO TO 60      MAT 540
C
C          CHECK TO SEE IF DIMENSIONS ARE CORRECT IF THEY ARE GIVEN      MAT 550
C
C          IF (NP.NE.10) GO TO 90      MAT 560
80
          IF (IARGS(3).EQ.IARGS(7).AND.IARGS(4).EQ.IARGS(8)) GO TO 90      MAT 570
          CALL ERROR (3)      MAT 580
C
          MAT 590
          IF (NP.NE.10) GO TO 90      MAT 600
          IF (IARGS(3).EQ.IARGS(7).AND.IARGS(4).EQ.IARGS(8)) GO TO 90      MAT 610
          CALL ERROR (3)      MAT 620

```

```

RETURN                                MAT 625
C
C   CHECK TO SEE IF ARGUMENTS ARE OUT OF RANGE      MAT 630
C
90    IF((L2.LT.3.OR.L2.GT.3).AND.KIND(NP).EQ.0) GO TO 100      MAT 640
J=2
GO TO 130                                MAT 660
100   J=3
IARGS(12)=IARGS(4)                        MAT 670
120   IARGS(11)=IARGS(3)                      MAT 680
      IF (NP.EQ.10) GO TO 210                  MAT 690
      IARGS(10)=IARGS(NP)
      IARGS(9)=IARGS(NP-1)
130   IF (NP.EQ.8.OR.(NP.EQ.7.AND.KIND(NP).NE.0)) GO TO 190      MAT 700
      IF(NP.EQ.6) GO TO 160                  MAT 720
      IARGS(6)=IARGS(5)
      IARGS(8)=1
      IARGS(7)=IARGS(3)
      IARGS(5)=1
      GO TO 210                                MAT 750
160   IARGS(8)=IARGS(3)                      MAT 760
      IARGS(7)=IARGS(4)
      GO TO 210                                MAT 770
190   IF (NP.EQ.8) GO TO 200                  MAT 780
      IARGS(5)=IARGS(6)
      IARGS(6)=IARGS(7)
200   IARGS(8)=IARGS(4)                      MAT 790
      IARGS(7)=IARGS(3)
210   CALL MTXCHK (J)                      MAT 810
      IF (J-1) 240,220,230
220   CALL ERROR (3)                      MAT 860
      RETURN
230   CALL ERROR (17)                      MAT 870
      RETURN
C   *
C   CHECK TO SEE IF THERE WERE PREVIOUS ERRORS      MAT 880
C   *
240   IF (NERROR.NE.0) RETURN                MAT 900
C   *
C   SUM ELEMENTS IN SCRATCH AREA              MAT 910
C   SUBTRACT ELEMENTS IN SCRATCH AREA          MAT 920
C   PRODUCTS AND QUOTIENTS FORMED USING DOUBLE PRECISION IN SCRATCH ARMAT1190
C   TRANPOSE IN SCRATCH AREA                  MAT 1040
C   *
IROW=IARGS(3)                            MAT 1050
ICOL=IARGS(4)                            MAT 1060
NROWPP=NROW
IF (L2-3) 260,250,290
250   IIB=ICOL
JJB=IROW
NROWPP=0
K=1
GO TO 280
260   NROWP=NROW
IBP=IARGS(5)
270   IIB=IROW
JJB=ICOL
K=0
280   IS=1

```

	IAP=IARGS(1)	MAT1370
	GO TO 320	MAT1380
290	IF (NP.GE.8) GO TO 260	MAT1390
	IF (KIND(NP).EQ.1) GO TO 300	MAT1400
	IBP=IARGS(5)	MAT1410
	GO TO 310	MAT1420
300	IARGS(9)=IARGS(5)	MAT1430
310	NROWP=0	MAT1440
	GO TO 270	MAT1450
320	DO 510 J=1,JJB	MAT1460
C	COMPUTE ADDRESSES	MAT1465
	IA=IAP+(J-1)*K	MAT1470
	IB=IBP	MAT1480
	DO 500 I=1,IIB	MAT1490
	GO TO (330,340,370,380,390,400,410,420), L2	MAT1500
330	A(IS)=RC(IA)+RC(IB)	MAT1510
	GO TO 470	MAT1520
340	A(IS)=RC(IA)-RC(IB)	MAT1530
	GO TO 470	MAT1540
350	A(IS)=RC(IA)*RC(IB)	MAT1550
	GO TO 470	MAT1560
360	IF(RC(IB).EQ.0.0) GO TO 365	MAT1562
	A(IS)=RC(IA)/RC(IB)	MAT1564
	GO TO 470	MAT1566
365	A(IS)=0.0	MAT1570
	GO TO 470	MAT1580
370	A(IS)=RC(IA)	MAT1590
	IA=IA+NROW	MAT1600
	GO TO 490	MAT1610
380	IF (NP.GE.8.OR.(KIND(NP).EQ.0.AND.NP.LT.8)) GO TO 330	MAT1620
	A(IS)=RC(IA)+ARGS(NP-2)	MAT1630
	GO TO 480	MAT1640
390	IF (NP.GE.8.OR.(KIND(NP).EQ.0.AND.NP.LT.8)) GO TO 340	MAT1650
	A(IS)=RC(IA)-ARGS(NP-2)	MAT1660
	GO TO 480	MAT1670
400	IF (NP.GE.8.OR.(KIND(NP).EQ.0.AND.NP.LT.8)) GO TO 350	MAT1680
	A(IS)=RC(IA)*ARGS(NP-2)	MAT1690
	GO TO 470	MAT1700
410	IF (NP.GE.8.OR.(NP.LT.8.AND.KIND(NP).EQ.0)) GO TO 360	MAT1710
	IF(ARGS(NP-2).EQ.0.0) GO TO 415	MAT1712
	A(IS)=RC(IA)/ARGS(NP-2)	MAT1714
	GO TO 470	MAT1716
415	A(IS)=0.0	MAT1720
	GO TO 470	MAT1730
420	IF (NP.GE.8.OR.(NP.LT.8.AND.KIND(NP).EQ.0)) GO TO 440	MAT1740
	IF (RC(IA)) 430,460,430	MAT1750
430	A(IS)=FEXP2(RC(IA),ARGS(NP-2))	MAT1760
	GO TO 470	MAT1770
440	IF (RC(IA)) 450,460,450	MAT1780
450	A(IS)=FEXP2(RC(IA),RC(IB))	MAT1790
	GO TO 470	MAT1800
460	A(IS)=0.0	MAT1810
470	IB=IB+1	MAT1820
480	IA=IA+1	MAT1830
490	IS=IS+1	MAT1840
500	CONTINUE	MAT1850
	IAP=IAP+NROWPP	MAT1860
	IBP=IBP+NROWP	MAT1870
510	CONTINUE	MAT1880

C	*	MAT1890
C	MOVE SUMS TO WORKSHEET	MAT1900
C	MOVE DIFFERENCES TO WORKSHEET	MAT1910
C	MOVE ARRAY PRODUCT TO WORKSHEET	MAT1920
C	MOVE ARRAY QUOTIENT TO WORKSHEET	MAT1930
C	MOVE TRANSPOSE TO WORKSHEET	MAT1940
C	MOVE RAISED MATRIX TO WORKSHEET	MAT1950
C	*	MAT1960
	IF (L2.NE.3) GO TO 520	MAT1970
	ICP=IARGS(5)	MAT1980
	GO TO 530	MAT1990
520	ICP=IARGS(9)	MAT2000
530	IS=1	MAT2010
	DO 550 J=1,JJB	MAT2020
	IC=ICP	MAT2030
	DO 540 I=1,IIB	MAT2040
	RC(IC)=A(IS)	MAT2050
	IC=IC+1	MAT2060
	IS=IS+1	MAT2070
540	CONTINUE	MAT2080
	ICP=ICP+NROW	MAT2090
550	CONTINUE	MAT2100
	RETURN	MAT2110
	END	MAT2120

```

SUBROUTINE MDAMAD                               MDA 10
C      VERSION 5.00      MDAMAD      5/15/70      MDA 20
C      SUBROUTINE MDAMAD                      R VARNER 9/26/67      MDA 30
C      *
C      SUBROUTINE TO PRE OR POST MULTIPLY A MATRIX BY A DIAGONAL STORED      MDA 50
C      AS A COLUMN                                         MDA 60
C      L2=1      M(AD)                                         MDA 70
C          MATRIX A IS POSTMULTIPLIED BY THE DIAGONAL D STORED IN COL I      MDA 80
C          GENERAL FORM OF COMMAND                                         MDA 90
C          M(AD) A(,) N,K, D IN COL I STORE IN C(,)                         MDA 100
C      L2=2      M(AD)                                         MDA 110
C          MATRIX A IS PREMULTIPLIED BY THE DIAGONAL D STORED IN COL I      MDA 120
C          GENERAL FORM OF COMMAND                                         MDA 130
C          M(DA), A(,) N,K K IN COL I STORE IN C(,)                         MDA 140
C      *
C      COMMON /SCRAT/ NS,NS2,A(13500)                                MDA 160
C      COMMON /BLOCRC/ NRC,RC(12600)                                MDA 170
C      COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NMDA 180
IARGS,VWXYZ(8),NERROR                                         MDA 190
DIMENSION ARGS(100)                                         MDA 200
EQUIVALENCE (ARGS(1),RC(12501))                                MDA 210
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG                           MDA 220
C
C      CHECK FOR CORRECT NUMBER OF ARGUMENTS                         MDA 230
C      *
C      IF (NARGS.NE.7) CALL ERROR (10)                                MDA 240
C      *
C      CHECK TO SEE THAT ALL ARGUMENTS ARE INTEGERS                  MDA 250
C      *
J=NARGS                                         MDA 300
CALL CKIND (J)                                         MDA 310
IF (J.NE.0) CALL ERROR (3)                                MDA 320
C
C      CHECK TO SEE IF DIMENSIONS ARE OUT OF RANGE                 MDA 330
C      COMPUTE ADDRESSES OF COLUMNS                                MDA 340
C      *
IARGS(12)=IARGS(4)                                         MDA 350
IARGS(11)=IARGS(3)                                         MDA 360
IARGS(10)=IARGS(7)                                         MDA 370
IARGS(9)=IARGS(6)                                         MDA 380
IARGS(8)=1                                              MDA 390
GO TO (10,20), L2                                         MDA 400
10   IARGS(7)=IARGS(4)                                         MDA 410
      GO TO 30                                         MDA 420
20   IARGS(7)=IARGS(3)                                         MDA 430
30   IARGS(6)=IARGS(5)                                         MDA 440
      IARGS(5)=1                                              MDA 450
      J=3                                              MDA 460
      CALL MTXCHK (J)                                         MDA 470
      IF (J-1) 60,40,50                                         MDA 480
40   CALL ERROR (3)                                         MDA 490
      RETURN                                         MDA 500
50   CALL ERROR (17)                                         MDA 510
      RETURN                                         MDA 520
C
C      CHECK FOR PREVIOUS ERRORS                                 MDA 530
C      *
60   IF (NERROR.NE.0) RETURN                                MDA 540
      IP=IARGS(4)                                         MDA 550
      *
      CHECK FOR PREVIOUS ERRORS                                MDA 560
      *
      IF (NERROR.NE.0) RETURN                                MDA 570
      IP=IARGS(4)                                         MDA 580
      *
      IF (NERROR.NE.0) RETURN                                MDA 590
      IP=IARGS(4)

```

	JP=IARGS(3)	MDA 600
	GO TO (80,70), L2	MDA 610
70	I1=0	MDA 620
	I2=1	MDA 630
	GO TO 90	MDA 640
80	I1=1	MDA 650
	I2=0	MDA 660
90	IA=IARGS(1)	MDA 670
	IDP=IARGS(5)	MDA 680
	IB=IARGS(9)	MDA 690
	DO 110 I=1,IP	MDA 700
	ID=IDP	MDA 710
	DO 100 J=1,JP	MDA 720
	RC(IB)=RC(ID)*RC(IA)	MDA 730
	ID=ID+I2	MDA 740
	IA=IA+1	MDA 750
	IB=IB+1	MDA 760
100	CONTINUE	MDA 770
	IB=IB+NROW-JP	MDA 780
	IA=IA+NROW-JP	MDA 790
	IDP=IDP+I1	MDA 800
110	CONTINUE	MDA 810
	RETURN	MDA 820
	END	MDA 830

```

SUBROUTINE MEIGEN                                     MEI 10
C VERSION 5.00      MEIGEN      5/15/70               MEI 20
C SUBROUTINE MEIGEN WRITTEN BY R VARNER   4/4/68     MEI 30
C *
C SUBROUTINE TO COMPUTE EIGENVALUES AND EIGENVECTORS MEI 40
C           GENERAL FORMS OF COMMANDS                 MEI 50
C             MEIGEN A(,,++) R=,, C=,, STORE VALUES IN COL ++
C             MEIGEN A(,,++) R=,, C=,, STORE VECTORS IN B(,,++)
C             MEIGEN A(,,++) R=,, C=,, STORE VALUES IN COL ++
C                           STORE VECTORS IN B(,,++)
C BOTH R AND C MUST BE SPECIFIED                   MEI 110
C NARGS= 5 COMPUTE ONLY EIGENVALUES                MEI 120
C NARGS= 6 COMPUTE ONLY EIGENVECTORS                MEI 130
C NARGS= 7 COMPUTE EIGENVALUES AND VECTORS          MEI 140
C *
C COMMON /BLOCRC/ NRC,RC(12600)                     MEI 160
C COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NMEI 170
C IARGS,VWXYZ(8),NERROR                            MEI 180
C DIMENSION ARGS(100)                               MEI 190
C EQUIVALENCE (ARGS(1),RC(12501))                  MEI 200
C COMMON /SCRAT/ NS,NS2,A(13500)                    MEI 210
C DIMENSION ISWCH(2)                                MEI 220
C *
C CHECK TO BE SURE THAT MATRIX IS NO BIGGER THAN 54X54 MEI 240
C *
C IF (IARGS(3).NE.IARGS(4)) CALL ERROR (230)        MEI 260
C IF (IARGS(3)**2.GT.NS2) CALL ERROR (23)            MEI 270
C *
C CHECK FOR CORRECT NUMBER OF ARGUMENTS             MEI 280
C *
C IF (NARGS.LT.5.OR.NARGS.GT.7) CALL ERROR (10)       MEI 310
C *
C CHECK TO SEE IF ARGUMENTS ARE ALL INTEGERS         MEI 330
C *
C J=NARGS                                         MEI 340
C CALL CKIND (J)                                    MEI 350
C IF (J.NE.0) CALL ERROR (3)                         MEI 360
C *
C CHECK TO SEE IF DIMENSIONS ARE OUT OF RANGE       MEI 380
C COMPUTE ADDRESSES                                 MEI 390
C *
C ISWCH(2)=NARGS-4                                MEI 400
C ISWCH(1)=0                                       MEI 410
C IF (NARGS.EQ.6) GO TO 10                          MEI 420
C IADD=1                                         MEI 430
C CALL ADDRESS (5,J)                                MEI 440
C IF (J.LE.0) CALL ERROR (11)                         MEI 450
C *
C J CONTAINS ADDRESS OF COLUMN                      MEI 460
C *
C IF (NARGS.EQ.5) GO TO 20                          MEI 470
C *
C IARGS(5)=IARGS(6)                                MEI 480
C IARGS(6)=IARGS(7)                                MEI 490
C *
C IADD=2                                         MEI 500
C IARGS(7)=IARGS(3)                                MEI 510
C IARGS(8)=IARGS(4)                                MEI 520
C GO TO 30                                         MEI 530
C *
C ISWCH(1)=1                                       MEI 540
C CALL MTXCHK (IADD)                                MEI 550
C *
C
10
20
30

```

```

40  IF (IADD-1) 60,40,50               MEI 600
    CALL ERROR (3)                      MEI 610
    RETURN                               MEI 620
50  CALL ERROR (17)                   MEI 630
    RETURN                               MEI 640
C   *                                  MEI 650
C   CHECK FOR PREVIOUS ERRORS        MEI 660
C   *                                  MEI 670
60  IF (NERROR.NE.0) RETURN          MEI 680
    IGP=IARGS(5)                      MEI 690
    IG=IARGS(1)                       MEI 700
    CALL HDIAG (RC(IG),IARGS(3),ISWCH,A,RC(J),RC(IGP),NROW,A(3000)) MEI 710
C   *                                  MEI 720
C   RC(IG) IS LOCATION OF MATRIX TO BE DIAGONALIZED MEI 730
C   IARG(3) GIVES SIZE OF MATRIX      MEI 740
C   ISWCH (1)=1           IF ONLY EIGENVALUES ARE TO BE COMPUTED MEI 750
C   ISWCH (1)=0           COMPUTE EIGENVALUES AND EIGENVECTORS MEI 760
C   ISWCH (2) = NARGS-4     AND IS USED FOR STORING RESULTS MEI 770
C   A IS LOCATION OF SCRATCH AREA    MEI 780
C   RC(J) TELLS WHERE TO STORE EIGENVALUES MEI 790
C   RC(IGP)   IS WHERE EIGENVECTORS ARE STORED MEI 800
C   *                                  MEI 810
    RETURN                           MEI 820
    END                             MEI 830

```

```

SUBROUTINE MISC2                                MIS 10
C      VERSION 5.00      MISC2      5/15/70        MIS 20
COMMON /BLOCRC/ NRC,RC(12600)                  MIS 30
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NMIS 40
IARGS,VWXYZ(8),NERROR                         MIS 50
DIMENSION ARGS(100)                           MIS 60
EQUIVALENCE (ARGS(1),RC(12501))              MIS 70
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG          MIS 80
COMMON /SCRAT/ NS,NS2,A(13500)                MIS 90
C      SUBROUTINE BY CARLA MESSINA   NSRDS - NBS JULY 1967    MIS 100
C                                         MIS 110
C      L2=4 IS EXPAND $$ TO ,, POWER IN INTERVALS OF ,, START STORE +MIS 120
C             THE POWERS MAY BE INTEGER OR NOT                   MIS 130
C      L2=1 IS CLOSE UP ROWS HAVING ** IN ++,++, ETC           MIS 140
C      L2=2 IS COUNT LENGTH OF COLUMN ++, STORE IN COLUMN ++    MIS 150
C      L2=3 IS SHORTEN COL ++ FOR COL ++ = ** STORE IN COL ++ AND COLMIS 160
C      L2=5 IS DUPLICATE ,, TIMES THE ARRAY IN ,, ++ R=,, C=,, START MIS 170
C             STORING IN ,, ++                               MIS 180
C                                         MIS 190
C      IF (NARGS-2) 10,40,40                                MIS 200
10     K=10                                         MIS 210
20     CALL ERROR (K)                                 MIS 220
30     RETURN                                       MIS 230
40     GO TO (50,80,50,340,540), L2                  MIS 240
50     IF (KIND(L2)) 60,60,70                      MIS 250
60     K=3                                         MIS 260
       GO TO 20                                     MIS 270
70     KIND(L2)=0                                    MIS 280
       ARG1=ARGS(L2)                                MIS 290
       IARGS(L2)=IARGS(L2+1)                          MIS 300
80     CALL CHKCOLUMN (J)                            MIS 310
       IF (J) 60,90,60                                MIS 320
90     DO 100 I=1,NARGS                            MIS 330
100    IARGS(I)=IARGS(I)-1                        MIS 340
       IF (L2-2) 120,120,110                      MIS 350
110    IF (NARGS-5) 10,120,10                      MIS 360
120    IF (NERROR.NE.0) GO TO 30                  MIS 370
       IF (NRMAX) 130,130,140                      MIS 380
130    K=9                                         MIS 390
       GO TO 20                                     MIS 400
140    IF (L2-2) 150,210,250                      MIS 410
C     CLOSE UP                                     MIS 420
150    DO 200 J=2,NARGS                            MIS 430
       K=IARGS(J)                                  MIS 440
       M=1                                         MIS 450
       DO 170 I=1,NRMAX                            MIS 460
       J1=K+I                                     MIS 470
       IF (RC(J1)-ARG1) 160,170,160                  MIS 480
160    K1=K+M                                     MIS 490
       RC(K1)=RC(J1)                                MIS 500
       M=M+1                                     MIS 510
170    CONTINUE                                     MIS 520
       IF (M-NRMAX) 180,180,200                  MIS 530
180    DO 190 I=M,NRMAX                            MIS 540
       J1=K+I                                     MIS 550
190    RC(J1)=0.0                                  MIS 560
200    CONTINUE                                     MIS 570
       GO TO 30                                     MIS 580
C     COUNT                                       MIS 590

```

```

210 J1=NRMAX MIS 600
J=IARGS(1)+NRMAX MIS 610
DO 230 I=1,NRMAX MIS 620
IF (RC(J)) 240,220,240 MIS 630
220 J1=J1-1 MIS 640
230 J=J-1 MIS 650
240 ARG1=J1 MIS 660
    IARGS(2)=IARGS(2)+1 MIS 670
    CALL VECTOR (ARG1,IARGS(2)) MIS 680
    GO TO 30 MIS 690
C SHORTEN MIS 700
250 IF (NRMAX-2) 30,260,260 MIS 710
260 DO 290 K=2,NRMAX MIS 720
    J1=IARGS(2)+K MIS 730
    IF(ARG1-RC(J1-1)) 270,300,280 MIS 740
270 IF(ARG1-RC(J1)) 290,310,310 MIS 750
280 IF(ARG1-RC(J1)) 310,310,290 MIS 760
290 CONTINUE MIS 770
    K=203 MIS 780
    CALL ERROR (K) MIS 790
    GO TO 320 MIS 800
300 NRMAX=K-1 MIS 810
    GO TO 320 MIS 820
310 NRMAX=K MIS 830
320 DO 330 I=1,NRMAX MIS 840
    K=IARGS(1)+I MIS 850
    J=IARGS(4)+I MIS 860
    M=IARGS(5)+I MIS 870
    K1=IARGS(2)+I MIS 880
    RC(M)=RC(K1) MIS 890
    RC(J)=RC(K) MIS 900
    GO TO 30 MIS 910
C EXPAND MIS 920
340 IF (NARGS-4) 10,350,10 MIS 930
350 CALL ADRESS (4,K1) MIS 940
    IF (K1) 60,60,360 MIS 950
360 IF (KIND(1)) 400,370,400 MIS 960
370 CALL ADRESS (1,IARGS(1)) MIS 970
    IF (IARGS(1)) 60,60,380 MIS 980
380 K=IARGS(1)-1 MIS 990
    DO 390 I=1,NRMAX MIS1000
    J=K+I MIS1010
390 A(I)=RC(J) MIS1020
    GO TO 420 MIS1030
400 DO 410 I=1,NRMAX MIS1040
410 A(I)=ARGS(1) MIS1050
420 IF (KIND(2)) 440,430,440 MIS1060
430 ARGS(2)=IARGS(2) MIS1070
440 IF (KIND(3)) 460,450,460 MIS1080
450 ARGS(3)=IARGS(3) MIS1090
460 IF (ARGS(2)*ARGS(3)) 470,470,480 MIS1100
470 K=20 MIS1110
    GO TO 20 MIS1120
480 IF (ABS(ARGS(3))-ABS(ARGS(2))) 490,490,470 MIS1130
490 IF (NERROR.NE.0) GO TO 30 MIS1140
    IF (NRMAX) 130,130,500 MIS1150
500 CC=ARGS(3) MIS1160
510 DO 520 I=1,NRMAX MIS1170
    K=K1-1+I MIS1180

```

520	RC(K)=FEXP2(A(I),CC)	MIS1190
	IF (ABS(CC)+.5E-6-ABS(ARGS(2))) 530,30,30	MIS1200
530	CC=CC+ARGS(3)	MIS1210
	IARGS(4)=IARGS(4)+1	MIS1220
	CALL ADDRESS (4,K1)	MIS1230
	IF (K1) 60,60,510	MIS1240
C	DUPLICATE	MIS1250
540	IF (NARGS.NE.7) GO TO 10	MIS1260
	IF (IARGS(1).LE.0) GO TO 60	MIS1270
	K1=MAX0(IARGS(1)*IARGS(4)+IARGS(6)-1,NRMAX)	MIS1280
	IF (K1.GT.NROW) GO TO 590	MIS1290
	J=7	MIS1300
	CALL CKIND (J)	MIS1310
	IF (J.NE.0) GO TO 60	MIS1320
	NARGS=6	MIS1330
	NDUP=IARGS(1)	MIS1340
	IARGS(61)=IARGS(6)	MIS1350
	IARGS(62)=IARGS(7)	MIS1360
	IARGS(63)=IARGS(4)	MIS1370
	IARGS(64)=IARGS(5)	MIS1380
	IARGS(65)=IARGS(6)	MIS1390
	IARGS(66)=IARGS(7)	MIS1400
	DO 550 I=1,6	MIS1410
550	IARGS(I)=IARGS(I+1)	MIS1420
	CALL MOVE	MIS1430
	IF (NDUP.EQ.1) GO TO 580	MIS1440
	DO 570 I=2,NDUP	MIS1450
	DO 560 J=1,6	MIS1460
560	IARGS(J)=IARGS(J+60)	MIS1470
	IARGS(5)=IARGS(65)+(I-1)*IARGS(63)	MIS1480
570	CALL MOVE	MIS1490
580	NRMAX=K1	MIS1500
	GO TO 30	MIS1510
590	K=16	MIS1520
	GO TO 20	MIS1530
	END	MIS1540

```

SUBROUTINE MIST (M,B,LCHK,NLA,IND) MST 10
C VERSION 5.00      MIST      5/15/70 MST 20
DIMENSION B(1), NBC(12) MST 30
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NMST 40
IARGS,VWXYZ(8),NERROR MST 50
COMMON /HEADER/ NOCARD(80),ITLE(60,6),LNCNT,IPRINT,NPAGE,IPUNCH MST 60
MM=M-2 MST 70
NN=NRMAX-3 MST 80
NC=10 MST 90
M1=(M-1)/NC+1 MST 100
NLU=56 MST 110
NCA=0 MST 120
NRA=0 MST 130
DO 120 KEN=1,M1 MST 140
NCP=MIN0(NC,M-NCA) MST 150
NRP=M-LCHK*NCA MST 160
NLP=(1+IND/7)*NRP+5 MST 170
IF (NLP.LT.NLU-NLA) GO TO 10 MST 180
CALL PAGE (4) MST 190
NLA=0 MST 200
10 DO 20 IYA=1,NCP MST 210
I1=NCA+IYA MST 220
20 NBC(IYA)=(IARGS(I1+1)-1)/NROW+1 MST 230
GO TO (30,40,50,60,70,80,90), IND MST 240
30 WRITE (IPRINT,130) MST 250
GO TO 100 MST 260
40 WRITE (IPRINT,140) MST 270
GO TO 100 MST 280
50 WRITE (IPRINT,150) MM MST 290
GO TO 100 MST 300
60 WRITE (IPRINT,160) MST 310
GO TO 100 MST 320
70 WRITE (IPRINT,170) MST 330
GO TO 100 MST 340
80 I1=(IARGS(NRA+3)-1)/NROW+1 MST 350
I2=NCA*M+2 MST 360
WRITE (IPRINT,180) NN,B(I2),I1,NBC(1) MST 370
GO TO 100 MST 380
90 WRITE (IPRINT,190) MST 390
100 WRITE (IPRINT,200) (NBC(I),I=1,NCP) MST 400
WRITE (IPRINT,210) MST 410
DO 110 NAGA=1,NRP MST 420
NBR=NRA+NAGA MST 430
I1=NCA*M+NBR MST 440
I2=I1+(NCP-MAX0(0,LCHK*(NCP-NAGA))-1)*M MST 450
NBR=(IARGS(NBR+1)-1)/NROW+1 MST 460
WRITE (IPRINT,220) ,NBR,(B(I),I=I1,I2,M) MST 470
IF (IND.NE.7) GO TO 110 MST 480
I1=I1+2*M*M MST 490
I2=I2+2*M*M MST 500
WRITE (IPRINT,230) (B(I),I=I1,I2,M) MST 510
WRITE (IPRINT,210) MST 520
110 CONTINUE MST 530
NLA=NLA+NLP MST 540
NRA=NRA+LCHK*NCP MST 550
120 NCA=NCA+NCP MST 560
RETURN MST 570
C
130 FORMAT (/1H ,44X,31HSIMPLE CORRELATION COEFFICIENTS) MST 580
MST 590

```

```
140 FORMAT (/1H ,22X,75HSIGNIFICANCE LEVELS OF SIMPLE CORRELATION COEFMST 600
1FICIENTS (ASSUMING NORMALITY)) MST 610
150 FORMAT (/1H ,25X,37HPARTIAL CORRELATION COEFFICIENTS WITH,I3,26H RMST 620
1EMAINING VARIABLES FIXED) MST 630
160 FORMAT (/1H ,22X,76HSIGNIFICANCE LEVELS OF PARTIAL CORRELATION COEMST 640
1FFICIENTS (ASSUMING NORMALITY)) MST 650
170 FORMAT (/1H ,30X,58HSPEARMAN RANK CORRELATION COEFFICIENTS (ADJUSTMST 660
1ED FOR TIES)) MST 670
180 FORMAT (/1H ,8X,79HSIGNIFICANCE LEVEL OF QUADRATIC FIT OVER LINEARMST 680
1 FIT BASED ON F RATIO WITH 1 AND,I5,19H DEGREES OF FREEDOM/1H ,7X,MST 690
213H(FOR EXAMPLE,F7.4,60H IS THE SIGNIFICANCE LEVEL OF THE QUADRATIMST 700
3C TERM WHEN COLUMN,I3,20H IS FITTED TO COLUMN,I3,1H)) MST 710
190 FORMAT (/1H ,17X,86HCONFIDENCE INTERVALS FOR SIMPLE CORRELATION COMST 720
1EFFICIENTS (USING FISHER TRANSFORMATION)/30X,68H95 PER CENT LIMITSMST 730
2 BELOW DIAGONAL, 99 PER CENT LIMITS ABOVE DIAGONAL) MST 740
200 FORMAT (/1H ,6HCOLUMN,10I11) MST 750
210 FORMAT (1H ) MST 760
220 FORMAT (1H ,I4,4X,10F11.4) MST 770
230 FORMAT (1H ,6X,10F11.4) MST 780
END MST 790
```

```

SUBROUTINE MKRON          MKR   10
C VERSION 5.00      MKRON      5/15/70    MKR   20
C ROUTINE WRITTEN FOR OMNITAB 11/ 3/67 BY S PEAVY    MKR   30
C
C KRONECKER PRODUCT OF TWO MATRICES A(N,C)*B(M,K)=D    MKR   40
C
C FIRST FOUR ARGUMENTS DEFINE MATRIX A STARTING POS AND SIZE    MKR   50
C NEXT FOUR ARGUMENTS DEFINE MATRIX B STARTING POS AND SIZE    MKR   60
C LAST TWO ARGUMENTS INDICATE WHERE RESULT IS TO BE STORED D    MKR   70
C COMMAND IS:    MKR   80
C MKRON A(,, ++),R=,, C=,, *B(,, ++),R=,, C=,, STORE D(,, ++)    MKR   90
C
C
COMMON /BLOCRC/ NRC,RC(12600)    MKR  100
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NMKR 110
IARGS,VWXYZ(8),NERROR    MKR  120
DIMENSION ARGS(100)    MKR  130
EQUIVALENCE (ARGS(1),RC(12501))    MKR  140
COMMON /SCRAT/ NS,NS2,A(13500)    MKR  150
IF (NARGS.NE.10) CALL ERROR (10)
J=NARGS
CALL CKIND (J)
IF (J.NE.0) CALL ERROR (3)
IF (NERROR.NE.0) RETURN
IARGS(11)=IARGS(3)*IARGS(7)
IARGS(12)=IARGS(4)*IARGS(8)
J=3
CALL MTXCHK (J)
IF (J.EQ.0) GO TO 10
CALL ERROR (17)
RETURN
10 NRA=IARGS(3)
NCA=IARGS(4)
NRB=IARGS(7)
NCB=IARGS(8)
NDS=1
KA=IARGS(1)
DO 40 ICA=1,NCA
LA=IARGS(5)
DO 30 ICB=1,NCB
K=KA
DO 20 IRA=1,NRA
T=RC(K)
K=K+1
L=LA
DO 20 IRB=1,NRB
A(NDS)=T*RC(L)
L=L+1
20 NDS=NDS+1
30 LA=LA+NROW
40 KA=KA+NROW
NCR=IARGS(11)
NCC=IARGS(12)
NDS=1
KA=IARGS(9)
DO 60 I=1,NCC
K=KA
DO 50 J=1,NCR
RC(K)=A(NDS)

```

	NDS=NDS+1	MKR 600
50	K=K+1	MKR 610
60	KA=KA+NROW	MKR 620
	RETURN	MKR 630
	END	MKR 640

```

SUBROUTINE MMULT
C      VERSION 5.00      MMULT      5/15/70          MMU   10
C      SUBROUTINE MMULT    10/ 4/67          MMU   20
C      *
C      SUBROUTINE TO MULTIPLY MATRICES        MMU   30
C          GENERAL FORMS OF MMULT            MMU   40
C          MMULT A(,) N,K, BY B(,) K,M AND STORE IN C(,)  MMU   50
C      *
C      COMMON /SCRAT/ NS,NS2,A(13500)          MMU   60
C      COMMON /BLOCRC/ NRC,RC(12600)          MMU   70
C      COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NMMU 140
IARGS,VWXYZ(8),NERROR          MMU   110
DIMENSION ARGS(100)          MMU   120
EQUIVALENCE (ARGS(1),RC(12501))  MMU   130
DOUBLE PRECISION X,SUM        MMU   140
DIMENSION X(1)                MMU   150
EQUIVALENCE (X,A)            MMU   160
C      *
C      CHECK TO SEE IF WE HAVE CORRECT NUMBER OF ARGUMENTS  MMU   170
C      *
10     IF(NARGS.NE.10) CALL ERROR(10)          MMU   180
C      *
C      CHECK TO SEE IF ALL ARGUMENTS ARE INTEGERS        MMU   190
C      *
C      J=NARGS          MMU   200
CALL CKIND (J)                MMU   210
IF (J.EQ.0) GO TO 20          MMU   220
CALL ERROR (3)                MMU   230
C      *
C      CHECK TO SEE IF DIMENSIONS ARE CORRECT        MMU   240
C      *
20     IF(IARGS(4).NE.IARGS(7)) CALL ERROR (26)  MMU   250
C      *
C      CHECK TO SEE IF ARGUMENTS ARE OUT OF RANGE       MMU   260
C      FIND COLUMN ADDRESSES          MMU   270
C      *
70     IARGS(12)=IARGS(NARGS-2)          MMU   280
80     IARGS(11)=IARGS(3)                MMU   290
100    J=3          MMU   300
CALL MTXCHK (J)                MMU   310
IF (J-1) 130,110,120          MMU   320
110    CALL ERROR (3)                MMU   330
RETURN          MMU   340
120    CALL ERROR (17)                MMU   350
RETURN          MMU   360
C      *
C      CHECK FOR PREVIOUS ERRORS        MMU   370
C      *
130    IF (NERROR.NE.0) RETURN          MMU   380
IROWA=IARGS(3)                MMU   390
ICOLA=IARGS(4)                MMU   400
ICOLB=IARGS(8)                MMU   410
C      BEGIN MULTIPLICATION        MMU   420
C      *
ISP=1          MMU   430
IBP=IARGS(5)                MMU   440
DO 160 ICB=1,ICOLB          MMU   450
IAP=IARGS(1)                MMU   460
DO 150 IRA=1,IROWA          MMU   470

```

```

IS=NS2                               MMU 840
IA=IAP                               MMU 850
IB=IBP                               MMU 860
DO 140 J=1,ICOLA                     MMU 870
X(IS)=RC(IA)*RC(IB)                  MMU 880
IS=IS-1                             MMU 890
IA=IA+NROW                           MMU 900
IB=IB+1                            MMU 910
140 CONTINUE                         MMU 920
C *
C CALL ROUTINE TO SORT PRODUCTS AND SUM
C *
CALL SORTSM (ICOLA,SUM)              MMU 940
A(ISP)=SUM                           MMU 950
ISP=ISP+1                           MMU 960
MMU 970
MMU 980
150 IAP=IAP+1                         MMU 990
160 IBP=IBP+NROW                      MMU1000
C *
C STORE MATRIX PRODUCT
C *
IS=1                                MMU1040
IC=IARGS(9)                          MMU1050
DO 180 J=1,ICOLB                     MMU1060
DO 170 I=1,IROWA                       MMU1070
RC(IC)=A(IS)                          MMU1080
IS=IS+1                            MMU1090
IC=IC+1                            MMU1100
170 CONTINUE                         MMU1110
180 IC=IC+NROW-IROWA                  MMU1120
RETURN
END
MMU1130
MMU1140

```

```

SUBROUTINE MOP                                MOP 10
C      VERSION 5.00      MOP      5/15/70      MOP 20
C      SUBROUTINE TO DO MDEFINE,ADEFINE,MZERO,AZERO,MERASE,AERASE,MIDENT MOP 30
C      S. PEAVY FOR OMNITAB 1108      4/2/68      MOP 40
C      COMMANDS ARE AS FOLLOWS      MOP 50
C
C          MDEFINE  MATRIX IN R , C SIZE N X M TO EQUAL K      MOP 60
C          ADEFINE  ARRAY   IN R , C SIZE N X M TO EQUAL K      MOP 70
C          MZERO    MATRIX IN R , C SIZE N X M      MOP 80
C          AZERO    ARRAY   IN R , C SIZE N X M      MOP 90
C          MERASE   MATRIX IN R , C SIZE N X M      MOP 100
C          AERASE   ARRAY   IN R , C SIZE N X M      MOP 110
C          MIDENT   MATRIX IN R , C SIZE N X N      MOP 120
C          MDIAGO   MATRIX IN R , C SIZE N X M EQUAL TO E ON DIAGONAL MOP 130
C
C          L2=1  MDEFINE,ADEFINE      MOP 140
C          L2=2  MZERO,AZERO,MERASE,AERASE      MOP 150
C          L2=3  MIDENT      MOP 160
C          L2=4  MDIAGONAL      MOP 170
C
COMMON /BLOCRC/ NRC,RC(12600)      MOP 340
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NMOP 350
1ARGS,VWXYZ(8),NERROR      MOP 360
DIMENSION ARGS(100)      MOP 370
EQUIVALENCE (ARGS(1),RC(12501))      MOP 380
COMMON /SCRAT/ NS,NS2,A(13500)      MOP 390
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG      MOP 400
DATA ONE/1.0/,ZERO/0.0/      MOP 410
GO TO (10,50,60,80), L2      MOP 420
10 IF(NARGS.NE.5) CALL ERROR (10)      MOP 430
IF (KIND(NARGS).NE.1) CALL ERROR (3)      MOP 440
IF (NARGS.EQ.4) IARGS(4)=IARGS(3)      MOP 450
CONST=ARGS(NARGS)      MOP 460
CONSTA=ARGS(NARGS)      MOP 470
J=NARGS-1      MOP 480
20 CALL CKIND (J)      MOP 490
IF (J.NE.0) CALL ERROR (3)      MOP 500
J=1      MOP 510
CALL MTXCHK (J)      MOP 520
IF (J.NE.0) CALL ERROR (17)      MOP 530
IF (NERROR.NE.0) RETURN      MOP 540
JB=IARGS(1)      MOP 550
N=IARGS(3)      MOP 560
K=IARGS(4)      MOP 570
JA=JB      MOP 580
IF (L2.EQ.4) GO TO 90      MOP 590
DO 40 KA=1,K      MOP 600
JC=JB      MOP 610
DO 30 NA=1,N      MOP 620
RC(JC)=CONST      MOP 630
30 JC=JC+1      MOP 640
IF (KA.GT.N) GO TO 40      MOP 650
RC(JA)=CONSTA      MOP 660
JA=JA+NROW+1      MOP 670
40 JB=JB+NROW      MOP 680
RETURN      MOP 690
50 IF(NARGS.NE.4) CALL ERROR (10)      MOP 700
CONST=ZERO      MOP 710
CONSTA=ZERO      MOP 720

```

J=NARGS	MOP 730
IF (NARGS.EQ.4) GO TO 20	MOP 740
IARGS(4)=IARGS(3)	MOP 750
J=NARGS-1	MOP 760
GO TO 20	MOP 770
60 IF(NARGS.NE.4) CALL ERROR (10)	MOP 780
CONST=ZERO	MOP 790
CONSTA=ONE	MOP 800
J=NARGS	MOP 810
GO TO 20	MOP 820
80 J=NARGS-1	MOP 910
IF (NARGS.NE.5) CALL ERROR(10)	MOP 920
GO TO 20	MOP 930
90 IF (KIND(NARGS).EQ.0) GO TO 110	MOP 970
DO 100 NA=1,N	MOP 980
RC(JB)=ARGS(NARGS)	MOP 990
100 JB=JB+1+NROW	MOP1000
RETURN	MOP1010
110 KIND(5)=0	MOP1020
CALL ADDRESS (5,M)	MOP1030
IF (M.GT.0) GO TO 120	MOP1040
CALL ERROR (11)	MOP1050
RETURN	MOP1060
120 DO 130 NA=1,N	MOP1070
A(NA)=RC(M)	MOP1080
130 M=M+1	MOP1090
DO 140 NA=1,N	MOP1100
RC(JB)=A(NA)	MOP1110
140 JB=JB+1+NROW	MOP1120
RETURN	MOP1130
END	MOP1140

```

SUBROUTINE MOVE          MOV   10
C   VERSION 5.00      MOVE      5/15/70      MOV   20
COMMON /BLOCRC/ NRC,RC(12600)      MOV   30
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NMOV 40
IARGS,VWXYZ(8),NERROR      MOV   50
DIMENSION ARGS(100)      MOV   60
EQUIVALENCE (ARGS(1),RC(12501))      MOV   70
C   THIS ROUTINE IS ALSO CALLED BLOCKTRANSFER      MOV   80
IF (NARGS.EQ.6) GO TO 50      MOV   90
K=10      MOV 100
10  CALL ERROR (K)      MOV 110
20  RETURN      MOV 120
30  K=20      MOV 130
   GO TO 10      MOV 140
40  K=11      MOV 150
   GO TO 10      MOV 160
50  IARGS(9)=IARGS(1)+IARGS(3)-1      MOV 170
   IARGS(13)=IARGS(5)+IARGS(3)-1      MOV 180
   IF (KIND(1)+KIND(3)+KIND(4)+KIND(5).NE.0) GO TO 30      MOV 190
   IF (IARGS(1).GT.0.AND.IARGS(3).GT.0.AND.IARGS(5).GT.0.AND.IARGS(9).LE.NROW.AND.IARGS(13).LE.NROW) GO TO 60      MOV 200
   K=16      MOV 210
   GO TO 10      MOV 220
60  IARGS(10)=IARGS(2)+IARGS(4)-1      MOV 240
   KIND(10)=0      MOV 250
   IARGS(14)=IARGS(6)+IARGS(4)-1      MOV 260
   KIND(14)=0      MOV 270
   DO 70 I=2,14,4      MOV 280
   CALL ADDRESS (I,IARGS(I))      MOV 290
   IF (IARGS(I)) 30,40,70      MOV 300
70  IARGS(I)=IARGS(I)-1      MOV 310
C
C   IF MOVE IS UP, IR = -1, IF DOWN, IR = +1      MOV 320
C   IF MOVE IS LEFT, IC = -1, IF RIGHT, IC = +1      MOV 330
C   DIRECTION OF MOVE IS SUCH THAT THE TWO AREAS CAN BE OVERLAPPING      MOV 340
C   AND IT WILL BE DONE PROPERLY.      MOV 350
C
IR=ISIGN(1,IARGS(5)-IARGS(1))      MOV 360
IC=ISIGN(1,IARGS(6)-IARGS(2))      MOV 370
MM=IARGS(4*IR+5)+IARGS(4*IC+6)      MOV 380
NN=IARGS(4*IR+9)+IARGS(4*IC+10)      MOV 390
IC=IC*NROW      MOV 400
MMM=IARGS(3)      MOV 410
NNN=IARGS(4)      MOV 420
DO 90 J=1,NNN      MOV 430
M=MM      MOV 440
N=NN      MOV 450
DO 80 I=1,MMM      MOV 460
RC(N)=RC(M)      MOV 470
M=M-IR      MOV 480
80  N=N-IR      MOV 490
MM=MM-IC      MOV 500
90  NN=NN-IC      MOV 510
   GO TO 20      MOV 520
END      MOV 530
                                MOV 540
                                MOV 550

```

```

C SUBROUTINE MPROP                               MPR 10
C VERSION 5.00 MPROP 5/15/70                      MPR 20
C WRITTEN FOR OMNITAB BY S PEAVY 5/7/68          MPR 30
C COMMAND IS AS FOLLOWS, X=A OR M DEPENDING APROP OR MPROP IS REQUIRMPR 40
C   I XPROP OF MATRIX (,,++) NO OF ROWS ,,, NO OF COL ++
C     INFORMATION PRINTED AND NO STORAGE           MPR 50
C                                         MPR 60
C                                         MPR 70
C II XPROP MATRIX (,,++) SIZE ,, BY ++ PROPERTIES STORED ++
C   PROPERTIES PRINTED AND STORED                 MPR 80
C                                         MPR 90
C                                         MPR 100
C III XPROP MATRIX (,,++) R=,, C=,, PROP ++ COL NORMS (,,++)
C   SAME AS II PLUS STORAGE OF COLUMN NORMS      MPR 110
C                                         MPR 120
C                                         MPR 130
C IV XPROP (,,++) R=,, C=,, PROP ++ COL NORMS (,,++) ROW NORMS (mpr 140
C   SAME AS III PLUS STORAGE OF ROW NORMS, ALSO (R+1,++) OF NORM MPR 150
C   AVERAGES WILL CONTAIN GRAND AVERAGE, IF X=A.    MPR 160
C                                         MPR 170
C V XPROP (,,++) R=,, C=,, COL NORMS (,,++)
C   SAME AS III EXCEPT PROPERTIES WILL NOT BE STORED MPR 180
C                                         MPR 190
C                                         MPR 200
C VI XPROP (,,++) R=,, C=,, COL NORMS (,,++) ROW NORMS (,,++)
C   SAME AS IV EXCEPT PROPERTIES WILL NOT BE STORED MPR 210
C                                         MPR 220
C                                         MPR 230
C VII SXPROP                                     MPR 240
C   IF COMMANDS I-VI ARE PREFACED WITH AN S PRINTOUT
C   OF PROPERTIES WILL BE SUPPRESSED              MPR 250
C                                         MPR 260
C                                         MPR 270
C L2 OPTIONS:
C   L2= 1 MPROP: L2=3 SMPROP                     MPR 280
C   L2= 2 APROP: L2=4 SAPROP                     MPR 290
C                                         MPR 300
C                                         MPR 310
C COMMON/HEADER/NOCARD(80),ITLE(60,6),LNCNT,IPRINT,NPAGE,IPUNCH MPR 320
C COMMON /SCRAT/ NS,NS2,A(13500)                  MPR 340
C COMMON /BLOCRC/ NRC,RC(12600)                  MPR 350
C COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NMPR 360
1ARGS,VWXYZ(8),NERROR                           MPR 370
DIMENSION ARGS(100)                             MPR 380
EQUIVALENCE (ARGS(1),RC(12501))                MPR 390
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFGLG          MPR 400
DIMENSION IRSLT(6), ERR(3), IPROP(5), IRSLLT(11), IRSLLTA(2,5) MPR 410
DATA NO/3H NO/,IYES/3HYES/,IBLK/3H   /,LOWRA/3H LO/,LOWRB/3HWER/,IMPR 420
1PPRA/3H UP/,IPPRB/3HPER/,IANDA/3H AN/,IANDB/3HD  /,IRWA,IRWB/3H MPR 430
2,3HROW/,ICLMA,ICLMB/3HCOL,3HUMN/,IBTHH,IBTHHA/3H  ,3H T/,IBTHA,IMPR 440
3BTHB/3HWO-,3HWAY/,NOA,NOAB/3H   ,3H NO/       MPR 450
IF(L2.LE.2.OR.NARGS.NE.4) GO TO 5             MPR 455
CALL ERROR (233)                                MPR 460
RETURN                                           MPR 465
5 IF (NARGS.LT.4.OR.NARGS.GT.9) CALL ERROR (10) MPR 467
LOC=IARGS(1)                                    MPR 470
J=NARGS                                         MPR 480
CALL CKIND (J)                                  MPR 490
IF (J.NE.0) CALL ERROR (3)                      MPR 500
K=5                                              MPR 510
J=1                                              MPR 520
IF (NARGS-5) 60,30,10                           MPR 530
10 IF (NARGS.EQ.6.OR.NARGS.EQ.8) GO TO 40        MPR 540
IS=IARGS(5)                                    MPR 550
J=2                                              MPR 560
K=9                                              MPR 570

```

```

IF (NARGS.EQ.7) GO TO 20
IARGS(11)=IARGS(3)
IARGS(12)=1
IARGS(10)=IARGS(9)
IARGS(9)=IARGS(8)
K=13
J=3
IF (L2.GT.2) IARGS(11)=IARGS(11)+1
20 IARGS(5)=IARGS(6)
IARGS(6)=IARGS(7)
IARGS(7)=1
IARGS(8)=IARGS(4)
IARGS(K)=IS
30 CALL ADRESS (K,KPROP)
IF (KPROP.LE.0) CALL ERROR (11)
GO TO 60
40 J=2
IF (NARGS.EQ.6) GO TO 50
IARGS(9)=IARGS(7)
IARGS(10)=IARGS(8)
IARGS(11)=IARGS(3)
IARGS(12)=1
J=3
IF (L2.GT.2) IARGS(11)=IARGS(11)+1
50 IARGS(7)=1
IARGS(8)=IARGS(4)
60 CALL MTXCHK (J)
KARGS=K
IF (J.NE.0) CALL ERROR (17)
IF (NERROR.NE.0) RETURN
K=IARGS(1)
IF (L2.EQ.2.OR.L2.EQ.4) GO TO 110
C COMMAND IS MPROP
C IS MATRIX A SQUARE ONE
IF (IARGS(3).NE.IARGS(4)) GO TO 90
C YES
CALL INVCHK (RC(K),NROW,IARGS(3),A(100),IARGS(3)+1,A,1,ERR,IND)
KA=K
M=100
L=IARGS(3)
DO 80 I=1,L
KB=KA
DO 70 J=1,L
A(M)=RC(KB)
KB=KB+1
M=M+1
70 KA=KA+NROW
80 CALL DETRNK (A(100),L,L,DET,RANK)
CALL PVTRI (RC(K),NROW,IARGS(3),INDU,INDB)
CALL PROCHK (RC(K),NROW,IARGS(3),IARGS(4),IPROP,A(1),NS2)
A(30)=0.
IF (INDU.EQ.0) A(30)=A(30)+1.0
IF (INDB.EQ.0) A(30)=A(30)+2.0
A(19)=DET
A(20)=RANK
A(21)=ERR(1)
A(22)=ERR(2)
A(23)=ERR(3)
A(24)=0.

```

MPR 580  
 MPR 590  
 MPR 600  
 MPR 610  
 MPR 620  
 MPR 630  
 MPR 640  
 MPR 650  
 MPR 660  
 MPR 670  
 MPR 680  
 MPR 690  
 MPR 700  
 MPR 710  
 MPR 720  
 MPR 730  
 MPR 740  
 MPR 750  
 MPR 760  
 MPR 770  
 MPR 780  
 MPR 790  
 MPR 800  
 MPR 810  
 MPR 820  
 MPR 830  
 MPR 840  
 MPR 850  
 MPR 860  
 MPR 870  
 MPR 880  
 MPR 890  
 MPR 900  
 MPR 910  
 MPR 920  
 MPR 930  
 MPR 940  
 MPR 950  
 MPR 960  
 MPR 970  
 MPR 980  
 MPR 990  
 MPR1000  
 MPR1010  
 MPR1020  
 MPR1030  
 MPR1040  
 MPR1050  
 MPR1060  
 MPR1070  
 MPR1080  
 MPR1090  
 MPR1100  
 MPR1110  
 MPR1120  
 MPR1130  
 MPR1140  
 MPR1150  
 MPR1160

```

IF (IPROP(3).EQ.0) A(24)=1.0          MPR1170
IF (IPROP(3).EQ.1) A(24)=2.0          MPR1180
A(25)=0.                            MPR1190
IF (IPROP(2).LT.2) A(25)=IPROP(2)+1   MPR1200
A(26)=0.                            MPR1210
IF (IPROP(2).GT.2) A(26)=IPROP(2)-2   MPR1220
A(27)=0.                            MPR1230
IF (IPROP(1).EQ.0) A(27)=1.           MPR1240
A(28)=0.0                           MPR1250
IF (IPROP(4).LT.2) A(28)=IPROP(4)+1   MPR1260
A(29)=0.                            MPR1270
IF (IPROP(5).LT.2) A(29)=IPROP(5)+1   MPR1280
GO TO 110                           MPR1290
90  CALL ORTHRV (RC(K),NROW,IARGS(3),IARGS(4),IPROP(4),A(1),NS2,A(1)) MPR1300
DO 100 I=19,31                      MPR1310
100 A(I)=0.0                          MPR1320
110 CALL RCSUM (RC(K),NROW,IARGS(3),IARGS(4),A(101))                  MPR1330
L=IARGS(3)                           MPR1340
IF (L.GT.IARGS(4)) L=IARGS(4)        MPR1350
ITRACE=L                           MPR1360
IPOS=0                             MPR1370
IZERO=0                           MPR1380
INEG=0                            MPR1390
TRACE=0.0                          MPR1400
AMX=RC(K)                          MPR1410
AMN=AMX                           MPR1420
LA=IARGS(3)+IARGS(4)                MPR1430
AVG=A(LA+102)/FLOAT(IARGS(3)*IARGS(4)) MPR1440
ABSMX=ABS(AMX)                     MPR1450
ABSMN=ABS(AMN)                     MPR1460
ABSMNZ=ABSMN                      MPR1470
SSQ=0.0                            MPR1480
SRSQ=0.0                           MPR1490
SCSQ=0.0                           MPR1500
IF (ABSMNZ.EQ.0.0) ABSMNZ=1.E35     MPR1510
KA=K                               MPR1520
DO 120 I=1,L                       MPR1530
TRACE=TRACE+RC(KA)                 MPR1540
120 KA=KA+NROW+1                   MPR1550
IN=IARGS(3)                         MPR1560
JK=IARGS(4)                         MPR1570
KA=K                               MPR1580
FIN=IN                             MPR1590
FJK=JK                            MPR1600
DO 200 J=1,JK                     MPR1610
KB=KA                             MPR1620
DO 190 I=1,IN                     MPR1630
KC=IARGS(4)+I                      MPR1640
SSQ=SSQ+(RC(KB)-AVG)**2            MPR1650
SCSQ=SCSQ+(RC(KB)-A(J+100)/FIN)**2 MPR1660
SRSQ=SRSQ+(RC(KB)-A(KC+100)/FJK)**2 MPR1670
IF (RC(KB)) 130,140,150           MPR1680
130 INEG=INEG+1                   MPR1690
GO TO 160                           MPR1700
140 IZERO=IZERO+1                 MPR1710
GO TO 160                           MPR1720
150 IPOS=IPOS+1                   MPR1730
160 IF (AMX.GT.RC(KB)) GO TO 170   MPR1740
AMX=RC(KB)                         MPR1750

```

	GO TO 180	MPR1760
170	IF (AMN.GT.RC(KB)) AMN=RC(KB)	MPR1770
180	RCAB=ABS(RC(KB))	MPR1780
	IF (ABSMX.LT.RCAB) ABSMX=RCAB	MPR1790
	IF (ABSMN.GT.RCAB) ABSMN=RCAB	MPR1800
	IF (ABSMNZ.GT.RCAB) IF (RCAB) 190,190,185	MPR1805
	GO TO 190	MPR1810
185	ABSMNZ=RCAB	MPR1820
190	KB=KB+1	MPR1830
200	KA=KA+NROW	MPR1840
	IF (L2.EQ.2.OR.L2.EQ.4.OR.IARGS(3).NE.IARGS(4)) GO TO 250	MPR1850
	ISTOCR=0	MPR1860
	ISTCHC=0	MPR1870
	IF (AMN.LT.0.0) GO TO 240	MPR1880
	DO 210 J=1,JK	MPR1890
	IF (A(J+100).EQ.1.0) GO TO 210	MPR1900
	GO TO 220	MPR1910
210	CONTINUE	MPR1920
	ISTCHC=2	MPR1930
220	DO 230 I=1,IN	MPR1940
	M=I+JK	MPR1950
	IF (A(M+100).EQ.1.0) GO TO 230	MPR1960
	GO TO 240	MPR1970
230	CONTINUE	MPR1980
	ISTOCR=1	MPR1990
240	A(31)=ISTCHC+ISTOCR	MPR2000
250	A(1)=TRACE	MPR2010
	TR2=0.0	MPR2020
	KB=K	MPR2030
	JKK=JK	MPR2040
	IF (JK.GT.IN) JKK=IN	MPR2050
	DO 260 J=2,JKK	MPR2060
	KA=K	MPR2070
	KB=KB+NROW+1	MPR2080
	KC=K+J-1	MPR2090
	KD=K+(J-1)*NROW	MPR2100
	II=J-1	MPR2110
	DO 260 I=1,II	MPR2120
	TR2=TR2+(RC(KA)*RC(KB)-RC(KC)*RC(KD))	MPR2130
	KA=KA+NROW+1	MPR2140
	KC=KC+NROW	MPR2150
260	KD=KD+1	MPR2160
	A(2)=TR2	MPR2170
	A(3)=AMX	MPR2180
	A(4)=AMN	MPR2190
	A(5)=ABSMX	MPR2200
	A(6)=ABSMN	MPR2210
	A(7)=ABSMNZ	MPR2220
	A(8)=IPOS	MPR2230
	A(9)=IZERO	MPR2240
	A(10)=INEG	MPR2250
	A(11)=A(LA+101)	MPR2260
	A(12)=AVG	MPR2270
	A(13)=A(LA+103)	MPR2280
	A(14)=SSQ	MPR2290
	A(15)=SRSQ	MPR2300
	A(16)=SCSQ	MPR2310
	A(17)=A(LA+104)	MPR2320
	A(18)=A(17)/FLOAT(IARGS(3)*IARGS(4))	MPR2330

```

IF (L2.GE.3) GO TO 570                               MPR2340
CALL PAGE (4)                                         MPR2350
IF (L2.NE.1) GO TO 370                               MPR2360
WRITE (IPRINT,640) IARGS(3),IARGS(4),LOC,IARGS(2)   MPR2370
IF (MOD(NARGS,2).EQ.0) GO TO 270                   MPR2380
WRITE (IPRINT,650) IARGS(KARGS)                      MPR2390
270 WRITE (IPRINT,660) ITRACE,(A(I),I=1,7),IPOS,IZERO,INEG,(A(I),I=11,116) MPR2400
      MPR2410
      WRITE (IPRINT,670) A(17),A(18)                  MPR2420
      WRITE (IPRINT,680)                                         MPR2430
      IF (IARGS(3).NE.IARGS(4)) GO TO 390           MPR2440
      IRANK=A(20)+.5E-5                            MPR2450
      WRITE (IPRINT,690) A(19),IRANK,(A(I),I=21,23)  MPR2460
      DO 280 I=1,6                                  MPR2470
      IRSLT(I)=IYES                                MPR2480
      IRSLT(I)=A(I+23)                             MPR2490
      IF (A(I+23).EQ.0.) IRSLT(I)=NO              MPR2500
280 CONTINUE                                         MPR2510
C SET IRSLT(I),I=1,6 FOR YES OR NO. ALSO A(I),I=24,29 MPR2520
DO 290 I=1,5                                         MPR2530
      IRSLT(I,I)=IBLK                            MPR2540
290 IRSLT(I,2,I)=IBLK                           MPR2550
      IRSLT(I,3)=NOA                            MPR2560
      IRSLT(2,3)=NOAB                           MPR2570
      IF (INDU.NE.0.AND.INDB.NE.0) GO TO 320       MPR2580
      IF (INDU.NE.0) GO TO 310                   MPR2590
      IF (INDB.EQ.0) GO TO 300                   MPR2600
      IRSLT(1,1)=IPPRA                          MPR2610
      IRSLT(2,1)=IPPRB                          MPR2620
      GO TO 320                                MPR2630
300 IRSLT(1,1)=IPPRA                          MPR2640
      IRSLT(2,1)=IPPRB                          MPR2650
      IRSLT(1,2)=IANDA                          MPR2660
      IRSLT(2,2)=IANDB                          MPR2670
310 IRSLT(1,3)=LOWRA                          MPR2680
      IRSLT(2,3)=LOWRB                          MPR2690
320 IRSLT(1,5)=NOA                            MPR2700
      IRSLT(2,5)=NOAB                           MPR2710
      IF (ISTOCR+ISTCHC.EQ.3) GO TO 340         MPR2720
      IF (ISTOCR.EQ.0) GO TO 330               MPR2730
      IRSLT(1,5)=IRWA                           MPR2740
      IRSLT(2,5)=IRWB                           MPR2750
      GO TO 350                                MPR2760
330 IF (ISTCHC.EQ.0) GO TO 350               MPR2770
      IRSLT(1,5)=ICLMA                          MPR2780
      IRSLT(2,5)=ICLMB                          MPR2790
      GO TO 350                                MPR2800
340 IRSLT(1,4)=IBTHH                          MPR2810
      IRSLT(2,4)=IBTHHA                         MPR2820
      IRSLT(1,5)=IBTHA                          MPR2830
      IRSLT(2,5)=IBTHB                          MPR2840
350 IRSLT(7)=A(30)                           MPR2850
      IRSLT(8)=A(31)                           MPR2860
      WRITE (IPRINT,700) (IRSLT(I),IRSLTP(I),I=1,6) MPR2870
      WRITE (IPRINT,710) ((IRSLTA(I,J),I=1,2),J=1,3),IRSLTP(7),((IRSLTA(1,J),I=1,2),J=4,5),IRSLTP(8) MPR2880
      DO 360 I=1,2                                MPR2890
      MPR2900
360 WRITE (IPRINT,720)                         MPR2910
      WRITE (IPRINT,730)                         MPR2920

```

	GO TO 570	MPR2930
C	APROP PRINT OUT	MPR2940
370	WRITE (IPRINT,740) IARGS(3),IARGS(4),LOC,IARGS(2)	MPR2950
	IF (MOD(NARGS,2).EQ.0) GO TO 380	MPR2960
	WRITE (IPRINT,650) IARGS(KARGS)	MPR2970
380	WRITE (IPRINT,660) ITRACE,(A(I),I=1,7),IPOS,IZERO,INEG,(A(I),I=11,MPR2980	
	116)	MPR2990
	WRITE (IPRINT,670) A(17),A(18)	MPR3000
	GO TO 570	MPR3010
C	MPROP PRINT OUT FOR A NON-SQUARE MATRIX	MPR3020
390	DO 400 I=1,2	MPR3030
	IRSLTA(1,I)=NOA	MPR3040
400	IRSLTA(2,I)=NOAB	MPR3050
	IF (IPROP(4).EQ.2) GO TO 550	MPR3060
	IF (IABS(IPROP(4))-4) 410,480,480	MPR3070
410	DO 420 I=1,2	MPR3080
	IRSLTA(1,I)=IRWA	MPR3090
420	IRSLTA(2,I)=IRWB	MPR3100
	IF (IPROP(4)) 430,430,440	MPR3110
430	A(26)=2	MPR3120
	GO TO 450	MPR3130
440	A(26)=1	MPR3140
450	IF (IPROP(5)) 460,460,470	MPR3150
460	A(27)=2	MPR3160
	GO TO 550	MPR3170
470	A(27)=1	MPR3180
	GO TO 550	MPR3190
480	DO 490 I=1,2	MPR3200
	IRSLTA(1,I)=ICLMA	MPR3210
490	IRSLTA(2,I)=ICLMB	MPR3220
	IF (IPROP(4)) 500,500,510	MPR3230
500	A(26)=4	MPR3240
	GO TO 520	MPR3250
510	A(26)=3	MPR3260
520	IF (IPROP(5)) 530,530,540	MPR3270
530	A(27)=4	MPR3280
	GO TO 550	MPR3290
540	A(27)=3	MPR3300
550	IRSLTP(1)=A(26)	MPR3310
	IRSLTP(2)=A(27)	MPR3320
	WRITE (IPRINT,750) ((IRSLTA(J,I),J=1,2),IRSLTP(I),I=1,2)	MPR3330
	DO 560 I=1,22	MPR3340
560	WRITE (IPRINT,720)	MPR3350
	WRITE (IPRINT,760)	MPR3360
570	IF (NARGS.EQ.4) RETURN	MPR3370
	IF (MOD(NARGS,2).EQ.0) GO TO 610	MPR3380
	IP=31	MPR3390
	IF (IARGS(3).EQ.IARGS(4)) GO TO 590	MPR3400
	IP=27	MPR3410
	DO 580 I=19,25	MPR3420
580	A(I)=0.0	MPR3430
590	IF (MOD(L2,2).EQ.0) IP=18	MPR3440
	IF (NROW.LT.IP) IP=NROW	MPR3450
	DO 600 I=1,IP	MPR3460
	RC(KPROP)=A(I)	MPR3470
600	KPROP=KPROP+1	MPR3480
	IF (NARGS.EQ.5) RETURN	MPR3490
610	KA=IARGS(5)	MPR3500
	ANRMAX=IARGS(3)	MPR3510

```

DO 620 I=1,JK                               MPR3520
RC(KA)=A(I+100)/ANRMX                      MPR3530
620   KA=KA+NROW                            MPR3540
      IF (NARGS.LT.8) RETURN                  MPR3550
      KA=IARGS(9)                            MPR3560
      ANRMX=IARGS(4)                          MPR3570
      KB=JK+101                             MPR3580
      DO 630 I=1,IN                           MPR3590
      RC(KA)=A(KB)/ANRMX                     MPR3600
      KA=KA+1                                MPR3610
630   KB=KB+1                                MPR3620
      IF (L2.GT.2) RC(KA)=AVG                 MPR3630
      RETURN                                  MPR3640
C                                         MPR3650
640   FORMAT (1H0,39X,14HPROPERTIES OF ,I3,3H X ,I3,27H MATRIX STARTING MPR3660
1LOCATION (,I3,1H,,I3,1H))                   MPR3670
650   FORMAT (23X,3HCOL,I7)                   MPR3680
660   FORMAT (30X,7HGENERAL/23X,1HR/23X,9H1 TRACE (,I3,13H VALUES USED),MPR3690
17X,1PE15.6/23X,32H2 TRACE NO. 2           ,E15.6//23X,32H3MPR3700
2 MAXIMUM ELEMENT ,E15.6/23X,20H4 MINIMUM ELEMENT ,MPR3710
3 12X ,E15.6/23X,32H5 MAXIMUM ELEMENT IN ABS VALUE ,E15.6/23XMPR3720
4,32H6 MINIMUM ELEMENT IN ABS VALUE ,E15.6/23X,32H7 MIN NON-ZERO EMPR3730
5LEM IN ABS VAL ,E15.6//23X,32H8 NUMBER OF POSITIVE ELEMENTS ,10MPR3740
6X,I5/,23X,32H9 NUMBER OF ZERO ELEMENTS ,10X,I5/22X,33H10 NUMMPR3750
7BER OF NEGATIVE ELEMENTS ,10X,I5//22X,33H11 SUM OF TERMS MPR3760
8 ,E15.6/22X,33H12 AVERAGE ,E15.6/22XMPR3770
9X,33H13 SUM OF SQUARES ,E15.6/22X,33H14 SUM OF SQUAMPR3780
$RES ABOUT MEAN ,E15.6/22X,33H15 WITHIN ROWS SUM OF SQUARES MPR3790
$,E15.6/22X,33H16 WITHIN COLS SUM OF SQUARES ,E15.6) MPR3800
670   FORMAT (22X,33H17 SUM OF ABSOLUTE VALUES ,1PE15.6/22X,33H18MPR3810
1 AVERAGE OF ABSOLUTE VALUES ,E15.6) MPR3820
680   FORMAT (1H0,29X,8HSPECIFIC/) MPR3830
690   FORMAT (22X,33H19 DETERMINANT ,1PE15.6/22X,33H18MPR3840
1 RANK ,7X,I8/30X,5HNORMS/22X,33H21 SQ ROOMPR3850
2 T OF SUM OF B(I,J)**2 ,E15.1/22X,33H22 N*MAX(B(I,J)) MPR3860
3 ,E15.1/22X,33H23 MAX VAL OF ROW SUM ,E15.1) MPR3870
700   FORMAT (1H0,21X,32H24 NORMALITY ,13X,A3,2H*(,I1MPR3880
1,1H)/22X,33H25 SYMMETRY ,12X,A3,2H*(,I1,1H)/2MPR3890
22X,33H26 SKEW SYMMETRY ,12X,A3,2H*(,I1,1H)/22X,33HMPR3900
327 DIAGONALITY ,12X,A3,2H*(,I1,1H)/22X,33H28 ORTMPPR3910
4HOGONALITY: A,A=I ,12X,A3,2H*(,I1,1H)/22X,2H29,17X,19HA,AMPR3920
5=DIAGONAL MATRIX,7X,A3,2H*(,I1,1H)//) MPR3930
710   FORMAT(22X,13H30 TRIANGULAR,20X,3A3,A1,2A3,3H**(,I1,1H)/22X,33H31 MPR3940
1STOCHASTIC (R AND/OR C SUMS=1),3X,4A3,4H***(,I1,1H)) MPR3950
720   FORMAT (1H ) MPR3960
730   FORMAT (9X,79H* IF ANSWER IS YES, (R,C)=1 OR 2. (1, IF EXACT; 2, MPR3970
1IF TOLERANCE IS SATISFIED.)/11X,25HIF ANSWER IS NO, (R,C)=0.//8X,1MPR3980
20HTRIANGULAR/8X,69H** (R,C)=0, IF ANSWER IS NO; (R,C)=1, IF UPPER PMPR3990
3ART OF MATRIX IS ZERO;/11X,74H(R,C)=2, IF LOWER PART IS ZERO; (R,CMPR4000
4)=3, IF ALL OFF DIAGONAL ELEMENTS = 0.//7X,10HSTOCHASTIC/7X,75H***MPR4010
5 (R,C)=0, IF MATRIX IS NOT STOCHASTIC; (R,C)=1, IF SUM OF EACH ROWMPR4020
6 = 1; /11X,75H(R,C)=2, IF SUM OF EACH COLUMN=1; (R,C)=3, IF SUM OFMPR4030
7 EACH ROW AND COLUMN=1.) MPR4040
740   FORMAT (1H0,39X,14HPROPERTIES OF ,I3,3H X ,I3,26H ARRAY STARTING LMPR4050
1LOCATION (,I3,1H,,I3,1H)) MPR4060
750   FORMAT(22X,32H26 ORTHOGONALITY: A,A=I ,9X,2A3,2H*(,I1,1H)/MPR4070
1 22X,2H27,17X,19HA,A=DIAGONAL MATRIX,3X2A3,2H*(,I1,1H)) MPR4080
760   FORMAT (9X,86H* (R,C)=0, IF MATRIX IS NOT ORTHOGONAL; (R,C)=1 OR 2MPR4090
1 IF MATRIX IS ORTHOGONAL ROW WISE;/11X,97H(R,C)=3 OR 4, IF MATRIX MPR4100

```

2IS ORTHOGONAL COLUMN WISE. ( (R,C)=I, IF I=1 OR 3 ORTHOGONALITY IS MPR4110  
3 EXACT;/11X,50HFOR I=2 OR 4 RELATIVE WITHIN ERROR BOUND OF .1E.6))MPR4120  
END MPR4130

```

SUBROUTINE MRAISE                               MRA   10
C      VERSION 5.00      MRAISE      5/15/70      MRA   20
C      *      MRA   30
C      SUBROUTINE TO RAISE A MATRIX TO A POWER      9/13/67      MRA   40
C          GENERAL FORMS OF MRAISE                  MRA   50
C          MRAISE A(,) N,N TO M POWER AND STORE IN C(,)      MRA   60
C          M MAY BE INTEGER OR REAL                 MRA   80
C          IF M=0 C=IDENTITY MATRIX                MRA   90
C          IF M=1 C=A                           MRA  100
C          *
C          COMMON /BLOCRC/ NRC,RC(12600)           MRA  120
C          COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NMRA 130
C          IARGS,VWXYZ(8),NERROR                  MRA  140
C          DIMENSION ARGS(100)                     MRA  150
C          EQUIVALENCE (ARGS(1),RC(12501))        MRA  160
C          COMMON /SCRAT/ NS,NS2,A(13500)         MRA  170
C          DOUBLE PRECISION X,SUM                 MRA  180
C          DIMENSION X(1)                         MRA  190
C          EQUIVALENCE (X,A)                      MRA  200
C          *
C          CHECK NUMBER OF ARGUMENTS             MRA  210
C          *
C          IF(NARGS.NE.7) CALL ERROR (10)        MRA  220
C          *
C          CHECK TO SEE IF ALL ARGUMENTS ARE INTEGER      MRA  230
C          *
C          J=NARGS                                MRA  240
C          CALL CKIND (J)                          MRA  250
C          IF (J.EQ.0) GO TO 20                   MRA  260
C          IF (KIND(NARGS-2).NE.0) GO TO 10       MRA  270
C          CALL ERROR (3)                        MRA  280
C          GO TO 20                                MRA  290
C          10 IARGS(NARGS-2)=ARGS(NARGS-2)        MRA  300
C          *
C          CHECK TO SEE IF M (POWER) IS NEGATIVE      MRA  310
C          *
C          20 IF (IARGS(NARGS-2).LT.0) CALL ERROR (3)    MRA  320
C          *
C          CHECK TO SEE IF DIMENSIONS ARE CORRECT      MRA  330
C          *
C          IF (IARGS(3).NE.IARGS(4)) CALL ERROR (3)    MRA  340
C          *
C          CHECK TO SEE IF ARGUMENTS ARE OUT OF RANGE      MRA  350
C          *
C          30 NPOW=IARGS(NARGS-2)-1                  MRA  360
C          40 IARGS(5)=IARGS(NARGS-1)                MRA  370
C          IARGS(6)=IARGS(NARGS)                  MRA  380
C          IARGS(7)=IARGS(3)                      MRA  390
C          IARGS(8)=IARGS(4)                      MRA  400
C          J=2                                    MRA  410
C          CALL MTXCHK (J)                      MRA  420
C          IF (J-1) 70,50,60                    MRA  430
C          50 CALL ERROR (3)                      MRA  440
C          RETURN                                MRA  450
C          *
C          60 CALL ERROR (17)                      MRA  460
C          RETURN                                MRA  470
C          *
C          CHECK TO SEE IF PREVIOUS ERRORS          MRA  480
C          *

```

```

70  IF (NERROR.NE.0) RETURN                                MRA 640
    ISIZE=IARGS(3)                                         MRA 650
C   *                                                       MRA 660
C   BEGIN MULTIPLICATION                                 MRA 670
C   *                                                       MRA 680
C   *                                                       MRA 690
C   MOVE ORIGINAL MATRIX TO SCRATCH AREA (COLUMNWISE) MRA 700
C   *                                                       MRA 710
    IF (NPOW) 80,90,110                                     MRA 720
80  IEXT=1                                                 MRA 730
    GO TO 100                                              MRA 740
90  IEXT=2                                                 MRA 750
100 ISAV=IARGS(5)                                         MRA 760
    GO TO 120                                              MRA 770
110 IEXT=3                                                 MRA 780
120 IP=IARGS(1)                                           MRA 790
    IC=1                                                   MRA 800
    DO 220 J=1,ISIZE                                       MRA 810
    DO 180 I=1,ISIZE                                       MRA 820
    GO TO (130,160,170), IEXT                            MRA 830
130 IF (I.EQ.J) GO TO 140                                MRA 840
    RC(ISAV)=0.0                                           MRA 850
    GO TO 150                                              MRA 860
140 RC(ISAV)=1.0                                           MRA 870
150 ISAV=ISAV+1                                           MRA 880
    GO TO 180                                              MRA 890
160 RC(ISAV)=RC(IP)                                       MRA 900
    IP=IP+1                                               MRA 910
    GO TO 150                                              MRA 920
170 A(IC)=RC(IP)                                         MRA 930
    IC=IC+1                                               MRA 940
    IP=IP+1                                               MRA 950
180 CONTINUE                                              MRA 960
    GO TO (190,200,210), IEXT                            MRA 970
190 ISAV=ISAV+NROW-ISIZE                                 MRA 980
    GO TO 220                                              MRA 990
200 ISAV=ISAV+NROW-ISIZE                                 MRA1000
210 IP=IP+NROW-ISIZE                                   MRA1010
220 CONTINUE                                              MRA1020
    IF (IEXT.LE.2) RETURN                                MRA1030
    IXP=NS-ISIZE*2                                         MRA1040
    DO 280 K=1,NPOW                                       MRA1050
    ISAVP=IARGS(5)                                         MRA1060
    IMP=NS2                                               MRA1070
    IF (K.GT.1) GO TO 230                                MRA1080
    IRP=IARGS(1)                                           MRA1090
    GO TO 240                                              MRA1100
230 IRP=IARGS(5)                                         MRA1110
240 DO 280 I=1,ISIZE                                     MRA1120
    ISAV=ISAVP                                           MRA1130
    IC=1                                                   MRA1140
    IR=IRP                                               MRA1150
    IX=IXP                                               MRA1160
C   *                                                       MRA1170
C   SAVE ROW OF MATRIX                                  MRA1180
C   *                                                       MRA1190
    DO 250 J=1,ISIZE                                     MRA1200
    A(IX)=RC(IR)                                         MRA1210
    IX=IX-1                                              MRA1220

```

	IR=IR+NROW	MRA1230
250	CONTINUE	MRA1240
	DO 270 J=1,ISIZE	MRA1250
	IX=IXP	MRA1260
	IM=IMP	MRA1270
	DO 260 JP=1,ISIZE	MRA1280
	X(IM)=A(IX)*A(IC)	MRA1290
	IM=IM-1	MRA1300
	IX=IX-1	MRA1310
	IC=IC+1	MRA1320
260	CONTINUE	MRA1330
	CALL SORTSM (ISIZE,SUM)	MRA1340
	RC(ISAV)=SUM	MRA1350
	ISAV=ISAV+NROW	MRA1360
270	CONTINUE	MRA1370
	ISAVP=ISAVP+1	MRA1380
	IRP=IRP+1	MRA1390
280	CONTINUE	MRA1400
	RETURN	MRA1410
	END	MRA1420

```

SUBROUTINE MSCROW                               MSC   10
C VERSION 5.00      MSCROW      5/15/70        MSC   20
COMMON /BLOCRC/ NRC,RC(12600)                  MSC   30
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NMSC 40
IARGS,VWXYZ(8),NERROR                         MSC   50
DIMENSION ARGS(100)                           MSC   60
EQUIVALENCE (ARGS(1),RC(12501))              MSC   70
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG          MSC   80
C   SUBROUTINE BY CARLA MESSINA 221.04        JUNE 1967    MSC   90
C   TYPE 1 IS      PARSUM OF COL ++ , STORE IN COL ++    MSC 100
C   TYPE 2 IS      PARPRODUCT OF COL ++, STORE IN COL ++  MSC 110
C   TYPE 3 IS      ROOT MEAN SQUARE      RMS OF COL ++, STORE IN COL  MSC 120
C   TYPE 4 IS      AVERAGE OF COL ++, STORE IN COL ++     (DOWN TO NMSC 130
C   TYPE 5 IS      SUM COL ++, STORE IN COL ++     (DOWN TO NRMAX)  MSC 140
C           SUM COL ++ FROM ROW ,, TO ROW ,, STORE IN COL ++  MSC 150
C           SUM COL ++ FROM ROWS NUMBERED ,, ,,, ,,, ETC STORE MSC 160
C THE THREE TYPES OF SUM ARE IDENTIFIED BY THE NO. OF NARGS =2,3 ANDMSC 170
ELEM=0.0                                         MSC 180
IF (NARGS-2) 10,40,40                           MSC 190
10 K=10                                           MSC 200
20 CALL ERROR (K)                             MSC 210
30 RETURN                                         MSC 220
40 CALL ADRESS (1,J1)                          MSC 230
IF (J1) 50,50,60                                MSC 240
50 K=3                                           MSC 250
GO TO 20                                         MSC 260
60 CALL ADRESS (NARGS,J2)                      MSC 270
IF (J2) 50,50,70                                MSC 280
70 IF (NARGS-3) 210,80,80                        MSC 290
80 IF (L2-5) 10,90,10                           MSC 300
90 NARG1=NARGS-1                                MSC 310
DO 110 I=2,NARG1                                MSC 320
IF (KIND(I).NE.0) GO TO 130                     MSC 330
IF (IARGS(I)) 130,130,100                       MSC 340
100 IF (IARGS(I)-NROW) 110,110,130             MSC 350
110 CONTINUE                                       MSC 360
IF (NERROR.NE.0) GO TO 30                      MSC 370
IF (NARGS-4) 120,120,180                        MSC 380
C   SUM FROM ROW ,, TO ROW ,,                   MSC 390
C
120 IF (IARGS(2)-IARGS(3)) 140,140,130         MSC 420
130 I=IARGS(2)                                    MSC 430
IARGS(2)=IARGS(3)                                MSC 440
IARGS(3)=I                                         MSC 450
140 IF (NRMAX) 150,150,160                      MSC 460
150 K=9                                         MSC 470
GO TO 20                                         MSC 480
160 J=J1+IARGS(2)                                MSC 490
ELEM=ELEM+RC(J-1)                                MSC 500
IARGS(2)=IARGS(2)+1                            MSC 510
IF (IARGS(2)-IARGS(3)) 160,160,170             MSC 520
170 CALL VECTOR (ELEM,J2)                      MSC 530
GO TO 30                                         MSC 540
180 IF (NRMAX) 150,150,190                      MSC 550
C   SUM DISCRETE ROWS                         MSC 560
C
190 DO 200 I=2,NARG1                            MSC 570
                                         MSC 580
                                         MSC 590

```

```

        J=J1+IARGS(I)                         MSC 600
200    ELEM=ELEM+RC(J-1)                   MSC 610
        GO TO 170                           MSC 620
210    IF (NERROR.NE.0) GO TO 30          MSC 630
        IF (NRMAX) 150,150,220             MSC 640
220    FNRMAX=NRMAX                      MSC 650
C
C      PARSUM, PARPRODUCT
C
        IF (L2-3) 230,260,280            MSC 660
230    J=L2-1                            MSC 670
        RC(J2)=RC(J1)                   MSC 680
        IF (NRMAX.EQ.1) GO TO 30          MSC 690
        DO 250 I=2, NRMAX              MSC 700
        J1=J1+1                          MSC 710
        J2=J2+1                          MSC 720
        IF (J.EQ.0) GO TO 240           MSC 730
        RC(J2)=RC(J2-1)*RC(J1)         MSC 740
        GO TO 250                       MSC 750
240    RC(J2)=RC(J2-1)+RC(J1)         MSC 760
250    CONTINUE                         MSC 770
        GO TO 30                         MSC 780
C
C      RMS
C
260    DO 270 I=1, NRMAX              MSC 790
        J=J1+I                          MSC 800
270    ELEM=ELEM+RC(J-1)**2          MSC 810
        ELEM=FSQRT(ELEM/FNRMAX)       MSC 820
        GO TO 170                       MSC 830
C
C      AVERAGE, SUM ENTIRE COLUMN
C
280    DO 290 I=1, NRMAX              MSC 840
        J=J1+I                          MSC 850
290    ELEM=ELEM+RC(J-1)             MSC 860
        IF (L2-5) 300,170,170          MSC 870
300    ELEM=ELEM/FNRMAX            MSC 880
        GO TO 170                       MSC 890
        END                             MSC 900
                                         MSC 910
                                         MSC 920
                                         MSC 930
                                         MSC 940
                                         MSC 950
                                         MSC 960
                                         MSC 970
                                         MSC 980
                                         MSC 990

```

```

SUBROUTINE MTRIAN                               MTR   10
C      VERSION 5.00      MTRIAN      5/15/70      MTR   20
C      MTRIAN SUBROUTINE FOR OMNITAB 11/27/67 BY S PEAVY      MTR   30
C
C      TRIANGULARIZATION OF NON-SINGULAR, REAL SYMMETRIC MATRIX      MTR   50
C      A=TT, LOWER TRINGLE IS COMPUTED      MTR   60
C      COMMAND IS:      MTR   70
C      MTRIAN A(,, ++),R=,, C=,, STORE T IN (,, ++)      MTR   80
C      OR      MTR   90
C      MTRIAN A(,, ++),R=,, C=,, STORE T IN(,, ++) AND T INVERSE (,, ++)      MTR 100
C      THE UPPER TRIANGLE IS SET = 0.      MTR 110
C
C      COMMON /SCRAT/ NS,NS2,A(13500)      MTR 120
C      COMMON /BLOCRC/ NRC,RC(12600)      MTR 130
C      COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NMTR 150
1     IARGS,VWXYZ(8),NERROR      MTR 160
      DIMENSION ARGS(100)      MTR 170
      EQUIVALENCE (ARGS(1),RC(12501))      MTR 180
      DIMENSION X(2)      MTR 190
      DOUBLE PRECISION X,SUM      MTR 200
      EQUIVALENCE (X,A)      MTR 210
      KRR=7      MTR 220
      KRRA=7      MTR 230
      KRRB=7      MTR 240
      KRRC=7      MTR 250
      J=2      MTR 260
      IF (NARGS.EQ.6.OR.NARGS.EQ.8) GO TO 10      MTR 270
      CALL ERROR (10)
      RETURN
10    IF (IARGS(3).EQ.IARGS(4)) GO TO 20      MTR 280
      CALL ERROR (KRR)
      RETURN
20    IF (NARGS.EQ.6) GO TO 30      MTR 290
      J=3      MTR 300
      IARGS(9)=IARGS(7)      MTR 310
      IARGS(10)=IARGS(8)      MTR 320
      IARGS(11)=IARGS(3)      MTR 330
      IARGS(12)=IARGS(4)      MTR 340
30    IARGS(7)=IARGS(3)      MTR 350
      IARGS(8)=IARGS(4)      MTR 360
      CALL MTXCHK (J)      MTR 370
      IF (J-1) 60,40,50      MTR 380
40    CALL ERROR (3)
      RETURN
50    CALL ERROR (17)
      RETURN
60    IF (NERROR.NE.0) RETURN      MTR 410
      IR=IARGS(3)
      IRM=IR-1
      K=IARGS(1)
      DO 70 I=1,IR      MTR 420
      IF (RC(K).GT.0.0) GO TO 70      MTR 430
C      *** ERRA -MATRIX CAN NOT BE TRIANLIZED SINCE ONE OF THE TERMS ON      MTR 440
C      THE DIAG. IS ZERO OR LESS.      MTR 450
      CALL ERROR (KRRA)
      RETURN
70    K=K+1+NROW      MTR 460
      K=IARGS(1)
      CALL SYMV (RC(K),NROW,IR,M)      MTR 470

```

```

      IF (M.LE.1) GO TO 80                                MTR 600
C      *** NON-SYMMETRIC MATRIX                         MTR 610
      CALL ERROR (KRRC)                                 MTR 620
      RETURN                                              MTR 630
80     M=2                                                MTR 640
      A(1)=FSQRT(RC(K))                               MTR 650
      K=K+1                                             MTR 660
      DO 90 I=2,IR                                     MTR 670
      A(M)=RC(K)/A(1)                                 MTR 680
      K=K+1                                             MTR 690
90     M=M+1                                            MTR 700
      KA=IARGS(1)                                      MTR 710
      KB=KA+NROW+1                                    MTR 720
      MA=2                                              MTR 730
      DO 140 I=2,IR                                     MTR 740
      MB=MA                                             MTR 750
      L=NS2-1                                           MTR 760
      X(NS2)=RC(KB)                                   MTR 770
      M=(I-1)*IR+I                                    MTR 780
      II=I-1                                           MTR 790
      DO 100 J=1,II                                     MTR 800
      X(L)=-A(MB)**2                                  MTR 810
      L=L-1                                            MTR 820
100    MB=MB-IR                                       MTR 830
      CALL SORTSM (I,SUM)                            MTR 840
      IF (SUM.GT.0.0) GO TO 110                      MTR 850
C      *** ERRB-LEADING SUBMATRIX IS SINGULAR        MTR 860
      CALL ERROR (KRRB)                             MTR 870
      RETURN                                            MTR 880
110    S=SUM                                           MTR 890
      S=FSQRT(S)                                      MTR 900
      A(M)=S                                         MTR 910
      M=M+1                                           MTR 920
      IF (I.EQ.IR) GO TO 140                        MTR 930
      IP=I+1                                           MTR 940
      KC=KB+1                                         MTR 950
      DO 130 J=IP,IR                                 MTR 960
      X(NS2)=RC(KC)                                   MTR 970
      KC=KC+1                                         MTR 980
      L=NS2-1                                         MTR 990
      MC=J                                           MTR1000
      MD=I                                           MTR1010
      DO 120 JJ=1,II                                 MTR1020
      X(L)=-A(MC)*A(MD)                           MTR1030
      MC=MC+IR                                       MTR1040
      MD=MD+IR                                       MTR1050
120     L=L-1                                         MTR1060
      CALL SORTSM (I,SUM)                            MTR1070
      A(M)=SUM/S                                     MTR1080
130     M=M+1                                         MTR1090
      MA=MA+IR+1                                    MTR1100
140     KB=KB+NROW+1                                MTR1110
      K=IARGS(5)-1                                  MTR1120
      KB=IARGS(5)                                    MTR1130
      DO 180 I=1,IR                                 MTR1140
      KA=K+I                                         MTR1150
      M=(I-1)*IR+I                                MTR1160
      KC=KB                                           MTR1170
      DO 150 J=I,IR                                 MTR1180

```

	RC(KA)=A(M)	MTR1190
	KA=KA+1	MTR1200
150	M=M+1	MTR1210
	IF (I.EQ.1) GO TO 170	MTR1220
	II=I-1	MTR1230
	DO 160 J=1,II	MTR1240
	RC(KC)=0.0	MTR1250
160	KC=KC+1	MTR1260
170	KB=KB+NROW	MTR1270
180	K=K+NROW	MTR1280
	IF (NARGS.EQ.6) RETURN	MTR1290
	KC=IARGS(5)	MTR1300
	DO 210 I=1,IR	MTR1310
	M=(I-1)*IR+I	MTR1320
	A(M)=1.0/RC(KC)	MTR1330
	IF (I.EQ.IR) GO TO 210	MTR1340
	M=M+1	MTR1350
	IP=I+1	MTR1360
	KB=KC+NROW+1	MTR1370
	JC=1	MTR1380
	DO 200 J=IP,IR	MTR1390
	KA=KC+J-I	MTR1400
	MA=(I-1)*IR+I	MTR1410
	L=NS2	MTR1420
	DO 190 JA=1,JC	MTR1430
	X(L)=RC(KA)*A(MA)	MTR1440
	MA=MA+1	MTR1450
	KA=KA+NROW	MTR1460
190	L=L-1	MTR1470
	CALL SORTSM (JC,SUM)	MTR1480
	S=SUM	MTR1490
	A(M)==-S/RC(KB)	MTR1500
	KB=KB+NROW+1	MTR1510
	M=M+1	MTR1520
200	JC=JC+1	MTR1530
210	KC=KC+NROW+1	MTR1540
	K=IARGS(9)-1	MTR1550
	KB=IARGS(9)	MTR1560
	DO 250 I=1,IR	MTR1570
	KA=K+I	MTR1580
	M=(I-1)*IR+I	MTR1590
	KC=KB	MTR1600
	DO 220 J=I,IR	MTR1610
	RC(KA)=A(M)	MTR1620
	KA=KA+1	MTR1630
220	M=M+1	MTR1640
	IF (I.EQ.1) GO TO 240	MTR1650
	II=I-1	MTR1660
	DO 230 J=1,II	MTR1670
	RC(KC)=0.	MTR1680
230	KC=KC+1	MTR1690
240	KB=KB+NROW	MTR1700
250	K=K+NROW	MTR1710
	RETURN	MTR1720
	END	MTR1730

```

SUBROUTINE MTXCHK (J) MCK 10
C VERSION 5.00 MTXCHK 5/15/70 MCK 20
C S PEAVY FOR OMNITAB 10/24/67 MCK 30
C J AS INPUT = NO OF MATRICES TO BE CHECKED MCK 40
C IARGS(1), IARGS(5),...,IARGS(4*(J-1)+1) STARTING ROW OF MAT MCK 50
C IARGS(2), IARGS(6),...,IARGS(4*(J-1)+2) STARTING COLUMN OF MAT MCK 60
C IARGS(3), IARGS(7),...,IARGS(4*(J-1)+3) NO. OF ROWS MCK 70
C IARGS(4), IARGS(8),...,IARGS(4*(J-1)+4) NO OF COLUMNS MCK 80
C MCK 90
C UPON RETURN MCK 100
C J=0 IF ALL MATRICES ARE IN WORK SHEET MCK 110
C AND MCK 120
C IARGS(1),IARGS(5),...,IARGS(4*(J-1)+1) WILL CONTAIN STARTING MCK 130
C ADDRESS OF MATRIX MCK 140
C J GT ZERO IF MATRIX IS NOT IN WORK SHEET MCK 150
C J=1 SOME IARGS ARE NEGATIVE, J=2 MATRIX TO BIG FOR WORK SHEET MCK 160
C COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NMCK 180
1ARGS,VWXYZ(8),NERROR MCK 190
JA=J MCK 220
JB=4*J MCK 230
J=0 MCK 240
DO 10 I=1,JB MCK 250
IF (IARGS(I).GT.0) GO TO 10 MCK 260
J=1 MCK 270
RETURN MCK 280
10 CONTINUE MCK 290
DO 20 I=1,JB,4 MCK 300
IF (IARGS(I)+IARGS(I+2)-1.GT.NROW) GO TO 30 MCK 310
IF (IARGS(I+1)+IARGS(I+3)-1.GT.NCOL) GO TO 30 MCK 320
20 IARGS(I)=IARGS(I)+(IARGS(I+1)-1)*NROW MCK 330
RETURN MCK 340
30 J=2 MCK 350
RETURN MCK 360
END MCK 370

```

```

SUBROUTINE MXTX          MXT   10
C VERSION 5.00      MXTX      5/15/70    MXT   20
C SUBROUTINE MXTX R.V.  5/7/68    MXT   30
C *
C SUBROUTINE TO MULTIPLY MATRIX A BY ITS TRANSPOSE    MXT   40
C           OR TRANSPOSE OF MATRIX A BY MATRIX A    MXT   50
C           L2=1 MULTIPLY MATRIX BY ITS TRANSPOSE    MXT   60
C           GENERAL FORM OF COMMAND    MXT   70
C           M(XXT) A(,) N,K, STORE IN C(,) N,K DEFINE X    MXT   80
C           L2=2 MULTIPLY TRANSPOSE OF MATRIX BY ITSELF    MXT  110
C           GENERAL FORM OF COMMAND    MXT  120
C           M(XTX) A(,) N,K STORE IN C(,) N,K DEFINE X    MXT  130
C *
C COMMON /SCRAT/ NS,NS2,A(13500)    MXT  150
C COMMON /BLOCRC/ NRC,RC(12600)    MXT  160
C COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NMXT 180
1ARGS,VWXYZ(8),NERROR    MXT  190
DIMENSION ARGS(100)    MXT  200
EQUIVALENCE (ARGS(1),RC(12501))    MXT  210
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG    MXT  220
DOUBLE PRECISION AP(3000)    MXT  230
EQUIVALENCE (A,AP)    MXT  240
C *
C CHECK FOR CORRECT NUMBER OF AGRUMENTS    MXT  250
C *
C *
C DECIDE WHETHER COMMAND IS M(XAX') OR M(X'AX)    MXT  260
C L2=3 MEANS M(XAX') L2=2 NARGS.GT. 6 MEANS M(X'AX)    MXT  270
C *
C IF (L2-2) 30,10,20    MXT  280
10 IF (NARGS.LE.6) GO TO 30    MXT  290
20 L2=4-L2    MXT  300
CALL TRANSF    MXT  310
RETURN    MXT  320
30 IF(NARGS.NE.6) CALL ERROR (10)    MXT  330
C *
C CHECK TO SEE IF ALL ARGUMENTS ARE INTEGERS    MXT  340
C *
C J=NARGS    MXT  350
CALL CKIND (J)    MXT  360
IF (J.NE.0) CALL ERROR (3)    MXT  370
C *
C CHECK TO SEE IF DIMENSIONS ARE OUT OF RANGE    MXT  380
C COMPUTE ADDRESSES    MXT  390
C *
C 40 GO TO (50,60), L2    MXT  400
50 IARGS(8)=IARGS(3)    MXT  410
IARGS(7)=IARGS(3)    MXT  420
GO TO 70    MXT  430
C *
C 60 IARGS(8)=IARGS(4)    MXT  440
IARGS(7)=IARGS(4)    MXT  450
70 J=2    MXT  460
CALL MTXCHK (J)    MXT  470
IF (J-1) 100,80,90    MXT  480
80 CALL ERROR (3)    MXT  490
RETURN    MXT  500
90 CALL ERROR (17)    MXT  510
RETURN    MXT  520
C *

```

```
C      CHECK FOR PREVIOUS ERRORS          MXT 660
C      *
100  IF (NERROR.NE.0) RETURN          MXT 670
    IG=IARGS(1)
    CALL MXTXP (RC(IG),NROW,IARGS(3),IARGS(4),A,L2,NS2,AP) MXT 680
    GO TO (110,120), L2          MXT 690
110  NROWP=IARGS(3)          MXT 700
    GO TO 130          MXT 710
120  NROWP=IARGS(4)          MXT 720
130  NCOLP=NROWP          MXT 730
    IG=IARGS(5)
    CALL STORMT (RC(IG),NROW,NROWP,NCOLP,A) MXT 740
    *
C      MOVE FROM SCRATCH AREA TO STORAGE MXT 750
C      *
C      RETURN          MXT 760
END          MXT 770
          MXT 780
          MXT 790
          MXT 800
          MXT 810
          MXT 820
```

```

C SUBROUTINE MXTXP (X,N,NP,K,A,L2,NASIZE,XP) MXP 10
C VERSION 5.00      MXTXP      5/15/70 MXP 20
C SUBROUTINE MXTXP      R VARNER      2/12/68 MXP 30
C *
C X IS MATLIX TO BE USED MXP 40
C N IS DIMENSIONED SIZE OF A MXP 50
C NP IS NUMBER OF ROWS IN A MXP 60
C K IS NUMBER OF COLUMNS IN A MXP 70
C L2=1 MULTIPLY X TIMES X TRANSPOSED MXP 80
C L2=2 MULTIPLY X TRANSPOSED TIMES X MXP 90
C NASIZE IS SIZE OF A DIVIDED BY 2 MXP 100
C A IS SCRATCH AREA WHERE MATRIX IS TO BE STORED MXP 110
C *
C DOUBLE PRECISION XP,SUM MXP 120
C DIMENSION X(N,1), A(1), XP(1) MXP 130
C IC=1
C GO TO (10,40), L2 MXP 140
C COMPUTE X TIMES X TRANSPOSED MXP 150
C *
10 DO 30 KK=1,NP MXP 160
DO 30 I=1,NP MXP 170
IS=NASIZE MXP 180
DO 20 J=1,K MXP 190
XP(IS)=X(I,J)*X(KK,J)
20 IS=IS-1 MXP 200
CALL SORTSM (K,SUM) MXP 210
A(IC)=SUM MXP 220
30 IC=IC+1 MXP 230
RETURN MXP 240
C *
C COMPUTE X TRANSPOSED TIMES X MXP 250
C *
40 DO 60 L=1,K MXP 260
DO 60 J=1,K MXP 270
IS=NASIZE MXP 280
DO 50 I=1,NP MXP 290
XP(IS)=X(I,J)*X(I,L)
50 IS=IS-1 MXP 300
CALL SORTSM (NP,SUM) MXP 310
A(IC)=SUM MXP 320
60 IC=IC+1 MXP 330
RETURN MXP 340
END MXP 350
MXP 360
MXP 370
MXP 380
MXP 390
MXP 400
MXP 410
MXP 420
MXP 430

```

```

C      SUBROUTINE NNAME (NAME)          NNA 10
C      VERSION 5.00      NNAME      5/15/70   NNA 20
C      COMMON /BLOCKA/ MODE,M,KARD(83),KARG,ARG,ARG2,NEWCD(80),KRDEND NNA 30
C      DIMENSION NAME(2), MISC(6)        NNA 40
C                                         NNA 50
C      THIS SUBROUTINE ASSEMBLES A NAME UP TO THE FIRST NON-LETTER OR UP NNA 60
C      SIX LETTER, WHICHEVER IS FIRST. THE INDEX, M, IS INITIALLY POINTINNNA 70
C      THE FIRST LETTER, IT IS LEFT POINTING AT THE FIRST NON-LETTER.   NNA 80
C                                         NNA 90
C      SPACE OUT SO THAT TABLE LIES ALL ON ONE PAGE      NNA 100
C                                         NNA 110
C                                         NNA 120
C      CONVERSION TABLE FOR ALPHABETIC TO NUMERIC AS USED BY OMNITAB. NNA 130
C                                         NNA 140
C
C      A      729      27      1      NNA 150
C      B      1458     54      2      NNA 160
C      C      2187     81      3      NNA 170
C      D      2916    108      4      NNA 180
C      E      3645    135      5      NNA 190
C      F      4374    162      6      NNA 200
C      G      5103    189      7      NNA 210
C      H      5832    216      8      NNA 220
C      I      6561    243      9      NNA 230
C      J      7290    270     10      NNA 240
C      K      8019    297     11      NNA 250
C      L      8748    324     12      NNA 260
C      M      9477    351     13      NNA 270
C      N     10206   378     14      NNA 280
C      O     10935   405     15      NNA 290
C      P     11664   432     16      NNA 300
C      Q     12393   459     17      NNA 310
C      R     13122   486     18      NNA 320
C      S     13851   513     19      NNA 330
C      T     14580   540     20      NNA 340
C      U     15309   567     21      NNA 350
C      V     16038   594     22      NNA 360
C      W     16767   621     23      NNA 370
C      X     17496   648     24      NNA 380
C      Y     18225   675     25      NNA 390
C      Z     18954   702     26      NNA 400
C                                         NNA 410
C                                         NNA 420
C
C      THE FIRST THREE CHARACTERS GO INTO THE FIRST WORD OF NAME      NNA 430
C      THE SECOND THREE CHARACTERS GO INTO THE SECOND WORD OF NAME      NNA 440
C                                         NNA 450
C                                         NNA 460
C
C      DO 10 I=1,6          NNA 470
10     MISC(I)=0          NNA 480
      DO 20 I=1,6          NNA 490
      L=KARD(M)-9          NNA 500
      IF (L.LT.1.OR.L.GE.27) GO TO 40      NNA 510
      MISC(I)=L          NNA 520
20     M=M+1          NNA 530
30     IF (KARD(M).LT.10.OR.KARD(M).GE.36) GO TO 40      NNA 540
      M=M+1          NNA 550
      GO TO 30          NNA 560
40     NAME(1)=MISC(3)+27*(MISC(2)+27*MISC(1))      NNA 570
      NAME(2)=MISC(6)+27*(MISC(5)+27*MISC(4))      NNA 580
      RETURN          NNA 590
      END          NNA 600

```

```

FUNCTION NONBLA (I)                               NON   10
C      VERSION 5.00      NONBLA     5/15/70        NON   20
C
C      SCAN KARD STARTING AT KARD(I) UNTIL A NON-BLANK CHARACTER IS    NON   30
C      FOUND. POINT M AT IT AND ALSO RETURN IT AS FUNCTION VALUE.    NON   40
C      COMMON /BLOCKA/ MODE,M,KARD(83),KARG,ARG,ARG2,NEWCD(80),KRDEND  NON   50
C      M=I
10     IF (KARD(M).NE.44) GO TO 20                NON   60
      M=M+1                                         NON   70
      GO TO 10                                       NON   80
20     NONBLA=KARD(M)                            NON   90
      RETURN                                         NON  100
      END                                           NON  110
                                                NON  120
                                                NON  130

```

```

SUBROUTINE NOTEPR(J)                           NOT   10
C      VERSION 5.00      NOTEPR     5/15/70        NOT   20
C      WRITTEN BY STP  4/21/70        NOT   30
C
C      IF J=0 BLANK OUT NOTE VARIABLE          NOT   40
C      IF J=1 STORE IN NOTE(1) THRU NOTE(60) FROM NEWCD (M-2) 60 CHAR  NOT   50
C      IF J=2 STORE IN NOTE(61) THRU NOTE(120) FROM NEWCD (M-2) 60 CHAR NOT   60
C      IF J=3 PRINT OUT NOTE(1) THRU NOTE(120)  NOT   70
C
C      COMMON/BLOCKA/MODE,M,KARD(83),KARG,ARG,ARG2,NEWCD(80),KRDEND  NOT   80
C      COMMON/HEADER/NOCARD(80),ITLE(60,6),LNCNT,IPRINT,NPAGE,IPUNCH  NOT   90
C      COMMON/NOTE/NOTE(120)
C      DATA IBLANK/1H /
C      IF (J.NE.0) GO TO 20
C      DO 10 I=1,120
10     NOTE(I)=IBLANK                         NOT  100
      RETURN                                     NOT  110
20     IF(J.NE.3) GO TO 40
      IF(NPAGE.EQ.0) CALL PAGE(0)               NOT  120
      WRITE (IPRINT,30) (NOTE(I),I=2,120)       NOT  130
      RETURN                                     NOT  140
      NOT 150
30     FORMAT(1X,119A1)                        NOT  160
40     MA=M+60                                NOT  170
      M=M+1
      IF(MA.GT.82) MA=82
      MB=(J-1)*60+1
      MC=MB+59
      IF(J.NE.1.AND.J.NE.2) RETURN
      DO 100 I=MB,MC
100    NOTE(I)=IBLANK
      I=MB
      DO 110 IC=M,MA
      NOTE(I)=NEWCD(IC-2)
110    I=I+1
      RETURN
      END
                                                NOT  210
                                                NOT  220
                                                NOT  230
                                                NOT  240
                                                NOT  250
                                                NOT  260
                                                NOT  270
                                                NOT  280
                                                NOT  290
                                                NOT  300
                                                NOT  310
                                                NOT  320
                                                NOT  330
                                                NOT  340
                                                NOT  350
                                                NOT  360

```

```

C SUBROUTINE OANOVA (YSUM,SU,ND9,FM,M,N,ND7,SSQ,IHC,NSU,B)          OAN 10
C VERSION 5.00      OANOVA      5/15/70                           OAN 20
C COMPUTE AND PRINT ANALYSIS OF VARIANCE                         OAN 30
C WRITTEN BY DAVID HOBGEN, SEL, NBS.    10/09/69.                  OAN 40
C *****
C COMMON/BLOCKD/IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,   OAN 60
1NARGS,VWXYZ(8),NERROR                                         OAN 70
COMMON/BLOCKE/NAME(4),L1,L2,ISRFLG                                OAN 90
COMMON/HEADER/NOCARD(80),ITLE(60,6),LNCNT,IPRINT,NPAGE,IPUNCH    OAN 100
COMMON/SCRAT/NS,NS2,A(13500)                                       OAN 110
DIMENSION B(1),IHC(1)                                              OAN 160
DOUBLE PRECISION YSUM                                            OAN 170
C *****
C
1850 RESMS = YSUM/SU                                             OAN 180
NSUA = NSU                                                       OAN 190
IT = 1                                                       OAN 200
IF (L2.EQ.3) IT=3                                              OAN 210
WRITE (IPRINT,1860) IHC(IT),IHC(IT+1)                            OAN 220
1860 FORMAT (////50X,20HANALYSIS OF VARIANCE/24X,73H-DEPENDENT ON ORDEROAN 240
1 VARIABLES ARE ENTERED, UNLESS VECTORS ARE ORTHOGONAL--//          OAN 250
21X,2A3,4X,           21H SS=RED. DUE TO COEF.,21H CUM. MS REDUCTIOOAN 260
3N ,6H D.F.,21H CUM. RESIDUAL MS ,6H D.F.,11H F(COEF=0),6H POAN 270
4(F),11H F(COEFS=0),6H P(F)//                                     OAN 280
IND9 = ND9+M                                              OAN 290
ASUM = 0.0                                                 OAN 300
VR = SU-FM                                              OAN 340
RESSS = VR*SSQ                                             OAN 350
IND7 = ND7+M                                              OAN 400
A(IND7) = RESSS                                           OAN 410
IF(M.EQ.1) GO TO 1866                                      OAN 415
DO 1865 I1=2,M                                              OAN 420
IND7 = IND7-1                                              OAN 430
A(IND7) = A(IND7+1) + A(IND9)                                OAN 440
1865 IND9 = IND9-1                                           OAN 450
1866 V1F2 = FM+1.0                                           OAN 460
B(1) = A(IND9-1)                                           OAN 470
B(2) = A(IND9-2)                                           OAN 480
A(IND9-1) = YSUM                                           OAN 482
A(IND9-2) = RESSS                                         OAN 484
CALL RFORMAT (A(IND9-2),M+2,8,NW1,NDEC1,18,A(1),A(1),0,0)    OAN 490
A(IND9-1) = B(1)                                             OAN 492
A(IND9-2) = B(2)                                             OAN 494
CALL RFORMAT (A(IND9),M,8,NW2,NDEC2,18,A(1),A(1),0,0)    OAN 496
SSU=SU                                                       OAN 498
DO 1867 I=1,M                                              OAN 500
SSU=SSU-1.0                                                 OAN 505
B(I)=A(IND7)/SSU                                           OAN 510
1867 IND7=IND7+1                                           OAN 515
IND7=IND7-M                                              OAN 520
CALL RFORMAT(B(1),M,8,NW3,NDEC3,18,A(1),A(1),0,0)    OAN 525
SSU = SU                                                       OAN 530
DO 1950 I=1,M                                              OAN 540
NSUA = NSUA-1                                              OAN 550
ASUM = ASUM+A(IND9)                                         OAN 560
SSU = SSU-1.0                                               OAN 570
CR = ASUM/FLOAT(I)                                         OAN 580
IF (ABS(SSU).GT.0.0) GO TO 1880                           OAN 590
RESMS = 0.0                                                 OAN 600
1870 F1 = 0.0                                              OAN 610

```

```

F2 = 0.0          OAN 620
PF1 = 1.0         OAN 630
PF2 = 1.0         OAN 640
GO TO 1890       OAN 650
1880  RESMS = A(IND7)/SSU      OAN 660
      V1F2 = V1F2-1.0           OAN 670
      IF (ABS(RESMS).LE.0.0) GO TO 1870   OAN 680
C     NEVER POOL             OAN 690
      F1 = A(IND9)/SSQ        OAN 700
      CALL PROB (1.,VR,F1,PF1) OAN 710
C     TEST HIGHER SUB-HYPOTHESES OAN 720
      F2 = ((A(IND7)+A(IND9)-RESSS)/V1F2)/SSQ OAN 730
      CALL PROB (V1F2,VR,F2,PF2) OAN 740
1890  II = IABS(I-1)          OAN 750
      CALL RFORMAT (A(1),1,8,NW1,NDEC1,0,A(IND9),B( 1),21-NW1,1) OAN 760
      CALL RFORMAT (A(1),1,8,NW2,NDEC2,0, CR ,B(22),21-NW2,1) OAN 770
      CALL RFORMAT (A(1),1,8,NW3,NDEC3,0, RESMS ,B(43),21-NW3,1) OAN 780
      IF (L2.EQ.1) GO TO 1920    OAN 790
      WRITE (IPRINT,1900) IARGS(I+3),(B(I1),I1=1,42),I,(B(I2),I2=43,63),OAN 800
      1NSUA,F1,PF1,F2,PF2      OAN 810
1900  FORMAT (1X,I4,6X,42A1,I6,21A1,I6,2(0PF11.3,F6.3)) OAN 820
      GO TO 1940              OAN 830
1920  WRITE (IPRINT,1900) II,(B(I1),I1=1,42),I,(B(I2),I2=43,63),OAN 840
      1NSUA,F1,PF1,F2,PF2      OAN 850
1940  IND7 = IND7+1          OAN 860
1950  IND9 = IND9+1          OAN 870
1951  FORMAT (/,1X,10HRESIDUAL ,21A1,21X,I6) OAN 880
1952  FORMAT (1X, 10HTOTAL   ,21A1,21X,I6) OAN 890
      F1 = RESSS               OAN 900
      CALL RFORMAT (A(1),1,8,NW1,NDEC1,0,F1           B(1),21-NW1,1) OAN 910
      WRITE (IPRINT,1951) (B(I),I=1,21),NSUA        OAN 920
      F2 = YSUM                 OAN 930
      CALL RFORMAT (A(1),1,8,NW1,NDEC1,0,F2,B(1),21-NW1,1) OAN 940
      WRITE (IPRINT,1952) (B(I),I=1,21),NSU        OAN 950
      RETURN                   OAN 960
      END                      OAN 970

```

```

SUBROUTINE OCOEFF (M1,N,ND18,ND17,IND19S,IND18S,IHC,B,IND7S,NSU,SSOCO 10
1,SSOLD,YSUM)                                            OCO 20
C      VERSION 5.00      OCOEFF      5/15/70          OCO 30
C      WRITTEN BY DAVID HOBGEN, SEL, NBS.    10/14/69.      OCO 40
C      *****
C      COMMON/BLOCKD/IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,      OCO 60
1NARGS,VWXYZ(8),NERROR                                OCO 70
COMMON/BLOCKE/NAME(4),L1,L2,ISRFLG                      OCO 90
COMMON/HEADER/NOCARD(80),ITLE(60,6),LNCNT,IPRINT,NPAGE,IPUNCH      OCO 100
COMMON/SCRAT/NS,NS2,A(13500)                            OCO 110
DIMENSION B(1),IHC(1)                                  OCO 160
DOUBLE PRECISION YSUM                                 OCO 165
C      *****
C      IT = 1                                         OCO 180
IF (L2.EQ.3) IT=3                                     OCO 190
M = M1+1                                         OCO 200
WRITE (IPRINT,1960) IHC(IT),IHC(IT+1),IHC(IT),IHC(IT+1)      OCO 210
1960 FORMAT (////20X,32HESTIMATES FROM LEAST SQUARES FIT,38X,18HFIT OMIOCO 220
1TTING LAST ,2A3//1X,2A3,5X,11HCOEFFICIENT,8X,14HS.D. OF COEFF.,4X,OCO 230
25HRATIO,3X,12H*ACC. DIGITS,9X,11HCOEFFICIENT,7X,14HS.D. OF COEFF.,OCO 240
35X,5HRATIO/)                                         OCO 250
IND18 = N+ND18+1                                      OCO 260
IND17 = ND17+1                                         OCO 270
IND1 = IND19S+1                                       OCO 280
IND19 = IND18S+1                                      OCO 290
IND7 = IND7S+1                                         OCO 300
CALL RFORMAT (A(IND19),M1+1,8,NW1,NDEC1,18,A(1),A(1),0,0)      OCO 310
CALL RFORMAT (A(IND1 ),M1+1,8,NW2,NDEC2,18,A(1),A(1),0,0)      OCO 320
IF (M1.EQ.0) GO TO 2005                               OCO 330
CALL RFORMAT (A(IND18),M1 ,8,NW3,NDEC3,18,A(1),A(1),0,0)      OCO 340
CALL RFORMAT (A(IND17),M1 ,8,NW4,NDEC4,18,A(1),A(1),0,0)      OCO 350
DO 2000 J=1,M1                                         OCO 360
IF (A(IND1)) 1982,1981,1982                         OCO 370
1981 F1 = 0.0                                         OCO 380
GO TO 1983                                         OCO 390
1982 F1 = A(IND19)/A(IND1)                           OCO 400
1983 IF (A(IND17)) 1985,1984,1985                  OCO 410
1984 F2 = 0.0                                         OCO 420
GO TO 1986                                         OCO 430
1985 F2 = A(IND18)/A(IND17)                           OCO 440
1986 CALL RFORMAT (A(1),1,8,NW1,NDEC1,0,A(IND19),B(1),20-NW1,1)      OCO 450
CALL RFORMAT (A(1),1,8,NW2,NDEC2,0,A(IND1 ),B(21),20-NW2,1)      OCO 460
CALL RFORMAT (A(1),1,8,NW3,NDEC3,0,A(IND18),B(41),20-NW3,1)      OCO 470
CALL RFORMAT (A(1),1,8,NW4,NDEC4,0,A(IND17),B(61),20-NW4,1)      OCO 480
JJ = L2/3                                         OCO 490
JJ = IABS(J-1)*(1-JJ)+IARGS(J+3)*JJ                 OCO 500
WRITE (IPRINT,1990) JJ,(B(I1),I1=1,40),F1,A(IND7),(B(I2),I2=41,80)OCO 510
1,F2                                         OCO 520
1990 FORMAT (1X,I4,2X, 40A1,0PF7.2,6X,F5.2,8X,40A1,F7.2)      OCO 530
IND7 = IND7+1                                         OCO 540
IND19 = IND19+1                                       OCO 550
IND17 = IND17+1                                       OCO 560
IND1 = IND1+1                                         OCO 570
2000 IND18 = IND18+1                                  OCO 580
2005 CALL RFORMAT (A(1),1,8,NW1,NDEC1,0,A(IND19),B(1),20-NW1,1)      OCO 590
CALL RFORMAT (A(1),1,8,NW2,NDEC2,0,A(IND1 ),B(21),20-NW2,1)      OCO 600
F1 = A(IND19)/A(IND1 )                           OCO 610
NSUA = NSU-M                                         OCO 620
NRM1= NSU-M1                                         OCO 630

```

```

JJ = L2/3          OCO 640
JJ = IABS(M1)*(1-JJ)+IARGS(M1+4)*JJ      OCO 650
WRITE (IPRINT,1990) JJ,(B(I),I=1,40),F1,A(IND7) OCO 660
CALL RFORMAT (SSOLD,1,8,NW2,NDEC2,18,A(1),A(1),0,0) OCO 670
B(52)=SS          OCO 672
C   ADJUST FOR M1=0          OCO 674
IF(M1.NE.0) GO TO 2009      OCO 676
B(52) = YSUM/FLOAT(N)      OCO 678
B(52) = FSQRT(B(52))      OCO 680
2009 CALL RFORMAT (B(52),1,8,NW4,NDEC4,18,A(1),A(1),0,0) OCO 682
CALL RFORMAT (A(1),1,8,NW2,NDEC2,0,SSOLD,B(1),20-NW2,0) OCO 690
CALL RFORMAT (A(1),1,8,NW4,NDEC4,0,B(52),B(21),20-NW4,0) OCO 700
WRITE (IPRINT,2010) (B(I),I=1,40),NSU,M,NSUA,NSU,M1,NRM1 OCO 710
2010 FORMAT (/1X,30HRESIDUAL STANDARD DEVIATION = ,3X,20A1,26X,20A1/4X,OCO 720
128HBASED ON DEGREES OF FREEDOM ,9X,I4,1H-,I2,3H = ,I3,33X,I4,1H-, OCO 730
2I2,3H = ,I3//120H * THE NUMBER OF CORRECTLY COMPUTED DIGITS IN EACOCO 740
3H COEFFICIENT USUALLY DIFFERS BY LESS THAN 1 FROM THE NUMBER GIVENOCO 750
4 HERE)           OCO 760
RETURN            OCO 770
END               OCO 780

```

```

C SUBROUTINE OCOVAR (M,ND7,MD1,IHC,B,IHT)          OCV 10
C VERSION 5.00      OCOVAR      5/15/70           OCV 20
C PRINT VARIANCE-COVARIANCE MATRIX             OCV 30
C WRITTEN BY DAVID HOGBEN, SEL, NBS.   10/10/69.    OCV 40
C *****
C COMMON/BLOCKD/IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,   OCV 60
1NARGS,VWXYZ(8),NERROR                         OCV 70
COMMON/BLOCKE/NAME(4),L1,L2,ISRFLG               OCV 80
COMMON/HEADER/NOCARD(80),ITLE(60,6),LNCNT,IPRINT,NPAGE,IPUNCH   OCV 90
COMMON/SCRAT/NS,NS2,A(13500)                     OCV 100
DIMENSION IHC(1),B(1),IHT(1)                   OCV 110
1750 IND7 = ND7+1                                OCV 120
1791 FORMAT (///31X,56H VARIANCE-COVARIANCE MATRIX OF THE ESTIMATED COE OCV 130
1FFICIENTS)                                     OCV 140
1792 FORMAT (/1X,2A3,1X,7(6X,I5,4X))            OCV 150
1793 FORMAT (1X,I4,3X,106A1)                      OCV 160
WRITE (IPRINT,1791)                            OCV 180
CALL RFORMAT (A(IND7),MD1,8,NW1,NDEC1,13,A(1),A(1),0,0) OCV 190
IF (L2.EQ.1) I6=-1                           OCV 200
IF (L2.EQ.3) I6=3                           OCV 210
C I1END = NUMBER OF BLOCKS OF PRINTING          OCV 220
I1END=(M+6)/7                                 OCV 230
DO 1820 I1=1,I1END                           OCV 240
I3BEG=7*(I1-1)+1                           OCV 250
I2BEG=I3BEG+I6                           OCV 260
I2END = MIN0 (M+I6,I2BEG+6)                 OCV 270
IF (L2-2) 1811,1811,1813                  OCV 280
1811 I7END=I2END+1-I2BEG                  OCV 290
DO 1812 I7=1,I7END                           OCV 300
1812 IHT(I7)=I2BEG-1+I7                  OCV 310
WRITE (IPRINT,1792) IHC(L2),IHC(L2+1),(IHT(I7),I7=1,I7END) OCV 320
GO TO 1814                           OCV 330
1813 WRITE (IPRINT,1792) IHC(L2),IHC(L2+1),(IARGS(I2),I2=I2BEG,I2END) OCV 340
1814 WRITE (IPRINT,1793)                      OCV 350
LOC1 = IND7+(I3BEG*(I3BEG+1))/2-1        OCV 360
C I3 IS FOR LOOP ON ROWS                      OCV 370
DO 1820 I3=I3BEG,M                         OCV 380
I4END=MIN0 (I3,I3BEG+6) + 1 - I3BEG       OCV 390
C I4 IS FOR LOOP ON COLUMNS                  OCV 400
DO 1815 I4 = 1, I4END                      OCV 410
CALL RFORMAT (A(1),1,8,NW1,NDEC1,0,A(LOC1),B(15*I4-14),15-NW1,0) OCV 420
1815 LOC1=LOC1+1                           OCV 430
I5END = 15*I4END                          OCV 440
IF (L2.EQ.1) I8 = I3-1                    OCV 450
IF (L2.EQ.3) I8=IARGS(I3+3)              OCV 460
WRITE (IPRINT,1793) I8,(B(I5),I5=1,I5END) OCV 470
1820 LOC1 = LOC1-I4END+I3                OCV 480
RETURN                                     OCV 490
END                                         OCV 500

```

```

C      SUBROUTINE OMCONV (NWCD,KRD,KRDEND)          OMC   10
C      VERSION 5.00      OMCONV      5/15/70        OMC   20
C      COMMON /ABCDEF/ L(48)                         OMC   30
C
C      ARRAY L CONTAINS THE ALPHABET FORMATTED 1H      OMC   40
C
C      THIS ROUTINE CONVERTS INPUT CARD IMAGES TO A STANDARD CODE SO OMC   70
C      THAT OMNITAB CAN DEAL WITH THE CHARACTERS AS INTEGERS.      OMC   80
C
C
C      THIS ROUTINE IS INCLUDED ONLY FOR COMPLETENESS. IT SHOULD BE OMC  110
C      REWRITTEN IN ASSEMBLY LANGUAGE FOR EACH COMPUTER. ALSO, IT OMC  120
C      CANNOT MEET ASA STANDARDS BECAUSE ASA DOES NOT REQUIRE THAT DATA OMC  130
C      READ WITH FORMAT A1 BE STORED THE SAME AS HOLLERITH DATA SETUP OMC  140
C      WITH 1H ALTHOUGH THEY WILL BE THE SAME ON MOST COMPUTERS.    OMC  150
C
C      ALSO, ASA DOES NOT RECOGNIZE THE CHARACTER ' APOSTROPHE      OMC  160
C
C      DIMENSION NWCD(1), KRD(1)                         OMC  220
C      DO 30 I=1,KRDEND                                OMC  230
C      K=NWCD(I)                                      OMC  240
C      SPECIAL CASE TO CHECK FOR BLANKS                OMC  250
C      IF(K.NE.L(45)) GO TO 10                          OMC  260
C      J=45                                           OMC  270
C      GO TO 30                                         OMC  280
C
C      THE UPPER BOUND OR LIMIT ON J MUST BE CHANGED IF MORE CHARACTERS OMC  300
C      ARE ADDED TO THE VECTOR L IN LABEL COMMON ABCDEF      OMC  310
C
C
10  DO 20 J=1,48                                     OMC  330
     IF(K.EQ.L(J)) GO TO 30                          OMC  340
20  CONTINUE                                         OMC  350
     J=47                                           OMC  360
30  KRD(I)=J-1                                      OMC  370
     RETURN                                         OMC  380
     END                                            OMC  410

```

```

SUBROUTINE OMNIT          OMN   10
C VERSION 5.00    OMNIT      5/15/70          OMN   20
C ***** THIS IS THE MAIN OMNITAB ROUTINE *****          OMN   30
COMMON /BLOCKA/ MODE,M,KARD(83),KARG,ARG,ARG2,NEWCD(80),KRDEND          OMN   40
COMMON /BLOCKB/ NSTMT,NSTMTH,NCOM,LCOM,IOVFL,COM(2000)          OMN   50
COMMON /BLOCKC/ KIO,INUNIT,ISCRAT,KBDOUT,KRDKNT,LLIST          OMN   60
COMMON /BLOCRC/ NRC,RC(12600)          OMN   70
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NOMN  80
IARGS,VWXYZ(8),NERROR          OMN   90
DIMENSION ARGS(100)          OMN  100
EQUIVALENCE (ARGS(1),RC(12501))          OMN  110
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG          OMN  120
COMMON /HEADER/NOCARD(80),ITLE(60,6),LNCNT,IPRINT,NPAGE,IPUNCH          OMN  130
C THE FOLLOWING CARDS ARE NEEDED ONLY FOR TAPE OPERATIONS          OMN  150
COMMON /TAPE/ NAME4(2),NTPCT,IPUNCP,INUNIP,L1TP          OMN  160
C ****          OMN  170
DATA IBLANK/1H/,LETSGO/-1/          OMN  180
C          OMN  190
C THIS IS THE MAIN OMNITAB PROGRAM          OMN  200
C          OMN  210
C          OMN  220
C SUBROUTINES CALLED BY THIS PROGRAM..          OMN  230
C SETUP,INPUT,ERROR,STMT,NNAME,AARGS,ASTER,SETQ,READQ,STORE,XECUTE          OMN  240
C AERR,XOMNIT,XFORMAT,LOOKUP          OMN  250
C          OMN  260
C          OMN  270
C MOD = 1  INTERPRETIVE MODE          OMN  280
C     = 2  DATA MODE (READ SET)          OMN  290
C     = 3  STORAGE MODE (BETWEEN BEGIN AND FINISH)          OMN  300
C     =4  IMPLIED STORAGE MODE (STATEMENT NUMBER GIVEN)          OMN  310
C          OMN  320
C          OMN  330
C 0 = 0, 1 = 1, ETC., 9 = 9, A = 10, B = 11, ETC, Z= 35, / = 36          OMN  340
C . = 37, - = 38, + = 39, * = 40, ( = 41, ) = 42, , = 43          OMN  350
C BLANK = 44, = = 45, $ AND OTHERS = 46          OMN  360
C          OMN  370
CALL SETUP          OMN  460
10 IF (MODE.EQ.3) NSTMT=NSTMT+10          OMN  470
IF (MODE.EQ.4) MODE=1          OMN  480
NAME(1)=0          OMN  490
NAME(2)=0          OMN  500
NAME(3)=0          OMN  510
NAME(4)=0          OMN  520
NARGS=0          OMN  530
J=0          OMN  540
C          OMN  550
C CHECK FOR ACCUMULATED ERRORS DURING LAST EXECUTED COMMAND          OMN  560
C          OMN  570
CALL AERR (0)          OMN  580
CALL INPUT          OMN  590
C          OMN  600
C SCANNING BEGINS WITH THE THIRD CHARACTER. THE FIRST TWO ARE DUMMY          OMN  610
C TO KEEP THE PROGRAM OUT OF TROUBLE. SCANNING TERMINATES WITH A $          OMN  620
C A $ HAS BEEN PLANTED IN THE (KRDEND+1)-TH POSITION.          OMN  630
C          OMN  640
M=2          OMN  650
30 M=M+1          OMN  660
K=KARD(M)          OMN  670
IF (K.GE.36) IF (K-46) 45,40,30          OMN  675

```

```

IF (K.GE.10) GO TO 60                                     OMN 680
C
C   A NUMBER IS THE FIRST ALPHANUMERIC CHARACTER ENCOUNTERED, ERROR IF OMN 690
C   IN MODE 3                                              OMN 700
C
C   CALL OUTPUT                                            OMN 710
C   IF (MODE.NE.3) GO TO 50                                OMN 720
35  CALL ERROR (2)                                         OMN 730
      GO TO 20                                             OMN 740
40  IF (MODE.NE.4) CALL OUTPUT                            OMN 750
      GO TO 10                                             OMN 760
C   CHECK FOR * OR '                                       OMN 770
45  IF (K-40) 30,190,30                                    OMN 780
50  CALL STMT (NSTMT)                                     OMN 790
      IF (KARG.NE.0) IF (MODE-2) 35,185,35                OMN 800
C
C   IF AN ILLEGAL STATEMENT NUMBER WAS FOUND, KARG = 1 (KARG = 0 IF OMN 810
C   LEGAL)                                                 OMN 820
C
C   MODE=4                                                 OMN 830
C
C   M IS POINTING AT THE FIRST LETTER ON THE CARD, ASSEMBLE NAME. OMN 840
C
60  CALL NNAME (NAME(1))                                  OMN 850
C
C
C   CHECK THE FIRST NAME FOR SPECIAL NAMES...             OMN 860
C   OMNITAB, FORMAT, NOTE, FOOTNOTE, HEAD,TITLE          OMN 870
C
C   OMNITAB                                              OMN 880
C
C   IF (NAME(1).NE.11300.OR.NAME(2).NE.7102)IF(LETSGO) 65,67,67 OMN 890
C
C   IF NOT THE FIRST OMNITAB CARD, WRITE EOF RECORD.     OMN 900
C
C   IF(LETSGO.NE.(-1)) WRITE (ISCRAT,390)                 OMN 910
65  LETSGO=LETSGO+1                                      OMN 920
CALL XOMNIT (LETSGO)                                     OMN 930
IF(LETSGO.NE.(-1)) GO TO 10                             OMN 940
LETSGO=0                                                 OMN 950
C
C   FINISH                                                OMN 960
C
C
67  IF (NAME(1).NE.4631.OR.NAME(2).NE.7082) GO TO 70    OMN 970
      MODE=1                                              OMN 980
      GO TO 40                                             OMN 990
C
C   FORMAT                                               OMN1000
C
C   IF (MODE.NE.4) CALL OUTPUT                            OMN1010
70  IF (NAME(1).NE.4797.OR.NAME(2).NE.9524) GO TO 90    OMN1020
      CALL XFORMAT                                         OMN1030
C
80  IF (MODE.GE.3) CALL ERROR (202)                      OMN1040
      IF (MODE.NE.3) MODE=1                               OMN1050
      GO TO 10                                             OMN1060
C
C   NOTE                                                 OMN1070
C
C
90  IF (NAME(1).NE.10631.OR.NAME(2).NE.3645) GO TO 100  OMN1080
      OMN1090
      OMN1100
      OMN1110
      OMN1120
      OMN1130
      OMN1140
      OMN1150
      OMN1160
      OMN1170
      OMN1180
      OMN1190
      OMN1200
      OMN1210
      OMN1220
      OMN1230
      OMN1240

```

```

K=KARD(M)                                     OMN1243
IF (K.EQ.1.OR.K.EQ.2) GO TO 95               OMN1245
IF (NPAGE.EQ.0) CALL PAGE (0)                 OMN1247
WRITE (IPRINT,400) (NEWCD(I-2),I=M,82)        OMN1250
LNCNT=LNCNT+1                                OMN1260
GO TO 80                                      OMN1270
95    CALL NOTEPR (K)                         OMN1273
GO TO 80                                      OMN1275
C
C     HEAD                                     OMN1280
C
C     TITLES.   TITLEX = TITLE5, TITLEY = TITLE6 OMN1290
C
100   IF (NAME(1).NE.5968.OR.NAME(2).NE.2916) GO TO 110 OMN1300
CALL XHEAD                                     OMN1310
GO TO 80                                      OMN1320
C
C     TITLES.   TITLEX = TITLE5, TITLEY = TITLE6 OMN1330
C
110   IF (NAME(1).NE.14843) GO TO 160          OMN1340
C     CHECK NAME TITLE                         OMN1350
IF (NAME(2).EQ.8883) GO TO 120               OMN1360
C     CHECK TITLEX, TITLEY                     OMN1370
K=5                                           OMN1380
M=M+1                                         OMN1390
IF (NAME(2).NE.8908) IF (NAME(2)-8907) 160,130,160 OMN1400
K=6                                           OMN1410
GO TO 130                                     OMN1420
120   K=KARD(M)                               OMN1430
IF (K.GE.1.AND.K.LE.4) GO TO 130             OMN1440
CALL ERROR (209)                            OMN1450
K=1                                           OMN1460
130   MM=MIN0(M+59,81)                        OMN1470
DO 140 I=1,60                                OMN1480
140   ITLE(I,K)=IBLANK                         OMN1490
I=1                                           OMN1500
DO 150 MA=M,MM                                OMN1510
ITLE(I,K)=NEWCD(MA-1)                         OMN1520
150   I=I+1                                    OMN1530
GO TO 80                                      OMN1540
C
C     STOP                                     OMN1550
C
160   IF (NAME(1).NE.14406.OR.NAME(2).NE.11664) GO TO 170 OMN1560
WRITE (ISCRAT,390)                           OMN1570
CALL XSTOP                                     OMN1580
STOP                                         OMN1590
C
C     M IS POINTING AT THE FIRST NON-LETTER AFTER NAME. LOOK FOR OMN1600
POSSIBLE NAME QUALIFIER OR ARGUMENTS OR END OF CARD. OMN1610
C
170   K=KARD(M)                               OMN1620
IF (K.LT.36) IF (K-10) 190,175,175           OMN1630
IF (K.EQ.40) GO TO 190                         OMN1640
IF (K.EQ.46) GO TO 320                         OMN1650
M=M+1                                         OMN1660
GO TO 170                                     OMN1670
C
C     A LETTER FOUND, ASSEMBLE SECOND NAME (COMMAND QUALIFIER). OMN1680
C
175   CALL NNAME (NAME(3))                    OMN1690
                                              OMN1700
                                              OMN1710
                                              OMN1720
                                              OMN1730
                                              OMN1740
                                              OMN1750
                                              OMN1760

```

```

C          OMN1770
C          CHECK SPECIAL CASE OF NAMES M(XAX'), M(X'AX), M(XX'), M(X'X)  OMN1780
C          OMN1790
C          SKIP ONE CHARACTER (') IF FIRST NAME =(M )  OMN1800
C          THE FOLLOWING CARD IS NEEDED ONLY FOR TAPE OPERATIONS  OMN1810
C          IS NAME(3) EQUAL TO TAP AND NAME(4)=E  OMN1820
C          IF (NAME(3).NE.14623.OR.NAME(4).NE.3645) GO TO 180  OMN1830
C          CALL TAPOP  OMN1840
C          GO TO 190  OMN1850
C          ****  OMN1860
180      IF (NAME(1).EQ.9477) M=M+1  OMN1870
C          GO TO 190  OMN1880
C          OMN1890
C          SCAN FOR ARGUMENTS AND END OF CARD  OMN1900
C          OMN1910
185      M=3  OMN1920
190      J=J+1  OMN1930
C          GO TO 210  OMN1940
200      M=M+1  OMN1950
210      K=KARD(M)  OMN1960
C          IF (K.GE.10) IF (K-40) 200,255,315  OMN1965
C          OMN1970
C          NUMBER FOUND, CONVERT ARGUMENT. IF KARG RETURNED = 0, NUMBER IS  OMN1980
C          INTEGER, IF KARG = 1, NUMBER IS FLOATING POINT, IF KARG = -1, ERROROMN1990
C          OMN2000
C          CALL AARGS  OMN2010
C          IF (KARG) 10,230,220  OMN2020
220      ARGTAB(J)=0.  OMN2030
C          J=J+1  OMN2040
C          GO TO 240  OMN2050
C          OMN2060
C          ARGUMENT IS AN INTEGER. ADD A BIAS OF 8192 THEN CHECK THAT IT IS  OMN2070
C          .GT. 0  OMN2080
C          OMN2090
230      ARG=ARG+8192.  OMN2100
C          IF (ARG.GT.0.) GO TO 240  OMN2110
C          CALL ERROR (18)  OMN2120
C          GO TO 10  OMN2130
240      ARGTAB(J)=ARG  OMN2140
250      NARGS=NARGS+1  OMN2150
C          GO TO 190  OMN2160
C          OMN2170
C          ASTERISK FOUND, CONVERT  OMN2180
C          OMN2190
C          IF BRACKETED BY SINGLE ASTERISKS, QUANTITY IS TO BE USED AS A  OMN2200
C          FLOATING POINT ARGUMENT. IF BRACKETED BY DOUBLE ASTERISKS, QUANTITYOMN2210
C          IS TO BE TRUNCATED AND USED AS AN INTEGER ARGUMENT.  OMN2220
C          OMN2230
255      KARG=1  OMN2240
C          M=M+1  OMN2250
C          IF (KARD(M).NE.40) GO TO 260  OMN2260
C          KARG=0  OMN2270
C          M=M+1  OMN2280
260      MS=M  OMN2290
C          CALL ASTER  OMN2300
C          OMN2310
C          THE TERMINAL ASTERISK(S) HAVE BEEN CHECKED TO BE THE SAME AS THE  OMN2320
C          INTITAL SET (IF NO ERROR) AND M IS POINTING AT THE FIRST CHARACTEROMN2330
C          AFTER THE LAST ASTERISK.  OMN2340

```

C KARG RETURNED AS 1 = ERROR FOUND OMN2350  
 C 2 = FLOATING POINT CONSTANT, Z.B. \*PI\* OMN2360  
 C 3 = INTEGER NAMED VARIABLE, Z.B. \*\*NRMAX\*\* OMN2370  
 C 4 = FL. PT. NAMED VARIABLE, Z.B. \*NRMAX\* OMN2380  
 C 5 = INTEGER ROW-COLUMN, Z.B. \*\*3,40\*\* OMN2390  
 C 6 = FL. PT. ROW-COLUMN, Z.B. \*1,2\* OMN2400  
 C 7 = STRING OF ASTERISKS Z.B. \*\*\* OMN2410  
 C OMN2420  
 C A STRING OF THREE OR MORE ASTERISKS IMPLIES -THRU- OMN2430  
 C EXAMPLE.. OMN2440  
 C ERASE 1 2 3 4 12 13 14 15 16 20 IS EQUIVALENT TO OMN2450  
 C ERASE 1 \*\*\* 4, 12 \*\*\* 16, 20 OMN2460  
 C OMN2470  
 C PRINT 1 20 19 18 17 16 15 14 IS EQUIVALENT TO OMN2480  
 C PRINT 1, 20 \*\*\* 14 OMN2490  
 C OMN2500  
 C OMN2510  
 C OMN2520  
 GO TO (270,220,280,280,290,290,300), KARG OMN2530  
 270 M=MS OMN2540  
 GO TO 210 OMN2550  
 280 ARGTAB(J)=-2.\*ARG-FLOAT(KARG-3) OMN2560  
 GO TO 250 OMN2570  
 290 ARGTAB(J)==-(ARG+8208.) OMN2580  
 ARG2=ARG2+8192. OMN2590  
 IF (KARG.EQ.6) ARG2=-ARG2 OMN2600  
 J=J+1 OMN2610  
 ARGTAB(J)=ARG2 OMN2620  
 GO TO 250 OMN2630  
 300 IF (J.GT.0) GO TO 310 OMN2640  
 CALL ERROR (211) OMN2650  
 GO TO 210 OMN2660  
 310 ARGTAB(J)=-1. OMN2670  
 GO TO 190 OMN2680  
 C OMN2690  
 C OMN2700  
 C ARGTAB SETUP OMN2710  
 C OMN2720  
 C IF ENTRY .GT. 0, IT IS AN INTEGER CONSTANT (Z.B. COLUMN NUMBER) OMN2730  
 C TO WHICH A BIAS OF 8192 HAS BEEN ADDED. THIS IS TO SAY THAT A OMN2740  
 C NEGATIVE INTEGER ARGUMENT MAY NOT BE EXPLICITLY GIVEN OR MODIFIED OMN2750  
 C TO BE LESS THAN -8191. OMN2760  
 C OMN2770  
 C IF ENTRY .EQ.0, THE NEXT ENTRY IS A FLOATING POINT CONSTANT. OMN2780  
 C OMN2790  
 C IF ENTRY .LT. 0, ARGUMENT IS A VARIABLE. SET SIGN POSITIVE AND. OMN2800  
 C OMN2810  
 C IF ENTRY .LT. 16, IT IS A NAMED VARIABLE REFERENCE NUMBER OMN2820  
 C OMN2830  
 C 2,3 NRMAX 6,7 V 10,11 X OMN2840  
 C IF 4,5 COLTOP 8,9 W 12,13 Y OMN2850  
 C 14,15 Z OMN2860  
 C OMN2870  
 C OMN2880  
 C V,W,X,Y,Z, ARE FOR PROGRAMMING CONVENIENCE ONLY AND DO NOT OMN2890  
 C AFFECT THE OPERATION OF OMNITAB OMN2900  
 C OMN2910  
 C IF ENTRY IS EVEN, CURRENT VALUE TO BE TRUNCATED AND USED OMN2920  
 C AS AN INTEGER ARGUMENT. OMN2930

```

C           IF ENTRY IS ODD. THE CURRENT VALUE IS TO BE USED AS A      OMN2940
C           FLOATING POINT ARGUMENT.                                OMN2950
C
C           IF ENTRY .GT. 16, IT IS A WORKSHEET REFERENCE (ROW,COLUMN) TO OMN2970
C               WHICH A BIAS OF 8192. HAS BEEN ADDED.                  OMN2980
C               ENTRY - 8208 = ROW NUMBER                            OMN2990
C               ABS(NEXT ENTRY) = COLUMN NUMBER TO WHICH A BIAS OF 8192. OMN3000
C                   HAS BEEN ADDED.                                OMN3010
C
C           IF NEXT ENTRY IS NEGATIVE, WORKSHEET CONTENTS ARE TO BE OMN3020
C           USED AS A FLOATING POINT CONSTANT. IF +, WORKSHEET VALUE OMN3040
C           TO BE TRUNCATED AND USED AS AN INTEGER ARGUMENT.        OMN3050
C
C
C   315    IF (K.NE.46) GO TO 200                                OMN3060
C
C   THE TERMINATION OF CARD FOUND ( $ ENCONTERED)                OMN3070
C
C   320    IF (J.EQ.0) J=1                                      OMN3100
C           IF (MODE.NE.2.OR.NAME(1).NE.0) GO TO 350            OMN3110
C
C           IN INPUT MODE AND NO POSSIBLE NAME, RETURN TO SET OR READ ROUTINE OMN3140
C
C   330    CALL EXPAND (J,ARGTAB)                                OMN3150
C           IF (ISRFLG.EQ.0) GO TO 340                          OMN3160
C           CALL SETQ
C           GO TO 10
C   340    CALL READQ
C           GO TO 10
C
C           LOOK UP NAME (AND POSSIBLE QUALIFIER) IN DICTIONARY. RETURN OMN3230
C           COORDINATES OF ENTRY. IF L1 = 0, NAME NOT FOUND          OMN3240
C
C   350    CALL LOOKUP
C           IF (L1.NE.0) GO TO 360
C           IF (MODE.EQ.2) GO TO 330
C           CALL ERROR (1)
C           GO TO 10
C
C           NAME FOUND
C
C           THE FOLLOWING CARDS ARE NEEDED ONLY FOR TAPE OPERATIONS OMN3350
C           STATEMENT WAS 220 IF (MODE.EQ.2) MODE=1                  OMN3360
C   360    IF (MODE.EQ.2) GO TO 370
C           ****
C           IF (MODE.EQ.1) GO TO 380
C           CALL STORE (J)
C           GO TO 10
C           THE FOLLOWING CARDS ARE NEEDED ONLY FOR TAPE OPERATIONS OMN3420
C   370    MODE=1
C           INUNIT=INUNIP
C           ****
C   380    CALL EXPAND (J,ARGTAB)                                OMN3460
C           CALL XECUTE
C           GO TO 10
C
C   390    FORMAT (1HZ,83X)                                     OMN3490
C   400    FORMAT (1X,80A1)                                     OMN3500
C           END                                              OMN3510
C

```

```

SUBROUTINE ONEWAY                               ONE 10
VERSION 5.00 ONEWAY   5/15/70                  ONE 20
C                                              ONE 30
C ****ONEWAY STAT. ANALYSIS.                   ONE 40
C WRITTEN BY DAVID HOGBEN, SEL, NBS. 10/25/69.    ONE 50
C ONEWAY ANALYSIS OF DATA IN ++ WITH TAG IN ++ STORE IN ++,++,...  ONE 70
C TAG NUMBERS DIFFERENTIATE BETWEEN GROUPS      ONE 80
C WHEN TAG IS NON-POSITIVE ZERO WEIGHT IS GIVEN TO MEASUREMENTS  ONE 90
C NUMBER OF GROUPS MUST BE GREATER THAN 1 AND MUST NOT EXCEED NLNTH2  ONE 100
C NRMAX MUST NOT EXCEED NLNTH1                 ONE 110
C SLOPE IN ANOVA IS ONLY GIVEN IF FPROB FOR BETWEEN IS LESS THAN .100  ONE 120
C ****ONE 130
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NONE 140
1ARGS,VWXYZ(8),NERROR                         ONE 150
COMMON /BLOCRC/ NRC,RC(12600)                  ONE 160
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG            ONE 170
DIMENSION ARGS(100)                           ONE 180
EQUIVALENCE (ARGS(1),RC(12501))               ONE 190
COMMON /HEADER/ NOCARD(80),ITLE(60,6),LNCNT,IPRINT,NPAGE,IPUNCH  ONE 200
COMMON /SCRAT/ NS,NS2,A(13500)                 ONE 210
COMMON /ABCDEF/ L(48)                          ONE 220
EQUIVALENCE (BLANK,L(45)), (SLO,L(22)), (HIGH,L(18))  ONE 230
NLNTH1 = LENGTH OF ARRAYS = 2700, MUST BE CHANGED IF NS CHANGED  ONE 240
C 5*NLNTH1 MUST BE LE NS           DIMENSION A3(NLNTH1)  ONE 250
C DIMENSION A2(2700), A3(2700), A4(2700), A5(2700)  ONE 260
C EQUIVALENCE (A2(1),A(2701))                ONE 270
C EQUIVALENCE (A3(1),A(5401)), (A4(1),A(8101)), (A5(1),A(10801))  ONE 280
C NLNTH2 = LENGTH OF ARRAYS = 540, MUST BE CHANGED IF NS CHANGED  ONE 290
C 10*NLNTH2 LE NLNTH1           DIMENSION B1(NLNTH2)  ONE 300
C DIMENSION B1(540), B2(540), B3(540), B4(540), B5(540), B6(540), B7ONE 310
1(540), B8(540), B9(540), B10(540)          ONE 320
C EQUIVALENCE (B1(1),A(1)), (B2(1),A(541)), (B3(1),A(1081)), (B4(1),ONE 330
1A(1621)), (B5(1),A(2161)), (B6(1),A(2701)), (B7(1),A(3241)), (B8(10NE 340
2),A(3781)), (B9(1),A(4321)), (B10(1),A(4861))          ONE 350
C                                         ONE 360
C EXECUTION TIME CAN BE CONSIDERABLY SHORTENED USING LESS ACCURATE  ONE 370
C VERSION OF FPPT.                            ONE 380
C                                         ONE 390
C ****ONE 400
C                                         ONE 410
C NLNTH1=NS/5                                ONE 420
C NLNTH2=NLNTH1/5                            ONE 430
C ERROR CHECKING                            ONE 440
IF (NRMAX.GT.NLNTH1) GO TO 50                ONE 450
IF (NRMAX.EQ.0) CALL ERROR (9)                ONE 460
IF (NARGS.EQ.6) GO TO 40                    ONE 470
IF (NARGS.EQ.2.AND.L2.EQ.13) GO TO 40        ONE 480
IF (NARGS.EQ.2.AND.L2.EQ.14) GO TO 10        ONE 490
IF (NARGS.EQ.3) GO TO 20                    ONE 500
CALL ERROR (10)                            ONE 510
RETURN                                     ONE 520
10 CALL ERROR (236)                          ONE 530
RETURN                                     ONE 540
20 DO 30 I=4,6                            ONE 550
IARGS(I)=IARGS(3)+1                        ONE 560
30 KIND(I)=0                            ONE 570
NARGS=6                            ONE 580
40 CALL CHKCOL (J)                        ONE 590

```

```

      IF (J.EQ.0) GO TO 60                               ONE 600
50   CALL ERROR (11)                                 ONE 610
60   IF (NERROR.NE.0) GO TO 510                      ONE 620
C   MOVE Y AND TAG TO SCRATCH AREA MOVING TO BOTTOM IF TAG      ONE 630
C   NON-POSITIVE, CONVERT TAG TO INTEGER, COMPUTE NZW AND K, SET UP I  ONE 640
C   M11=IARGS(1)-1                                     ONE 650
C   M12=IARGS(2)-1                                     ONE 660
K=0
NZW=0
DO 80 I=1,NRMAX                                     ONE 680
M02=M11+I                                         ONE 700
M03=M12+I                                         ONE 710
IF (RC(M03).GE.1.0) GO TO 70                      ONE 720
NZW=NZW+1                                         ONE 730
M01=NRMAX-NZW+1                                    ONE 740
A3(M01)=RC(M02)                                    ONE 750
A4(M01)=0.0                                         ONE 760
GO TO 80                                           ONE 770
70   J=I-NZW                                         ONE 780
A3(J)=RC(M02)                                      ONE 790
A4(J)=AINT(RC(M03)+1.E-8)                         ONE 800
80   K=MAX0(K,INT(A4(J)+1.0E-6))                   ONE 810
NZW=NRMAX-NZW                                      ONE 820
IF (NZW.LE.K) GO TO 50                           ONE 830
IF (K.LT.2) GO TO 50                           ONE 840
IF (K.GT.NLNTH2) GO TO 50                        ONE 850
M34=NZW+1                                         ONE 860
C   COMPUTE NI,MEAN,S(R),SETUP MIN + MAX,IBAR, FOR I=1,K      ONE 870
DO 90 I=1,NLNTH1                                    ONE 880
90   A(I)=0.0                                         ONE 890
CALL RANKO (NZW,A3(1),A2(1),A5(1),A(49))        ONE 900
A(49)=12.0*A(49)                                    ONE 910
A(133)=NZW                                         ONE 920
DO 100 I=1,NZW                                      ONE 930
M40=A4(I)                                         ONE 940
B2(M40)=B2(M40)+1.0                                ONE 950
B3(M40)=B3(M40)+A3(I)                            ONE 960
B5(M40)=B5(M40)+A5(I)                            ONE 970
B6(M40)=A3(I)                                       ONE 980
B7(M40)=A3(I)                                       ONE 990
A(21)=A(21)+A3(I)                                 ONE1000
100  A(101)=A(101)+A4(I)                           ONE1010
A(21)=A(21)/A(133)                                ONE1020
A(101)=A(101)/A(133)                                ONE1030
DO 110 I=1,K                                       ONE1040
IF (B2(I).GT.0.0) B3(I)=B3(I)/B2(I)                ONE1050
B8(I)=B3(I)                                         ONE1060
C   COMPUTE MIN,MAX,SD,S,SS,DF,MS,F,FPROB,S(1/NI),S(NI**3),S(R**2/NONE1070
110  B9(I)=B2(I)                                     ONE1080
DO 120 I=1,NZW                                      ONE1090
M40=A4(I)+.0001                                     ONE1100
B6(M40)=AMIN1(B6(M40),A3(I))                     ONE1110
B7(M40)=AMAX1(B7(M40),A3(I))                     ONE1120
B4(M40)=B4(M40)+(A3(I)-B3(M40))**2            ONE1130
A(1)=A(1)+(B3(M40)-A(21))**2                    ONE1140
A(4)=A(4)+(A3(I)-B3(M40))**2                    ONE1150
120  A(5)=A(5)+(A3(I)-A(21))**2                  ONE1160
A(17)=0.0                                         ONE1170
A(22)=B3(1)                                         ONE1180

```

```

A(23)=B3(1)                                     ONE1190
A(24)=FSQRT(B4(1))                            ONE1200
A(25)=0.                                         ONE1210
A(26)=B6(1)                                     ONE1220
A(27)=B7(1)                                     ONE1230
A(48)=0.0                                       ONE1240
DO 150 I=1,K                                    ONE1250
B10(I)=(B2(I)*(B2(I)-1.0))/B4(I)             ONE1260
A(126)=A(126)+B10(I)*B3(I)                   ONE1270
A(127)=A(127)+B10(I)                         ONE1280
IF (B2(I)-1.) 150,140,130                     ONE1290
130  B4(I)=FSQRT(B4(I)/(B2(I)-1.0))           ONE1300
A(121)=A(121)+(B2(I)-1.0)*FLOG(B4(I)*B4(I)) ONE1310
A(131)=A(131)+1.                             ONE1320
A(25)=AMAX1(A(25),B4(I))                     ONE1330
A(24)=AMIN1(A(24),B4(I))                     ONE1340
A(120)=A(120)+1.0/(B2(I)-1.0)                ONE1350
140  A(2)=A(2)+B2(I)*(FLOAT(I)-A(101))*(B3(I)-A(21)) ONE1360
A(114)=A(114)+B2(I)*((FLOAT(I)-A(101))**2)  ONE1370
A(22)=AMIN1(A(22),B3(I))                     ONE1380
A(23)=AMAX1(A(23),B3(I))                     ONE1390
A(26)=AMIN1(A(26),B6(I))                     ONE1400
A(27)=AMAX1(A(27),B7(I))                     ONE1410
A(17)=A(17)+B5(I)**2/B2(I)                   ONE1420
A(18)=A(18)+1./B2(I)                         ONE1430
A(48)=A(48)+(B3(I)-A(21))**2                ONE1440
A(122)=A(122)+B4(I)**2                       ONE1450
A(129)=A(129)+B2(I)**2                       ONE1460
150  A(118)=A(118)+B2(I)**3                   ONE1470
A(126)=A(126)/A(127)                         ONE1480
A(2)=A(2)**2/A(114)                           ONE1490
A(3)=A(1)-A(2)                                ONE1500
C   DEGREES OF FREEDOM FOR ANOVA
M1=K-1                                         ONE1510
A(136)=FLOAT(M1)                             ONE1520
M2=1                                           ONE1540
M3=K-2                                         ONE1550
M4=NZW-K                                      ONE1560
A(134)=FLOAT(M4)                             ONE1570
M5=NZW-1                                      ONE1580
C   MEAN SQUARES
A(6)=A(1)/A(136)                            ONE1600
A(7)=A(2)/FLOAT(M2)                          ONE1610
A(8)=A(3)/FLOAT(M3)                          ONE1620
A(9)=A(4)/FLOAT(M4)                          ONE1630
A(10)=A(5)/FLOAT(M5)                         ONE1640
A(11)=A(6)/A(9)                             ONE1650
A(12)=A(7)/((A(3)+A(4))/(A(133)-2.))    ONE1660
A(13)=A(8)/A(9)                             ONE1670
CALL PROB (A(136),FLOAT(M4),A(11),A(14))  ONE1680
CALL PROB (FLOAT(M2),FLOAT(M4),A(12),A(15))  ONE1690
CALL PROB (FLOAT(M3),FLOAT(NZW-2),A(13),A(16)) ONE1700
C   COMPUTE FOR KRUSKAL-WALLIS TEST
A(117)=Nzw*(Nzw+1)                           ONE1720
A(17)=(12.*A(17))/A(117)-3.*FLOAT(Nzw+1)    ONE1730
A(102)=1.0-A(49)/FLOAT(Nzw**3-Nzw)           ONE1740
A(17)=A(17)/A(102)                           ONE1750
A(106)=(FLOAT(Nzw**3)-A(118))/A(117)         ONE1760
A(105)=FLOAT(2*M1)-.4*FLOAT(3*K*M3+Nzw*(2*K*(K-3)+1))/A(117)-6.*A(ONE1770

```

```

118)/5.                                     ONE1780
A(103)=A(136)*(A(136)*(A(106)-A(136))-A(105))/(.5*A(105)*A(106))  ONE1790
A(104)=(A(106)-A(136))*A(103)/A(136)          ONE1800
A(19)=A(17)*(A(106)-A(136))/(A(136)*(A(106)-A(17)))           ONE1810
CALL PROB (AINT(A(103)+.5),AINT(A(104)+.5),A(19),A(20))           ONE1820
C COMPUTE TOTAL STATISTICS                  ONE1830
A(31)=FSQRT(A(9))                         ONE1840
A(32)=FSQRT(A(48)/A(136))                 ONE1850
A(33)=FSQRT(A(10))                        ONE1860
A(34)=A(31)/FSQRT(A(133))                 ONE1870
A(35)=A(32)/FSQRT(FLOAT(K))              ONE1880
A(36)=A(33)/FSQRT(A(133))                 ONE1890
CALL TPCTPT (FLOAT(M4),A(37))             ONE1900
CALL TPCTPT (A(136),A(38))                ONE1910
CALL TPCTPT (FLOAT(M5),A(39))             ONE1920
A(41)=A(21)-A(34)*A(37)                  ONE1930
A(42)=A(21)-A(35)*A(38)                  ONE1940
A(43)=A(21)-A(36)*A(39)                  ONE1950
A(44)=A(21)+A(34)*A(37)                  ONE1960
A(45)=A(21)+A(35)*A(38)                  ONE1970
A(46)=A(21)+A(36)*A(39)                  ONE1980
C SORT XBAR FOR MULTIPLE COMPARISONS OF MEANS   ONE1990
DO 160 I=1,M1                           ONE2000
M37=K-I                                ONE2010
DO 160 J=1,M37                          ONE2020
IF (B8(J).LE.B8(J+1)) GO TO 160        ONE2030
A(113)=B8(J)                           ONE2040
B8(J)=B8(J+1)                          ONE2050
B8(J+1)=A(113)                         ONE2060
A(113)=B9(J)                           ONE2070
B9(J)=B9(J+1)                          ONE2080
B9(J+1)=A(113)                         ONE2090
160 CONTINUE                            ONE2100
CALL FPPT (A(136),FLOAT(M4),.05,A(115))  ONE2110
A(116)=A(31)*FSQRT(A(136)*A(115))      ONE2120
C TESTS FOR HOMOGENEITY OF VARIANCES       ONE2130
A(51)=(A(25)*A(25))/A(122)              ONE2140
A(123) = AINT (A(133)/FLOAT(K)-0.5)     ONE2150
CALL PROB (A(123),A(123)*(A(131)-1.), (A(131)-1.)*A(51)/(1.-A(51)),ONE2155
1 A(52))                                ONE2160
A(52) = (A(131)-1.)*A(52)               ONE2165
IF (A(52).GT.1.) A(52)=1.0              ONE2170
A(57)=(A(25)/A(24))**2                 ONE2180
A(121)=A(134)*FLOG(A(9))-A(121)       ONE2190
A(124)=(A(120)-1.0/A(134))/(3.*A(136))  ONE2200
A(125)=(A(131)+1.)/(A(124)*A(124))    ONE2210
A(53)=(A(125)*A(121))/((A(131)-1.0)*(A(125)/(1.0-A(124)+2.0/A(125)ONE2220
1)-A(121)))                            ONE2230
CALL PROB (A(131)-1.0,AINT(A(125)+.5),A(53),A(54))      ONE2240
A(130)=(A(133)-A(129)/A(133))/A(136)      ONE2250
A(47)=(A(6)-A(9))/A(130)                  ONE2260
C COMPUTATIONS ARE NOW COMPLETE *****ONE2270
M0=L2-12                                ONE2280
GO TO (170,480), M0                      ONE2290
C AUTOMATIC PRINTING WHEN L2=13          ONE2300
C FORMAT STATEMENTS                     ONE2310
170 CALL PAGE (4)                         ONE2320
C PRINT ANOVA                           ONE2330
WRITE (IPRINT,520)                       ONE2340

```

```

      WRITE (IPRINT,530) M1,A(1),A(6),A(11),A(14)          ONE2350
      IF (K.LT.3) GO TO 180                                ONE2360
      IF (A(14).GE..10) GO TO 180                           ONE2370
      WRITE (IPRINT,540) M2,A(2),A(7),A(12),A(15)          ONE2380
      WRITE (IPRINT,550) M3,A(3),A(8),A(13),A(16)          ONE2390
180   WRITE (IPRINT,560) M4,A(4),A(9)                      ONE2400
      WRITE (IPRINT,570) M5,A(5)                          ONE2410
C     PRINT KRUSKAL-WALLIS TEST                         ONE2420
      WRITE (IPRINT,580) A(17),A(20)                      ONE2430
C     PRINT ESTIMATES                                 ONE2440
      WRITE (IPRINT,590)                               ONE2450
      DO 220 I=1,K                                     ONE2460
      A(107)=BLANK                                    ONE2470
      A(108)=BLANK                                    ONE2480
      IF (B2(I)-1.0) 220,190,190                      ONE2490
190   IF (B3(I).LE.A(22)) A(107)=SLO                ONE2500
      IF (B3(I).GE.A(23)) A(107)=HIGH               ONE2510
      IF (B4(I).LE.A(24)) A(108)=SLO                ONE2520
      IF (B4(I).GE.A(25)) A(108)=HIGH               ONE2530
      M8=B2(I)                                       ONE2540
      IF (M8-1) 220,200,210                           ONE2550
200   WRITE (IPRINT,610) I,M8,B3(I),A(107),B6(I),B7(I),B5(I)  ONE2560
      GO TO 220                                      ONE2570
210   A(109)=B4(I)/FSQRT(B2(I))                  ONE2580
      CALL TPCTPT (FLOAT(M8-1),A(112))            ONE2590
      A(110)=B3(I)-A(109)*A(112)                 ONE2600
      A(111)=B3(I)+A(109)*A(112)                 ONE2610
      WRITE (IPRINT,600) I,M8,B3(I),A(107),B4(I),A(108),A(109),B6(I),B7(  ONE2620
      1I),B5(I),A(110),A(111)                     ONE2630
220   CONTINUE                                     ONE2640
      WRITE (IPRINT,620) NZW,A(21),A(26),A(27),A(31),A(34),A(41),A(44),A(  ONE2650
      1(32),A(35),A(42),A(45),A(33),A(36),A(43),A(46)  ONE2660
C     COMPUTE AND PRINT FOR MULTIPLE COMPARISONS        ONE2670
      IF (A(14)-0.10) 230,450,450                  ONE2680
230   IF (M4.LT.4) GO TO 340                      ONE2690
      WRITE (IPRINT,630)                            ONE2700
C     NEWMAN-KEULS-HARTLEY                         ONE2710
      WRITE (IPRINT,640)                            ONE2720
      J=1                                         ONE2730
      M28=0                                       ONE2740
240   I=K                                         ONE2750
250   IF (I-M28) 330,330,260                      ONE2760
260   IF (I.EQ.J) GO TO 280                      ONE2770
      A(135)=ABS(B8(I)-B8(J))                  ONE2780
C     MANDEL APPROXIMATION TO PERCENT POINT OF STUDENTIZED RANGE  ONE2790
      A(137)=I-J+1                                ONE2800
      RX=-.283917+2.63532*(A(134)-1.00123)**(-.95862)  ONE2810
      U1=-.314115+2.38301*(A(134)-1.03428)**(-.864005)  ONE2820
      U2=3.65961*U1**2-1.00891*U1-0.166346           ONE2830
      C=2.3849867-2.9051857*(A(137)-0.57583164)**(-.069648109)  ONE2840
      V1=1.30153-1.95073*(A(137)+.394915)**(-.139783)  ONE2850
      V2=4.72863*V1**2+0.404271*V1-0.135104           ONE2860
      A(119)=6.15075+4.441409*RX+6.7514569*C+7.4671282*U1*V1-.157537*U2*  ONE2870
      1V2                                         ONE2880
      A(119)=A(119)*FSQRT(.5*(1.0/B9(I)+1.0/B9(J)))*A(31)  ONE2890
      IF (A(135)-A(119)) 280,280,270               ONE2900
270   I=I-1                                       ONE2910
      GO TO 250                                     ONE2920
280   IF (J.EQ.1) GO TO 310                      ONE2930

```

```

      IF (J-M28) 290,290,300          ONE2940
290  WRITE (IPRINT,670)              ONE2950
      GO TO 310                      ONE2960
300  WRITE (IPRINT,680)              ONE2970
310  WRITE (IPRINT,660) (B8(M29),M29=J,I)  ONE2980
      IF (I-K) 320,340,340          ONE2990
320  M28=I                          ONE3000
330  J=J+1                          ONE3010
      GO TO 240                      ONE3020
C   SCHEFFE METHOD                 ONE3030
340  WRITE (IPRINT,650)              ONE3040
      J=1                            ONE3050
      M28=0                          ONE3060
350  I=K                            ONE3070
360  IF (I-M28) 440,440,370        ONE3080
370  IF (I.EQ.J) GO TO 390          ONE3090
      A(135)=ABS(B8(I)-B8(J))       ONE3100
      A(119)=A(116)*FSQRT(1.0/B9(I)+1.0/B9(J))  ONE3110
      IF (A(135)-A(119)) 390,390,380  ONE3120
380  I=I-1                          ONE3130
      GO TO 360                      ONE3140
390  IF (J.EQ.1) GO TO 420          ONE3150
      IF (J-M28) 400,400,410        ONE3160
400  WRITE (IPRINT,670)              ONE3170
      GO TO 420                      ONE3180
410  WRITE (IPRINT,680)              ONE3190
420  WRITE (IPRINT,660) (B8(M29),M29=J,I)  ONE3200
      IF (I-K) 430,450,450          ONE3210
430  M28=I                          ONE3220
440  J=J+1                          ONE3230
      GO TO 350                      ONE3240
450  IF (A(131).LT.2.0) GO TO 480  ONE3250
      WRITE (IPRINT,690) A(51),A(52),A(53),A(54),A(57)  ONE3260
      IF (A(52).GT..10.AND.A(54).GT..10) GO TO 470  ONE3270
      DO 460 I=1,K                  ONE3280
      A(55)=A(55)+B10(I)*(B3(I)-A(126))**2  ONE3290
460  A(128)=A(128)+(1.-B10(I)/A(127))**2/(B2(I)-1.)  ONE3300
      A(128)=(FLOAT(K**2)-1.)/(3.*A(128))  ONE3310
      A(55)=(A(55)/FLOAT(M1))/(1.+(2.*FLOAT(M3))/(3.*A(128)))  ONE3320
      CALL PROB (FLOAT(M1),AINT(A(128)+.5),A(55),A(56))  ONE3330
      WRITE (IPRINT,700) A(55),A(56)          ONE3340
470  WRITE (IPRINT,710) A(47)          ONE3350
C   AUTOMATIC PRINTOUT IS FINISHED - NOW STORE RESULTS *****
480  C
      IF (NARGS.EQ.2) GO TO 510          ONE3360
      C
      M13=IARGS(3)                      ONE3370
      M14=IARGS(4)                      ONE3380
      M15=IARGS(5)                      ONE3390
      M16=IARGS(6)                      ONE3400
      DO 500 I=1,K                      ONE3410
      C   DONT STORE IF N=0            ONE3420
      IF (B2(I).LE.0.0) GO TO 490        ONE3430
      C   TAG                         ONE3440
      RC(M13)=I                        ONE3450
      C   N                           ONE3460
      RC(M14)=B2(I)                    ONE3470
      C   XBAR                        ONE3480
      RC(M15)=B3(I)                    ONE3490
      C   STANDARD DEVIATION          ONE3500

```

```

        RC(M16)=B4(I)          ONE3530
490      M13=M13+1           ONE3540
        M14=M14+1           ONE3550
        M15=M15+1           ONE3560
500      M16=M16+1           ONE3570
510      RETURN             ONE3580
C
C
520      FORMAT (//,48X,20HANALYSIS OF VARIANCE//17X,6HSOURCE,14X,4HD.F.,4XONE3610
1,14HSUM OF SQUARES,5X,12HMEAN SQUARES,9X,7HF RATIO,4X,7HF PROB./) ONE3620
530      FORMAT (17X,14HBETWEEN GROUPS,5X,I4,1P2E18.6,4X,0PF11.3,F10.3)   ONE3630
540      FORMAT (20X,5HSLOPE,14X,I4,1P2E18.6,3X,0PF11.3,F10.3)       ONE3640
550      FORMAT (20X,16HDEVS. ABOUT LINE,3X,I4,1P2E18.6,3X,0PF11.3,F10.3) ONE3650
560      FORMAT (17X,13HWITHIN GROUPS,6X,I4,1P2E18.6)                  ONE3660
570      FORMAT (17X,5HTOTAL,14X,I4,1PE18.6//)                      ONE3670
580      FORMAT (11X,65HKRUSKAL-WALLIS RANK TEST FOR DIFFERENCE BETWEEN GROONE3680
1UP MEANS * H =,0PF9.3,10H, F PROB =,F6.3,10H (APPROX.//)          ONE3690
590      FORMAT (55X,9HESTIMATES//1X,5HGROUP,5X,3HNO.,6X,4HMEAN,7X,11HWITHIONE3700
1N S.D.,2X,12HS.D. OF MEAN,5X,7HMINIMUM,7X,7HMAXIMUM,6X,4HS(R),4X,2ONE3710
23H95PCT CONF INT FOR MEAN/)                                     ONE3720
600      FORMAT (1X,I4,I8,1PE14.5,A1,E13.5,A1,E13.5,2E14.5,0PF9.1,1PE13.5,3ONE3730
1H TO,E12.5)                                         ONE3740
610      FORMAT (1X,I4,I8,1PE14.5,A1,3X,24H ESTIMATE NOT AVAILABLE ,1P2E14.ONE3750
15,0PF9.1,3X,25H***** TO *****)                                ONE3760
620      FORMAT (/,1X,5HTOTAL,I7,1PE14.5,28X,2E14.5/7X,20HFIXED EFFECTS MODONE3770
1EL ,2E14.5,37X,E13.5,3H TO,E12.5/7X,20HRANDOM EFFECTS MODEL,,2E14.ONE3780
25,37X,E13.5,3H TO,E12.5/7X,14HUNGROUPED DATA,6X,2E14.5,37X,E13.5,3ONE3790
3H TO,E12.5/)                                              ONE3800
630      FORMAT (1X,120HPAIRWISE MULTIPLE COMPARISON OF MEANS. THE MEANS AONE3810
1RE PUT IN INCREASING ORDER IN GROUPS SEPARATED BY *****. A MEAN IONE3820
2S /120H ADJUDGED NON-SIGNIFICANTLY DIFFERENT FROM ANY MEAN IN THEONE3830
3 SAME GROUP AND SIGNIFICANTLY DIFFERENT AT THE .05 LEVEL FROM /1200ONE3840
4H ANY MEAN IN ANOTHER GROUP. ***** ***** INDICATES ADJACENT GROUPONE3850
5S HAVE NO COMMON MEAN. )                                     ONE3860
640      FORMAT (/3X,89HNEWMAN-KEULS TECHNIQUE, HARTLEY MODIFICATION. (APPRONE3870
10XIMATE IF GROUP NUMBERS ARE UNEQUAL.))                      ONE3880
650      FORMAT (/3X,18HSCHEFFE TECHNIQUE.)                      ONE3890
660      FORMAT (3X,9(1PE12.5,1H,))                           ONE3900
670      FORMAT (6X,5H*****)                                 ONE3910
680      FORMAT (3X,11H***** *****)                         ONE3920
690      FORMAT (/,36H TESTS FOR HOMOGENEITY OF VARIANCES./7X,13HCOCHRAN,S ONE3930
1C =31H MAX. VARIANCE/SUM(VARIANCES) =,F6.4,5H, P =,F6.3,10H (APPROONE3940
2X.)/7X,16HBARTLETT-BOX F =,F9.3,5H, P =,F6.3/7X,37HMAXIMUM VARIANCEONE3950
3E / MINIMUM VARIANCE =,F10.3)                               ONE3960
700      FORMAT (7X,70HAPPROX BETWEEN MEANS F-TEST IN PRESENCE OF HETEROGENONE3970
1EOUS VARIANCE. F =,F8.3,5H, P =,F6.3)                      ONE3980
710      FORMAT (/,1X,35HMODEL II - COMPONENTS OF VARIANCE. /7X,29HESTIMATEONE3990
1 OF BETWEEN COMPONENT,1PE15.7)                            ONE4000
    END                                                 ONE4010

```

```

SUBROUTINE OPONE (N,M,MX,NX,ND2,ND3,ND19,B,SSQ,IX)          OPO 10
C VERSION 5.00      OPONE      5/15/70                      OPO 20
C SUBROUTINE TO PRINT PAGE 1 OF POLYFIT AND FIT             OPO 30
C *****
C WRITTEN BY DAVID HOGBEN, SEL, NBS.   10/14/69.            OPO 40
COMMON/BLOCRC/NRC,RC(12600)                                OPO 60
COMMON/BLOCKE/NAME(4),L1,L2,ISRFLG                         OPO 70
COMMON/HEADER/NOCARD(80),ITLE(60,6),LNCNT,IPRINT,NPAGE,IPUNCH OPO 80
COMMON/SCRAT/NS,NS2,A(13500)                                OPO 90
COMMON/KFMT/KFMT(100)                                     OPO 100
COMMON/FMAT/IFMTX(6),IOSWT,IFMTS(6),LHEAD(96)              OPO 110
COMMON/BLOCKD/IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,
1NARGS,VWXYZ(8),NERROR                                    OPO 120
DIMENSION ARGS(100),IIRGS(100)                            OPO 130
EQUIVALENCE (ARGS(1),RC(12501)),(IIRGS(1),KFMT(1))        OPO 140
DIMENSION B(1)                                              OPO 150
C *****
C *****
C *****
C *****
IF (L2.EQ.1) GO TO 1620                                     OPO 160
IF (MX.GT.1) GO TO 1640                                     OPO 170
1620 WRITE (IPRINT,1625) (LHEAD(I1),I1=13,24),(LHEAD(I2),I2=1,12),
1 IARGS(IX),IARGS(1)                                         OPO 180
1625 FORMAT (/5X,2(4X,12A1,4X),6X,9HPREDICTED,8X,12HSTD. DEV. OF,25X,
1 4HSTD./2X,3HROW,3X, 9HIN COLUMN,I5,6X, 9HIN COLUMN,I5,10X,6HVALUEOPO 260
2S,10X,12HPRED. VALUES,8X,9HRESIDUALS,8X,4HRES.,3X,7HWEIGHTS/) OPO 270
GO TO 1660                                                 OPO 280
1640 IF (MX.GT.2) GO TO 1650                                 OPO 290
  WRITE (IPRINT,1645) (LHEAD(I),I=1,12),IARGS(NX+4),IARGS(NX+5),
1 IARGS(1)                                              OPO 300
1645 FORMAT (/8X,22HPREDICTOR VARIABLES IN,6X,12A1,8X,9HPREDICTED,6X,
1 12HSTD. DEV. OF,22X,4HSTD./2X,3HROW,3X,4HCOL.,I4,6X,4HCOL.,I4,8X,OPO 330
24HCOL.,I4,11X,6HVALUES,8X,12HPRED. VALUES,6X,9HRESIDUALS,7X,4HRES.OPO 340
3,3X,7HWEIGHTS/)                                         OPO 350
GO TO 1660                                                 OPO 360
1650 WRITE (IPRINT,1655) (LHEAD(I),I=1,12),IARGS(NX+4),IARGS(NX+5),
1 IARGS(NX+6),IARGS(1)                                       OPO 370
1655 FORMAT (/12X,22HPREDICTOR VARIABLES IN,9X,12A1,6X,9HPREDICTED,4X
1,12HSTD. DEV. OF,19X,4HSTD./2X,3HROW,2X,3(4HCOL.,I4,4X),2X,4HCOL.,OPO 400
2I4,9X,6HVALUES,6X,12HPRED. VALUES,4X,9HRESIDUALS,6X,4HRES.,3X,
37HWEIGHTS/)                                             OPO 410
GO TO 1660                                                 OPO 420
1660 IX = IIRGS(IX)                                         OPO 430
  IY = IIRGS(1)                                           OPO 440
  IND3 = ND3+1                                           OPO 450
  IND2 = ND2+1                                           OPO 460
  IND4 = ND19+1                                           OPO 470
  LL = 0                                                 OPO 480
  NSD = 8                                                 OPO 490
  NWM = 18                                               OPO 500
  IF (L2.EQ.1 .OR. MX.EQ.1) GO TO 1666                  OPO 510
  IX2 = IIRGS(NX+5)                                       OPO 520
  NWM = 2*(2/MX)                                         OPO 530
  LL = 4-NWM                                            OPO 540
  NSD = 4+NWM                                           OPO 550
  NWM = NWM+10                                          OPO 560
  CALL RFORMAT (RC(IX2),N,NSD,NW2,NDEC2,NWM,A(1),A(1),0,0) OPO 570
  IF (MX.EQ.2) GO TO 1666                               OPO 580
  IX3 = IIRGS(NX+6)                                       OPO 590

```

```

1666 CALL RFORMAT (RC(IX3),N,NSD,NW3,NDEC3,NWM,A(1),A(1),0,0) OPO 600
      CALL RFORMAT (RC(IX ),N,NSD,NW1,NDEC1, NWM ,A(1),A(1),0,0) OPO 610
      CALL RFORMAT (RC(IY ),N, 8,NW4,NDEC4,18-LL,A(1),A(1),0,0) OPO 620
      CALL RFORMAT (A(IND3),N, 8,NW5,NDEC5,17-LL,A(1),A(1),0,0) OPO 630
      CALL RFORMAT (A(IND2),N, 8,NW6,NDEC6,17-LL,A(1),A(1),0,0) OPO 640
      CALL RFORMAT (A(IND4),N, 8,NW7,NDEC7,17-LL,A(1),A(1),0,0) OPO 650
      IF (KIND(2).EQ.1) GO TO 1667 OPO 660
      IW = IIRGS(2) OPO 670
      CALL RFORMAT (RC(IW),N,4,NW9,NDEC9,9,A(1),A(1),0,0) OPO 680
      GO TO 1680 OPO 690
1667 CALL RFORMAT (ARGS(2),1,4,NW9,NDEC9,9,A(1),A(1),0,0) OPO 700
      CALL RFORMAT (A(1),1,4,NW9,NDEC9,0,ARGS(2),B(98),11-NW9,1) OPO 710
      WT=ARGS(2) OPO 715
1670 FORMAT (1X,I4,97A1,0PF7.2,11A1) OPO 720
1680 DO 1745 I=1,N OPO 730
      CALL RFORMAT (A(1),1,NSD,NW1,NDEC1,0,RC(IX ),B(1),NWM+2-NW1,1) OPO 740
      IF (L2.EQ.1 .OR. MX.EQ.1) GO TO 1685 OPO 750
      CALL RFORMAT (A(1),1,NSD,NW2,NDEC2,0,RC(IX2),B(NWM+3),NWM+2-NW2,1) OPO 760
      IX2 = IX2+1 OPO 770
      IF (MX.EQ.2) GO TO 1685 OPO 780
      CALL RFORMAT (A(1),1,NSD,NW3,NDEC3,0,RC(IX3),B(25),12-NW3,1) OPO 790
      IX3 = IX3+1 OPO 800
1685 CALL RFORMAT (A(1),1,8,NW4,NDEC4,0,RC(IY),B(4*LL+21),20-LL-NW4,1) OPO 810
      CALL RFORMAT (A(1),1,8,NW5,NDEC5,0,A(IND3),B(3*LL+41),19-LL-NW5,1) OPO 820
      CALL RFORMAT (A(1),1,8,NW6,NDEC6,0,A(IND2),B(2*LL+60),19-LL-NW6,1) OPO 830
      CALL RFORMAT (A(1),1,8,NW7,NDEC7,0,A(IND4),B(LL + 79),19-LL-NW7,1) OPO 840
      IF (KIND(2).EQ.1) GO TO 1730 OPO 850
      CALL RFORMAT (A(1),1,4,NW9,NDEC9,0,RC(IW),B(98),11-NW9,1) OPO 860
      WT=RC(IW) OPO 865
      IW = IW+1 OPO 870
1730 IF(WT.GT.0.0) STDRES=A(IND4)/FSQRT(SSQ/WT-A(IND2)**2) OPO 875
      IF(WT.LE.0.0) STDRES=0.0 OPO 880
      WRITE (IPRINT,1670) I,(B(II),II=1,97),STDRES,(B(II),II=98,108) OPO 890
1740 IX = IX+1 OPO 900
      IY = IY+1 OPO 910
      IND3 = IND3+1 OPO 920
      IND2 = IND2+1 OPO 930
1745 IND4 = IND4+1 OPO 940
      RETURN OPO 950
      END OPO 960

```

```

SUBROUTINE ORTHO          ORT 10
C VERSION 5.00      ORTHO      5/15/70      ORT 20
C DOUBLE PRECISION FDSQRT,DK2,SUM,YSUM      ORT 40
C COMMON /ABCDEF/ L(48)      ORT 50
C COMMON /BLOCRC/ NRC,RC(12600)      ORT 60
C COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NORT 70
IARGS,VWXYZ(8),NERROR      ORT 80
C DIMENSION ARGS(100)      ORT 90
C EQUIVALENCE (ARGS(1),RC(12501))      ORT 100
C COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG      ORT 110
C COMMON /HEADER/ NOCARD(80),ITLE(60,6),LNCNT,IPRINT,NPAGE,IPUNCH      ORT 120
C COMMON /SCRAT/ NS,NS2,A(13500)      ORT 130
C COMMON /KFMT/ KFMT(100)      ORT 140
C COMMON /FMAT/ IFMTX(6),IOSWT,IFMTS(6),LHEAD(96)      ORT 150
C DIMENSION IIRGS(100)      ORT 160
C EQUIVALENCE (IIRGS(1),KFMT(1)), (B(1),IB)      ORT 170
C DIMENSION IMTRXA(2,3)      ORT 180
C DIMENSION B(120), IHC(4), IHT(8)      ORT 190
C DATA IHC(1),IHC(2),IHC(3),IHC(4)/3H TE,3HRM ,3HCOL,3HUMN/      ORT 200
C *** *****      ORT 210
C ORTHONORMALIZATION PROGRAM BY PHILIP J. WALSH JULY 1, 1967      ORT 220
C REVISED BY S. PEAVY 5/28/68      ORT 230
C REVISED BY DAVID HOGBEN AND SALLY PEAVY, SEL, NBS. 9/23/69.      ORT 240
C LEAST SQUARES PROGRAM USING GRAM SCHMIDT PROCESS      ORT 250
C      ORT 260
C POLYFIT Y IN COL ++ WITH WEIGHTS (ALL EQUAL TO **)      ORT 270
C             (IN COL++)      ORT 280
C USING POLYNOMIAL OF DEGREE , TO X IN COL ++      ORT 290
C STORE: COEFFICIENTS IN COL ++      ORT 300
C             DEVIATIONS IN COL ++      ORT 310
C             STANDARD DEV OF PREDICTED VALUES IN COL ++      ORT 320
C             FOURIER COEFFICIENTS IN COL ++      ORT 330
C             VARIANCE COVARIANCE MATRIX STARTING IN (,,++)      ORT 340
C ONLY FIRST 4 ARGUMENTS MUST BE SPECIFIED.      ORT 350
C STORAGE WILL TAKE PLACE FOR ONLY THE STORAGE ARGUMENTS      ORT 360
C PROVIDED      ORT 370
C             L2=1 POLYFIT: L2=2 SPOLYFIT      ORT 380
C      ORT 390
C FIT Y IN COL ++ WITH WEIGHTS(ALL EQUAL TO **) AS A      ORT 400
C             (IN COL++)      ORT 410
C FUNCTION OF K=++ VARIABLES IN COLS ++,++,++,...,++      ORT 420
C STORE: COEFFICIENTS IN COL ++      ORT 430
C             DEVIATIONS IN COL ++      ORT 440
C             STANDARD DEV OF PREDICTED VALUES IN COL ++      ORT 450
C             FOURIER COEFFICIENTS, ETC. IN COL ++      ORT 460
C             VARIANCE COVARIANCE MATRIX STARTING IN (++,,)      ORT 470
C MINIMUM OF 4 ARGUMENTS IS NEEDED BEFORE COMMAND IS EXEC.      ORT 480
C             L2=3 FIT      : L2=4 SFIT      ORT 490
C      ORT 500
C MORTHO X (,,++) R=,, C=,, WITH WEIGHTS ( ALL EQUAL TO **)      ORT 510
C             ( IN COL++)      ORT 520
C STORE IN M (,,++) A MATRIX IN (,,++)      ORT 530
C             L2=5 MORTHO:      ORT 540
C      ORT 550
C      ORT 560
C IX1(I,J,IN)=IN+(I*(I-1))/2+J      ORT 570
C PRECHECKING SECTION      ORT 580
C IF (L2.EQ.1.OR.L2.EQ.3) GO TO 8      ORT 582
C IF (L2.EQ.2) IF (NARGS-4) 8,6,8      ORT 584

```

```

      IF (NARGS.NE.IARGS(3)+3) GO TO 8          ORT 586
6     CALL ERROR (236)                         ORT 588
      RETURN                                     ORT 590
8     IREFIT=0                                 ORT 595
      IF (NARGS.GT.0) GO TO 10                  ORT 600
      CALL ERROR (10)                           ORT 610
      RETURN                                     ORT 620
10    IF (NRMAX.NE.0) GO TO 20                  ORT 630
      CALL ERROR (9)                           ORT 640
      RETURN                                     ORT 650
20    NMUI=1                                    ORT 660
C     IF L2 =5 THEN COMMAND IS MORTHO        ORT 670
      IF (L2.EQ.5) GO TO 1850                  ORT 680
      IF (NARGS.LT.4) GO TO 1830                  ORT 690
C     COMMAND IS POLYFIT OR FIT              ORT 700
      CALL ADRESS (1,IIRGS(1))                 ORT 710
      IF (IIRGS(1).LE.0) CALL ERROR (11)         ORT 720
      IF (KIND(2).EQ.1) GO TO 30                  ORT 730
      CALL ADRESS (2,IIRGS(2))                 ORT 740
      IF (IIRGS(2).LE.0) CALL ERROR (11)         ORT 750
      GO TO 40                                  ORT 760
30    SU=NRMAX                                ORT 770
      WSUM=SU                                  ORT 780
      IF (ARGS(2).LE.0.0) CALL ERROR (24)         ORT 790
      NMUI=2                                    ORT 800
40    NST=1                                    ORT 810
      IF (KIND(3).EQ.1) IARGS(3)=ARGS(3)         ORT 815
      NEND=NARGS                                ORT 820
      J=NARGS-4                                ORT 825
      IF (L2.GT.2) J=J-IARGS(3)+1                ORT 830
      IF (J.LE.4.AND.J.GE.0) GO TO 50            ORT 835
      IF (J.NE.6) GO TO 1830                      ORT 837
45    NEND=NARGS-2                            ORT 840
      NST=2                                    ORT 850
50    DO 60 I=4,NEND                          ORT 860
      CALL ADRESS (I,IIRGS(I))                 ORT 870
      IF (IIRGS(I).LE.0) CALL ERROR (11)         ORT 880
60    CONTINUE                                 ORT 890
      M=IARGS(3)                                ORT 900
      IF (L2.LE.2) M=M+1                          ORT 910
      N=NRMAX                                   ORT 920
      FN=N                                      ORT 930
      GO TO (100,70), NST                        ORT 940
70    CALL ADRESS (NARGS,IST)                  ORT 950
      IF (IST.GT.0) GO TO 80                      ORT 960
      CALL ERROR (11)                           ORT 970
      RETURN                                     ORT 980
80    MMTXR=M                                 ORT 990
      MMTXC=M                                 ORT 1000
      IST=IST-1+IARGS(NARGS-1)                  ORT 1010
      IF (IARGS(NARGS)+M-1.GT.NCOL) MMTXC=NCOL-IARGS(NARGS)+1 ORT 1020
      IF (IARGS(NARGS-1)+M-1.GT.NROW) MMTXR=NROW-IARGS(NARGS-1)+1 ORT 1030
      IF (MMTXR.GT.0) GO TO 90                  ORT 1040
      NARGS=NARGS-2                            ORT 1050
      CALL ERROR (213)                           ORT 1060
      GO TO 100                                 ORT 1070
90    IF (MMTXR.NE.M.OR.MMTXC.NE.M) CALL ERROR (213) ORT 1080
100   GO TO (110,140), NMUI                   ORT 1090
110   SU=0.0                                    ORT 1100

```

```

WSUM=0.0                                     ORT1110
L22=IIRGS(2)                                 ORT1120
L22A=L22                                     ORT1130
DO 130 I=1,N                                 ORT1140
IF (RC(L22A)) 1840,130,120                 ORT1150
120   SU=SU+1.0                               ORT1160
      WSUM=WSUM+RC(L22A)                      ORT1170
130   L22A=L22A+1                            ORT1180
140   FM=M                                    ORT1190
      IF (SU-FM) 150,160,170                  ORT1200
150   CALL ERROR (24)                         ORT1210
      RETURN                                  ORT1220
160   DENOM=1.0                                ORT1230
      GO TO 180                                ORT1240
170   DENOM=FSQRT(SU-FM)                      ORT1250
180   NPM=N+M                                 ORT1260
      M1=M-1                                 ORT1270
      M2=M+1                                 ORT1280
      N1=N-1                                 ORT1290
      N2=N+1                                 ORT1300
      MD1=(M*(M2))/2                          ORT1310
C
C
      ND1=M2*NPM                            ORT1320
C   X REQUIRES ND1 CELLS                     ORT1330
C   GET A (ND1 + 1) FOR START OF PK          ORT1340
      ND2=M*NPM                             ORT1350
      MD3=ND2+N                            ORT1360
      ND3=ND1                                ORT1370
      ND3=ND1                                ORT1380
      ND3=ND1                                ORT1390
C   ADD NPM TO REACH XP                      ORT1400
      ND4=ND3+NPM                           ORT1410
C   ADD NPM TO REACH QK                      ORT1420
      ND5=ND4+NPM                           ORT1430
C   ADD (M+1) TO REACH CV                   ORT1440
      ND6=ND5+M2                            ORT1450
C   ADD (M*(M+1))/2 + M TO REACH VCV        ORT1460
      ND66=MD1+M                           ORT1470
      ND7=ND6+ND66                           ORT1480
C   ADD THE SAME AMOUNT TO REACH Q          ORT1490
      ND8=ND7+ND66                           ORT1500
C   Q IS (M+1) CELLS LONG THEN COMES Q2     ORT1510
      ND9=ND8+M2                            ORT1520
C   Q2 E AND EP ARE EACH M CELLS LONG       ORT1530
      ND10=ND9+M                           ORT1540
      ND11=ND10+M                           ORT1550
      ND12=ND11+M                           ORT1560
C   THE A MATRIX IS NEXT                     ORT1570
      ND13=ND12+MD1                           ORT1580
C   GRAM FACTOR STORAGE                     ORT1590
      ND14=ND13+M2                           ORT1600
C   ENF                                     ORT1610
C   CV DIAGONALS                           ORT1620
      ND16=ND14+M                           ORT1630
C   VCV DIAGONALS                           ORT1640
      ND17=ND16+M                           ORT1650
      ND18=ND17+M                           ORT1660
      ND19=ND18+NPM                          ORT1670
      ND20=ND19+N                           ORT1680
      IF (IREFIT.EQ.1) GO TO 190            ORT1690

```

```

      IF (ND20.GT.NS) CALL ERROR (23)          ORT1700
      IF (NERROR.NE.0) RETURN                  ORT1710
190   NRBAR=1                                ORT1720
      I=IIRGS(1)                            ORT1730
      L22A=L22                                ORT1732
      CONS3=RC(L22)                           ORT1733
      IF(KIND(2).EQ.1) CONS3=ARGS(2)           ORT1735
      CONS1=RC(I)                            ORT1736
      CONS2=RC(I)                            ORT1737
      DO 195  I1=1,NRMAX                      ORT1738
      IF(KIND(2).EQ.0) CONS3=RC(L22A)           ORT1740
      IF(CONS3) 192,192,191                   ORT1741
191   CONS=RC(I)                            ORT1742
      IF(CONS.LT.CONS1) CONS1=CONS            ORT1743
      IF(CONS.GT.CONS2) CONS2=CONS            ORT1745
192   I=I+1                                ORT1746
195   L22A=L22A+1                           ORT1747
      YCONS=(CONS2+CONS1)/2.0                 ORT1748
      GO TO (200,200,240,240,240), L2         ORT1750
C     THIS IS POLYFIT                      ORT1760
200   L33=IIRGS(4)                           ORT1770
      MXARGS=5                             ORT1780
      L33A=L33                                ORT1790
      K=NPM+1                               ORT1800
      DO 210  I=1,N                          ORT1810
      A(I)=1.0                             ORT1820
      A(K)=RC(L33A)                         ORT1830
      K=K+1                                ORT1840
210   L33A=L33A+1                           ORT1850
      IF (M.EQ.2) GO TO 320                 ORT1860
      DO 230  K=2,M1                         ORT1870
      L33A=L33                                ORT1880
      K2=K*NPM+1                           ORT1890
      K1=K2-NPM                            ORT1900
      DO 220  I=1,N                          ORT1910
      A(K2)=A(K1)*RC(L33A)                 ORT1920
      K2=K2+1                               ORT1930
      K1=K1+1                               ORT1940
220   L33A=L33A+1                           ORT1950
230   CONTINUE                            ORT1960
      GO TO 320                            ORT1970
240   I=4                                 ORT1980
C     FIND OUT IF ALL X(I) = 1.0, IF SO SET NX=1 AND MX=M-1. ORT1990
      L33=IIRGS(4)                           ORT2000
      NX=1                                ORT2010
      DO 250  NW6=1,N                        ORT2020
      IF (ABS(RC(L33)-1.0).LE.1.E-6) GO TO 250 ORT2030
      NX=0                                ORT2040
      GO TO 260                            ORT2050
250   L33=L33+1                           ORT2060
260   MX=MAX0(1,M-NX)                     ORT2070
      IF(NX.EQ.0.OR.L2.EQ.5) YCONS=0.0       ORT2075
      MXARGS=M+4                           ORT2080
      L44=MXARGS-1                         ORT2090
      J=1                                 ORT2100
270   DO 310  I1=I,L44                      ORT2150
      K1=J                                ORT2160
      L33=IIRGS(I1)                         ORT2170
      K2=K1                                ORT2180

```

```

DO 300 I2=1,N                                     ORT2190
290 A(K2)=RC(L33)                                ORT2220
      K2=K2+1                                     ORT2230
300 L33=L33+1                                    ORT2240
      IJKLM=2                                     ORT2250
310 J=J+NPM                                      ORT2260
C   GENERATE IDENTITY MATRIX AUGMENTATION        ORT2270
320 K1=N2                                         ORT2280
      DO 340 K=1,M                                ORT2290
      K2=K1                                         ORT2300
      DO 330 I=1,M                                ORT2310
      A(K2)=0.                                     ORT2320
330 K2=K2+1                                      ORT2330
      K2=K1+K-1                                   ORT2340
      K1=K1+NPM                                    ORT2350
340 A(K2)=1.0                                     ORT2360
C   BEGIN THE G.S. PROCESS                         ORT2370
      NBEI=1                                       ORT2380
      NRHI=1                                       ORT2390
      I18=1+ND13                                    ORT2400
      NGAI=2                                       ORT2410
      NSII=2                                       ORT2420
      NDEI=1                                       ORT2430
      NNUI=1                                       ORT2440
      LZ1=1                                         ORT2450
      LZ2=1                                         ORT2460
C   K CONTROLS WHOLE LOOP                         ORT2470
      K=1                                           ORT2480
350 NTHI=1                                         ORT2490
360 NALI=1                                         ORT2500
      NOMI=1                                       ORT2510
      NJ=ND3+N+1                                    ORT2520
      DO 370 J=1,M                                ORT2530
      A(NJ)=0.                                     ORT2540
370 NJ=NJ+1                                       ORT2550
C   BOX 6.                                         ORT2560
380 KD1=(K-1)*NPM                                 ORT2570
      I1=ND3+1                                     ORT2580
      I2=KD1+1                                     ORT2590
      L22A=L22                                       ORT2600
      DO 420 I=1,N                                ORT2610
      GO TO (390,400), NMUI                      ORT2620
C   PK(I)                                         ORT2630
390 A(I1)=A(I2)*RC(L22A)                        ORT2640
      L22A=L22A+1                                  ORT2650
      GO TO 410                                     ORT2660
400 A(I1)=A(I2)*ARGS(2)                         ORT2670
410 I1=I1+1                                       ORT2680
420 I2=I2+1                                       ORT2690
      GO TO (430,460), NOMI                      ORT2700
430 IA1=1                                         ORT2710
      IA2=ND5+1                                     ORT2720
      DO 450 I=1,K                                ORT2730
      I2=IA1                                         ORT2740
      SUM=0.0                                       ORT2750
      J2=ND3+1                                     ORT2760
      DO 440 J=1,NPM                            ORT2770
      SUM=SUM+A(J2)*A(I2)                         ORT2780
      I2=I2+1                                         ORT2790

```

```

440    J2=J2+1                               ORT2800
      C          QK(I)                         ORT2810
      A(IA2)=SUM                           ORT2820
      IA1=IA1+NPM                          ORT2830
450    IA2=IA2+1                           ORT2840
      GO TO 490                           ORT2850
460    DK2=0 .
      I1=(K-1)*NPM+1                      ORT2860
      IND3=ND3+1                           ORT2870
      DO 470 I=1,NPM                      ORT2880
      DK2=DK2+A(IND3)*A(I1)               ORT2890
      I1=I1+1                            ORT2900
470    IND3=IND3+1                         ORT2910
      DK=FDSQRT(DK2)                      ORT2920
      C          GRAM FACTORS              ORT2930
      A(I18)=DK                           ORT2940
      I18=I18+1                           ORT2950
      K1=(K-1)*NPM+1                      ORT2960
      DO 480 I=1,NPM                      ORT2970
      A(K1)=A(K1)/DK                     ORT2980
480    K1=K1+1                           ORT2990
      NOMI=1                            ORT3000
      GO TO 380                           ORT3010
      C          BOX8                         ORT3020
490    GO TO (500,560), NDEI             ORT3030
500    LZ1=-LZ1                           ORT3040
      IF (LZ1) 550,510,510               ORT3050
      C          BOX8A                      ORT3060
510    K1=K-1                           ORT3070
      IRUTH=ND5+1                         ORT3080
      DO 520 I=1,K1                      ORT3090
      A(IRUTH)=-A(IRUTH)                 ORT3100
520    IRUTH=IRUTH+1                      ORT3110
      IRUTH=K+ND5                         ORT3120
      A(IRUTH)=1.0                        ORT3130
      J2=ND4+1                           ORT3140
      DO 540 I=1,NPM                      ORT3150
      SUM=0.0                            ORT3160
      J1=I                                ORT3170
      J3=ND5+1                           ORT3180
      DO 530 J=1,K                        ORT3190
      SUM=SUM+A(J1)*A(J3)                ORT3200
      J1=J1+NPM                          ORT3210
      J3=J3+1                           ORT3220
530    J3=J3+1                           ORT3230
      C          XP(I)                      ORT3240
      A(J2)=SUM                          ORT3250
540    J2=J2+1                           ORT3260
      GO TO 640                           ORT3270
      C          BOX8B      GET QK(I18)   ORT3280
550    ISAL=I18+M2                         ORT3290
      IRUTH=ND5+K                         ORT3300
      A(ISAL)=FSQRT(A(IRUTH))           ORT3310
      GO TO 510                           ORT3320
      C          NDE1                      ORT3330
560    LZ2=-LZ2                           ORT3340
      IF (LZ2) 570,510,510               ORT3350
      C          GET E AMD OTHER VECTORS ORT3360
570    DO 580 I=1,M                      ORT3370
      IND5=ND5+I                          ORT3380

```

```

IND9=ND9+I                                     ORT3390
IND8=ND8+I                                     ORT3400
A(IND8)=A(IND5)                                ORT3410
580 A(IND9)=A(IND5)*A(IND5)                   ORT3420
A(IND8+1)=A(IND5+1)                            ORT3430
A(ND10+1)=A(IND8+1)-A(ND9+1)                  ORT3440
IND10=ND10+1                                    ORT3450
IND9=ND9+1                                      ORT3460
DO 590 J=2,M                                    ORT3470
IND10=IND10+1                                  ORT3480
IND9=IND9+1                                    ORT3490
590 A(IND10)=A(IND10-1)-A(IND9)               ORT3500
FI=1.0                                         ORT3510
IND10=ND10                                     ORT3520
IND11=ND11                                     ORT3530
DO 635 I=1,M                                    ORT3540
IND10=IND10+1                                  ORT3550
IND11=IND11+1                                  ORT3560
IF (FN-FI) 630,630,600                         ORT3570
600 IF (A(IND10)) 610,620,620                ORT3580
610 A(IND11)=FSQRT(ABS(A(IND10))/(FN-FI))    ORT3590
GO TO 635                                     ORT3600
620 A(IND11)=FSQRT(A(IND10)/(FN-FI))         ORT3610
GO TO 635                                     ORT3620
630 A(IND10)=-1.0                             ORT3630
635 FI=FI+1.0                                 ORT3640
GO TO 510                                     ORT3650
C      BOX9                                    ORT3660
640 GO TO (650,670,800), NTHI                 ORT3670
650 K1=(K-1)*NPM+1                           ORT3680
IND4=ND4+1                                    ORT3690
DO 660 I=1,NPM                               ORT3700
A(K1)=A(IND4)                                ORT3710
K1=K1+1                                      ORT3720
660 IND4=IND4+1                             ORT3730
GO TO 760                                     ORT3740
670 IND18=ND18+1                           ORT3750
IND4=ND4+1                                    ORT3760
DO 680 I=1,N                                    ORT3770
A(IND18)=A(IND4)                            ORT3780
IND18=IND18+1                                ORT3790
680 IND4=IND4+1                             ORT3800
NI=N+1                                       ORT3810
DO 690 I=1,M                                    ORT3820
KK1=ND18+NI                                 ORT3830
IND4=ND4+NI                                 ORT3840
A(KK1)=-A(IND4)                            ORT3850
690 NI=NI+1                                 ORT3860
IND4=ND4                                     ORT3870
IND19=ND19                                   ORT3880
DO 700 I=1,N                                    ORT3890
IND4=IND4+1                                 ORT3900
IND19=IND19+1                                ORT3910
700 A(IND19)=A(IND4)                            ORT3920
IF (L2.EQ.5.OR.IREFIT.EQ.1) GO TO 750       ORT3930
IF (NARGS-MXARGS) 750,730,710                ORT3940
710 L66=IIRGS(MXARGS+1)-1                     ORT3950
L66A=L66                                     ORT3960
IND4=ND4                                     ORT3970

```

```

DO 720 I=1,N
L66A=L66A+1
IND4=IND4+1
720 RC(L66A)=A(IND4)
730 L55=IIRGS(MXARGS)
L55A=L55
NI=N+ND4+1
RC(L55A)=-A(NI)+YCONS
IF (M.EQ.1) GO TO 750
DO 740 I=2,M
NI=NI+1
L55A=L55A+1
740 RC(L55A)=-A(NI)
750 NTHI=3
GO TO 650
C      BOX10
760 GO TO (770,780), NALI
770 NOMI=2
NALI=2
GO TO 380
780 IF (K-M) 790,830,830
790 K=K+1
GO TO 350
800 GO TO (810,820), NNUI
810 NNUI=2
GO TO 920
820 SS=DK/DENOM
SSQ=SS*SS
GO TO 920
830 GO TO (840,800), NBEI
C      ***** ***** ***** ***** ***** *****
C      GET THE A MATRIX
840 K1=1
DO 860 I=1,M
I1=I*N+(I-1)*M
DO 850 J=1,I
I2=J+I1
K2=K1+ND12
A(K2)=A(I2)
850 K1=K1+1
860 CONTINUE
C
NDEI=2
NBEI=2
NTHI=2
K=K+1
GO TO (800,870), NGAI
C      GET CV MATRIC
870 CONTINUE
DO 890 IL=1,M
LOC=IX1(IL,0,ND6)+1
DO 890 J=1,IL
SUM=0.
DO 880 KK=IL,M
LOC1=IX1(KK,IL,ND12)
LOC2=IX1(KK,J,ND12)
880 SUM=SUM+A(LOC1)*A(LOC2)
A(LOC)=SUM
890 LOC=LOC+1

```

```

J1=3+ND6                                     ORT4570
J=ND16+2                                     ORT4580
A(ND16+1)=FSQRT(A(ND6+1))                  ORT4590
IF (M.EQ.1) GO TO 910                      ORT4600
DO 900 I=2,M                                 ORT4610
C           THE ARGUMENT IN THE FOLLOWING SQRT OCCASIONALLY IS NEGATIVE   ORT4620
A(J)=FSQRT(A(J1))                           ORT4630
J=J+1                                       ORT4640
900 J1=J1+I+1                               ORT4650
910 NGA1=1                                  ORT4660
GO TO 800                                  ORT4670
920 GO TO (930,990), NRHI                   ORT4680
930 IF (NRBAR) 940,1030,940                 ORT4690
940 NRBAR=NRBAR-1                           ORT4700
NTHI=2                                      ORT4710
NRHI=2                                      ORT4720
L11=IIRGS(1)-1                            ORT4730
L11A=L11+1                                 ORT4740
I1=ND2+1                                    ORT4750
DO 970 I=1,N                                ORT4760
IF (L2.NE.5) GO TO 960                     ORT4770
GO TO (960,950), NMUI                     ORT4780
950 A(I1)=ARGS(2)                           ORT4790
GO TO 970                                  ORT4800
960 A(I1)=RC(L11A)-YCONS                 ORT4810
L11A=L11A+1                               ORT4820
970 I1=I1+1                                 ORT4830
I1=MD3+1                                   ORT4840
DO 980 I=1,M                                ORT4850
A(I1)=0.                                    ORT4860
980 I1=I1+1                                 ORT4870
GO TO 360                                  ORT4880
990 GO TO (930,1000), NSII                  ORT4890
C GET VCV AND DEV AND COEF                ORT4900
1000 IND7=ND7+1                            ORT4910
IND6=ND6+1                                 ORT4920
DO 1010 I=1,MD1                           ORT4930
A(IND7)=SSQ*A(IND6)                        ORT4940
IND7=IND7+1                               ORT4950
1010 IND6=IND6+1                           ORT4960
IND16=ND16+1                             ORT4970
IND17=ND17+1                             ORT4980
DO 1020 I=1,M                                ORT4990
A(IND17)=SS*A(IND16)                      ORT5000
IND16=IND16+1                           ORT5010
1020 IND17=IND17+1                           ORT5020
GO TO 930                                  ORT5030
C THE CALCULATIONS ARE COMPLETED. NOW OUTPUT THE RESULTS          ORT5040
1030 A(ND8+1)=A(ND8+1)+YCONS*FSQRT(WSUM)    ORT5050
A(ND9+1)=A(ND8+1)**2                      ORT5060
IND18=ND18+N+1                           ORT5070
A(IND18)=A(IND18)+YCONS                  ORT5080
IF (IREFIT.EQ.1) GO TO 1640               ORT5090
IF (L2.EQ.5) GO TO 1960                  ORT5100
IF (NARGS.LT.MXARGS) GO TO 1190          ORT5110
L55A=L55+M                                ORT5120
ISF=1                                      ORT5130
L55B=L55A+M-1                           ORT5140
IF (L55B.LE.NROW+L55-1) GO TO 1040        ORT5150

```

```

IF (M.GE.NROW) GO TO 1190          ORT5160
ISF=2                               ORT5170
L55B=L55+NROW-1                   ORT5180
1040 IND17=ND17+1                  ORT5190
DO 1050 I=L55A,L55B               ORT5200
RC(I)=A(IND17)                     ORT5210
1050 IND17=IND17+1                ORT5220
GO TO (1060,1190), ISF            ORT5230
1060 IMS=NROW-2*M                 ORT5240
IF (IMS.GT.6) IMS=6               ORT5250
GO TO (1180,1170,1160,1150,1140,1070), IMS
1070 L11A=IIRGS(1)                ORT5260
L22A=L22                           ORT5270
YBAR=0.0                            ORT5280
DO 1100 I=1,N                      ORT5290
GO TO (1080,1090), NMUI           ORT5300
1080 YBAR=YBAR+RC(L22A)*RC(L11A) ORT5310
L22A=L22A+1                        ORT5320
GO TO 1100                          ORT5330
1090 YBAR=YBAR+RC(L11A)           ORT5340
1100 L11A=L11A+1                  ORT5350
YBAR=YBAR/WSUM                      ORT5360
L11A=IIRGS(1)                     ORT5370
L22A=L22                           ORT5380
YYBAR=0.                            ORT5390
DO 1130 I=1,N                      ORT5400
GO TO (1110,1120), NMUI           ORT5410
1110 YYBAR=RC(L22A)*(RC(L11A)-YBAR)**2+YYBAR ORT5420
L22A=L22A+1                        ORT5430
GO TO 1130                          ORT5440
1120 YYBAR=ARGS(2)*(RC(L11A)-YBAR)**2+YYBAR ORT5450
1130 L11A=L11A+1                  ORT5460
R2=1.-SSQ*(SU-FM)/YYBAR          ORT5470
IF (R2.LT.0.) R2=0.0               ORT5480
IF (R2.GT.1.0) R2=1.0             ORT5490
RC(L55B+6)=R2                      ORT5500
1140 RC(L55B+5)=SS                 ORT5510
1150 RC(L55B+4)=SS                 ORT5520
1160 RC(L55B+3)=SU-FM              ORT5530
1170 RC(L55B+2)=FM                 ORT5540
1180 RC(L55B+1)=SU                 ORT5550
C COMPUTE PREDICTED VALUES S.D. OF PREDICTED
1190 IND2=ND2+1                  ORT5560
IND3=ND3+1                          ORT5570
IND4=ND19+1                        ORT5580
IPIC=1                             ORT5590
IY=IIRGS(1)                       ORT5600
YSUM=0.0                            ORT5610
L22A=L22                           ORT5620
DO 1230 I=1,N                      ORT5630
IP=IPIC                           ORT5640
SP=0.0                             ORT5650
DO 1200 II=1,M                     ORT5660
SP=SP+A(IP)**2                   ORT5670
1200 IP=IP+NPM                      ORT5680
A(IND2)=FSQRT(SP)*SS              ORT5690
IPIC=IPIC+1                        ORT5700
IND2=IND2+1                        ORT5710
A(IND3)=RC(IY)-A(IND4)           ORT5720
                                         ORT5730
                                         ORT5740

```

```

IND3=IND3+1                                     ORT5750
IND4=IND4+1                                     ORT5760
GO TO (1210,1220), NMUI                         ORT5770
1210 YSUM=YSUM+RC(IY)**2*RC(L22A)               ORT5780
L22A=L22A+1                                     ORT5790
GO TO 1230                                       ORT5800
1220 YSUM=YSUM+RC(IY)**2*ARGS(2)                ORT5810
1230 IY=IY+1                                     ORT5820
C   CHECK TO SEE IF RESULTS ARE TO BE PUT IN WORK SHEET ORT5830
    IF (NARGS.LE.MXARGS+1) GO TO 1460           ORT5840
    IF (NARGS-(MXARGS+3)) 1440,1390,1240        ORT5850
C   STORE VARIANCE COVARIANCE MATRIX IN WORK SHEET ORT5860
1240 LSWT=0                                       ORT5870
IND7=ND7+1                                       ORT5880
1250 ISTR=IST                                       ORT5890
ISTC=IST                                         ORT5900
MSTOP=MIN0(MMTXC,MMTXR)                         ORT5910
DO 1270 I=1,MSTOP                                ORT5920
ISTRRR=ISTRR                                      ORT5930
ISTCC=ISTC                                         ORT5940
DO 1260 J=1,I                                     ORT5950
RC(ISTRRR)=A(IND7)                               ORT5960
RC(ISTCC)=A(IND7)                               ORT5970
IND7=IND7+1                                      ORT5980
ISTRRR=ISTRRR+NROW                             ORT5990
1260 ISTCC=ISTCC+1                                ORT6000
ISTR=ISTR+1                                      ORT6010
1270 ISTC=ISTC+NROW                               ORT6020
    IF (MMTXC.EQ.MMTXR) GO TO 1330              ORT6030
C   VARIANCE COVARIANCE MATRIX IS STORED AS A RECTANGULAR MATRIX ORT6040
    MSTP=MAX0(MMTXC,MMTXR)-MSTOP                ORT6050
    IF (MMTXC-MMTXR) 1280,1280,1290             ORT6060
1280 ICONA=NROW                                    ORT6070
ICONB=1                                         ORT6080
ISTART=ISTR                                      ORT6090
GO TO 1300                                       ORT6100
1290 ICONA=1                                      ORT6110
ICONB=NROW                                      ORT6120
ISTART=ISTC                                      ORT6130
1300 DO 1320 I=1,MSTP                            ORT6140
ISTRC=ISTART                                     ORT6150
DO 1310 J=1,MSTOP                                ORT6160
RC(ISTRC)=A(IND7)                               ORT6170
IND7=IND7+1                                      ORT6180
1310 ISTRC=ISTRC+ICONA                           ORT6190
IND7=IND7+I                                      ORT6200
1320 ISTART=ISTART+ICONB                           ORT6210
C   STORE GRAM FACTORS, VECTOR NORMS AND GRAM DETERMINANTS ORT6220
1330 MSTOP=NROW-1-IARGS(NARGS-1)-M                ORT6230
    IF(MSTOP.LT.(-1)) GO TO 1380                ORT6240
IND13=ND13+1                                      ORT6250
IND14=ND14+1                                      ORT6260
GMDT=1.0                                         ORT6270
ISTR=IST+M                                       ORT6280
DO 1370 I=1,MMTXC                                ORT6290
    IF (MSTOP) 1360,1350,1340                  ORT6300
1340 GMDT=GMDT*(A(IND13)/A(IND14))**2            ORT6310
    RC(ISTR+2)=GMDT                            ORT6320
1350 RC(ISTR)=A(IND13)                           ORT6330

```

```

IND13=IND13+1                                     ORT6340
1360 RC(ISTR+1)=A(IND14)                         ORT6350
IND14=IND14+1                                     ORT6360
1370 ISTR=ISTR+NROW                             ORT6370
C STORE FOURIER COEFFICIENTS                  ORT6380
1380 IF (LSWT.EQ.1) RETURN                      ORT6390
1390 LST=IIRGS(MXARGS+3)                        ORT6400
IND9=ND9                                         ORT6410
DO 1400 I=1,M                                     ORT6420
IND9=IND9+1                                      ORT6430
RC(LST)=A(IND9)                                    ORT6440
1400 LST=LST+1                                    ORT6450
IF (NROW-(M+1)) 1440,1420,1410                 ORT6460
1410 RC(LST+1)=YSUM                            ORT6470
1420 RC(LST)=(SU-FM)*SSQ                         ORT6480
LST=LST+2                                       ORT6490
IF (M+2.GE.NROW) GO TO 1440                     ORT6500
LSTA=LST+M-1                                     ORT6510
IF (2*M+2.GT.NROW) LSTA=IIRGS(MXARGS+3)+NROW-1 ORT6520
IND8=ND8                                         ORT6530
DO 1430 I=LST,LSTA                           ORT6540
IND8=IND8+1                                      ORT6550
1430 RC(I)=A(IND8)                            ORT6560
C STORE S.D. OF PREDICTED VALUES               ORT6570
1440 LSTOR=IIRGS(MXARGS+2)                      ORT6580
IPIC=1                                           ORT6590
IND2=ND2+1                                      ORT6600
DO 1450 I=1,N                                     ORT6610
RC(LSTOR)=A(IND2)                                ORT6620
IND2=IND2+1                                      ORT6630
1450 LSTOR=LSTOR+1                               ORT6640
C START PRINTING                                ORT6650
1460 GO TO (1480,1470,1480,1470,1470), L2      ORT6660
1470 RETURN                                       ORT6670
1480 ITITLE=1                                    ORT6680
IF (L2.EQ.3) ITITLE=2                           ORT6690
IPG=1                                           ORT6700
NSU=SU+.5E-5                                     ORT6710
CALL PREPAK (5,NW1,NW1,IARGS(1),LHEAD)          ORT6720
IF (NW1.EQ.0) GO TO 1500                         ORT6730
DO 1490 I=1,4                                     ORT6740
LHEAD(I)=L(45)                                    ORT6750
1490 LHEAD(I+8)=L(45)                            ORT6760
LHEAD(5)=L(14)                                    ORT6770
LHEAD(6)=L(11)                                    ORT6780
LHEAD(7)=L(30)                                    ORT6790
LHEAD(8)=L(11)                                    ORT6800
1500 CALL PREPAK (5,NW1,NW1,IARGS(4),LHEAD(13)) ORT6810
IF (NW1.EQ.0) GO TO 1510                         ORT6820
LHEAD(13)=L(45)                                    ORT6830
LHEAD(14)=L(32)                                    ORT6840
LHEAD(15)=L(11)                                    ORT6850
LHEAD(16)=L(28)                                    ORT6860
LHEAD(17)=L(19)                                    ORT6870
LHEAD(18)=L(11)                                    ORT6880
LHEAD(19)=L(12)                                    ORT6890
LHEAD(20)=L(22)                                    ORT6900
LHEAD(21)=L(15)                                    ORT6910
LHEAD(22)=L(45)                                    ORT6920

```

```

LHEAD(23)=L(34)          ORT6930
LHEAD(24)=L(45)          ORT6940
1510 CALL PAGE (4)        ORT6950
      WRITE (IPRINT,1990) (LHEAD(I),I=1,12),IARGS(1)
      GO TO (1520,1820), ITITLE
C PRINT POLYFIT TITLE     ORT6980
1520 WRITE (IPRINT,2000) IARGS(3),(LHEAD(I),I=13,24),IARGS(4)  ORT6990
1530 GO TO (1540,1550), NMUI                                ORT7000
1540 NZW=NRMAX-NSU                                         ORT7010
      WRITE (IPRINT,2010) NSU,NZW,IARGS(2)
      GO TO 1560
1550 CALL RFORMAT (ARGS(2),1,8,NW1,NDEC1,10,A(1),A(1),0,0)   ORT7040
      CALL RFORMAT (A(1),1,8,NW1,NDEC1,0,ARGS(2),B(1),0,0)
      WRITE (IPRINT,2020) NSU,(B(I),I=1,10)                   ORT7060
1560 GO TO (1570,1580,1610,1650), IPG                      ORT7070
1570 IXA=4                                                 ORT7080
      IF (L2.EQ.3) IXA=IXA+(M-MX)
      IX=IXA
      CALL OPONE (N,M,MX,NX,ND2,ND3,ND19,B,SSQ,IX)
      IPG=2
1580 IF (NSU.GE.3) GO TO 1590                               ORT7130
      WRITE (IPRINT,2030)
      GO TO 1600
1590 CALL ORTPLT (ND19,ND2,N,SSQ,ND3,IB,IIRGS(IXA),IIRGS(2)) ORT7160
1600 IPG=3                                                 ORT7170
      GO TO 1510
1610 CALL OCOVAR (M,ND7,MD1,IHC,B,IHT)                     ORT7190
C PRINT ANALYSIS OF VARIANCE                            ORT7200
      CALL OANOVA (YSUM,SU,ND9,FM,M,N,ND7,SSQ,IHC,NSU,B)
      GO TO 1660
C REFIT FOR M=M-1                                       ORT7240
1620 IREFIT=1                                             ORT7250
      M=M-1
      FM=M
      M1=M-1
      M2=M+1
      SSOLD=SS
      IND17=ND17+1
      IND18S=ND18+N
      IND19=ND19+1
      IND19S=ND19
      DO 1630 J=1,M2
      A(IND19)=A(IND17)
      IND19=IND19+1
1630 IND17=IND17+1
      IF(M.EQ.0) GO TO 1640
      GO TO 170
C BEGIN REFIT TO PREDICTED VALUES                      ORT7410
1640 M1=M
C REFIT FOR M=M-1 COMPLETE-OUTPUT PAGE 3             ORT7440
      IPG=4
      M = M + 1
      GO TO 1510
1650 CALL OCOEFF (M1,N,ND18,ND17,IND19S,IND18S,IHC,B,IND7S,NSU,SS,SSOLD) ORT7470
      1,YSUM)
      RETURN
1660 IW=IIRGS(2)
      IND2=ND2+1
      IND3=ND3+1

```

```

IDPG=2*M+13                         ORT7530
IFI=1                                 ORT7540
DO 1700 I=1,N                         ORT7550
GO TO (1670,1680), NMUI               ORT7560
1670 A(IND2)=A(IND3)*RC(IW)          ORT7570
IW=IW+1                               ORT7580
GO TO 1690                            ORT7590
1680 A(IND2)=A(IND3)*ARGS(2)         ORT7600
1690 IND3=IND3+1                      ORT7610
1700 IND2=IND2+1                      ORT7620
IND5=ND5+1                            ORT7630
DO 1720 J=1,M                         ORT7640
IF=IFI                                ORT7650
ASUM=0.0                               ORT7660
IND2=ND2+1                            ORT7670
DO 1710 I=1,N                         ORT7680
ASUM=A(IF)*A(IND2)+ASUM              ORT7690
IND2=IND2+1                            ORT7700
1710 IF=IF+1                           ORT7710
A(IND5)=ASUM                          ORT7720
IFI=IFI+NPM                           ORT7730
1720 IND5=IND5+1                      ORT7740
ADEV=0.0                               ORT7750
IW=IIRGS(2)                           ORT7760
IFI=1                                  ORT7770
IND2=ND3+1                            ORT7780
DO 1770 I=1,N                         ORT7790
IND5=ND5+1                            ORT7800
IF=IFI                                ORT7810
ASUM=0.0                               ORT7820
DO 1730 J=1,M                         ORT7830
ASUM=ASUM+A(IF)*A(IND5)              ORT7840
IF=IF+NPM                             ORT7850
1730 IND5=IND5+1                      ORT7860
DEV=A(IND2)-ASUM                      ORT7870
GO TO (1740,1750), NMUI               ORT7880
1740 DEV=DEV**2*RC(IW)                ORT7890
IW=IW+1                               ORT7900
GO TO 1760                            ORT7910
1750 DEV=DEV**2*ARGS(2)              ORT7920
1760 ADEV=ADEV+DEV                   ORT7930
IND2=IND2+1                           ORT7940
1770 IFI=IFI+1                        ORT7950
IND18=ND18+1+N                       ORT7960
IM=ND12                               ORT7970
IND7S=ND18+N-M                       ORT7980
IND7=IND7S+1                          ORT7990
DO 1810 I=1,M                         ORT8000
IM=IM+I                               ORT8010
COEF=0.0                               ORT8020
IS=IM                                 ORT8030
IND5=ND5+I                            ORT8040
DO 1780 J=I,M                         ORT8050
COEF=COEF+A(IS)*A(IND5)              ORT8060
IS=IS+J                               ORT8070
1780 IND5=IND5+1                      ORT8080
DIF=A(IND18)-COEF                    ORT8090
IF (ABS(DIF).GT.0.0) GO TO 1790      ORT8100
C   8.0 EQUAL NUMBER OF DIGITS IN COMPUTER    ORT8110

```

```

      DIG=8.0                                     ORT8120
      GO TO 1800                                  ORT8130
1790  DIG=-FLOG10(ABS(DIF))+FLOG10(ABS(COEF))  ORT8140
      DIG=A MIN1(8.0,DIG)                         ORT8150
      DIG=A MAX1(-8.0,DIG)                        ORT8160
1800  A(IND7)=DIG                             ORT8170
      IND7=IND7+1                                ORT8180
1810  IND18=IND18+1                            ORT8190
C    DELETE GRAM FACTORS, VECTOR NORMS, GRAM DETERMINANT ORT8200
      GO TO 1620                                  ORT8210
C    TITLE FOR PRINT                           ORT8220
1820  II=IARGS(3)+3                           ORT8230
      IBA=II                                    ORT8233
      IBC=L(44)                                 ORT8235
      IF(II.GT.11)II=11                         ORT8237
      IF(M.GT.1) GO TO 1823                      ORT8240
      WRITE (IPRINT,2050) M,IARGS(4)             ORT8241
      GO TO 1530                                ORT8242
1823  IF(IBA.EQ.II) IBC=L(45)                 ORT8243
      WRITE (IPRINT,2050) M,IARGS(4),(L(44),IARGS(I),I=5,II),IBC ORT8244
      DO 1827 J=1,4                            ORT8245
      IF(M.LE.24*(J-1)+8) GO TO 1530           ORT8246
      II=24*j+11                               ORT8247
      III = II-23                             ORT8248
      II=MIN0(II,IARGS(3)+3)                   ORT8249
      IF(II.NE.III) GO TO 1825                ORT8250
      WRITE (IPRINT,2060) IARGS(II)             ORT8252
      GO TO 1530                                ORT8254
1825  III=III+1                             ORT8256
      IF(II.EQ.IBA) IBC=L(45)                 ORT8257
1827  WRITE (IPRINT,2060) IARGS(III-1),(L(44),IARGS(I),I=III,II),IBC ORT8258
      GO TO 1530                                ORT8260
1830  CALL ERROR (10)                          ORT8270
      RETURN                                     ORT8280
1840  CALL ERROR (25)                          ORT8290
      RETURN                                     ORT8300
C    MORTHO CHECK                           ORT8310
1850  IF (NARGS.EQ.7.OR.NARGS.EQ.9) GO TO 1860 ORT8320
      CALL ERROR (10)                           ORT8330
      RETURN                                     ORT8340
1860  IF (IARGS(3).GE.IARGS(4)) GO TO 1870   ORT8350
      CALL ERROR (26)                           ORT8360
      RETURN                                     ORT8370
1870  CALL ADRESS (2,IXM)                     ORT8380
      IF (IXM.LE.0) CALL ERROR (11)            ORT8390
      IF (IARGS(1)+IARGS(3)-1.GT.NROW) CALL ERROR (17) ORT8400
      IF (IARGS(2)+IARGS(4)-1.GT.NCOL) CALL ERROR (17) ORT8410
      IXM=IXM-1+IARGS(1)                      ORT8420
      IF (NERROR.GT.0) RETURN                  ORT8430
      J=7                                       ORT8440
      JJ=1                                      ORT8450
1880  CALL ADRESS (J,IMTRXA(JJ,1))           ORT8460
      IF (IMTRXA(JJ,1).GT.0) GO TO 1900       ORT8470
1890  CALL ERROR (11)                          ORT8480
      RETURN                                     ORT8490
1900  IMTRXA(JJ,2)=IARGS(3)                 ORT8500
      IF (JJ.EQ.2) IMTRXA(JJ,2)=IARGS(4)       ORT8510
      IMTRXA(JJ,3)=IARGS(4)                   ORT8520
      IF (IARGS(J-1).GT.NROW) GO TO 1890       ORT8530

```

```

IMTRXA(JJ,1)=IMTRXA(JJ,1)-1+IARGS(J-1)          ORT8540
IF (IARGS(J)+IARGS(4)-1.GT.NCOL) IMTRXA(JJ,3)=NCOL-IARGS(J)+1   ORT8550
IF (IARGS(J-1)+IARGS(3)-1.GT.NROW) IMTRXA(JJ,2)=NROW-IARGS(J-1)+1 ORT8560
IF (JJ.EQ.2.OR.NARGS.EQ.7) GO TO 1910           ORT8570
J=9                                         ORT8580
JJ=2                                         ORT8590
GO TO 1880                                     ORT8600
1910 IF (NERROR.NE.0) RETURN                   ORT8610
      IF (IMTRXA(1,2).NE.IARGS(3).OR.IMTRXA(1,3).NE.IARGS(4)) CALL ERRORORT8620
      1 (213)                                     ORT8630
      IF (NARGS.EQ.7) GO TO 1920                 ORT8640
      IF (IMTRXA(2,2).NE.IARGS(4).OR.IMTRXA(2,3).NE.IARGS(4)) CALL ERRORORT8650
      1 (213)                                     ORT8660
1920 IF (KIND(5).EQ.1) GO TO 1930             ORT8670
      CALL ADRESS (5,IIRGS(2))                  ORT8680
      IIRGS(1)=IIRGS(2)                         ORT8690
      IF (IIRGS(2).GT.0) GO TO 1940             ORT8700
      CALL ERROR (11)                           ORT8710
      RETURN                                     ORT8720
1930 SU=IARGS(3)                                ORT8730
      IF (ARGS(5).LE.0.0) CALL ERROR (25)        ORT8740
      NMUI=2                                     ORT8750
      KIND(2)=1                                 ORT8760
      ARGS(2)=ARGS(5)                           ORT8770
1940 M=IARGS(4)                                ORT8780
      DO 1950 I=1,M                            ORT8790
1950 IIRGS(I+3)=IXM+(I-1)*NROW               ORT8800
      N=IARGS(3)                                ORT8810
      FN=N                                      ORT8820
      GO TO 100                                 ORT8830
C      START STORING RESULTS FOR MORTHO       ORT8840
1960 IST=IMTRXA(1,1)                           ORT8850
      K=1                                       ORT8860
      MMTXC=IMTRXA(1,3)                         ORT8870
      MMTXR=IMTRXA(1,2)                         ORT8880
      DO 1980 I=1,MMTXC                         ORT8890
      KK=K                                      ORT8900
      ISTRR=IST                                ORT8910
      DO 1970 J=1,MMTXR                         ORT8920
      RC(ISTRR)=A(KK)                           ORT8930
      KK=KK+1                                  ORT8940
1970 ISTRR=ISTRR+1                           ORT8950
      K=K+NPM                                  ORT8960
1980 IST=IST+NROW                           ORT8970
      IF (NARGS.EQ.7) RETURN                   ORT8980
      LSWT=1                                    ORT8990
      IND7=ND12+1                             ORT9000
      IST=IMTRXA(2,1)                           ORT9010
      MMTXR=IMTRXA(2,2)                         ORT9020
      MMTXC=IMTRXA(2,3)                         ORT9030
      GO TO 1250                               ORT9040
C                                         ORT9050
1990 FORMAT (/35X,22HLEAST SQUARES FIT FOR ,12A1,11H IN COLUMN ,I4) ORT9060
2000 FORMAT (25X,26HAS A POLYNOMIAL OF DEGREE ,I2,4H IN ,12A1,11H IN COORT9070
      1LUMN ,I4)                                ORT9080
2010 FORMAT (20X,6HUSING ,I4,22H NON-ZERO WEIGHTS AND ,I4,24H ZERO WEIGORT9090
      1HTS IN COLUMN ,I4)                      ORT9100
2020 FORMAT (35X,6HUSING ,I4,19H NON-ZERO WEIGHTS =,10A1)          ORT9110
2030 FORMAT (60H0 PLOTS ARE NOT PRINTED BECAUSE NO. OF POINTS IS LESS TORT9120

```

1HAN 3) ORT9130  
2050 FORMAT (23X,24HAS A LINEAR FUNCTION OF ,I2,31H PREDICTOR VARIABLESORT9150  
1 IN COLUMNS,I4,8(A1,I4)) ORT9160  
2060 FORMAT (I4,24(A1,I4)) ORT9165  
END ORT9170

```

SUBROUTINE ORTHRV (A,NROW,N,NCOL,IND,X,NASIZE,XP)          ORV 10
C VERSION 5.00      ORTHRV   5/15/70                      ORV 20
C SUBROUTINE ORTHRV(A,NROW,N,NCOL,IND,X,NASIZE,XP)          ORV 30
C SUBROUTINE TO CHECK TO SEE IF MATRIX IS ORTHOGONAL        ORV 40
C DIMENSION A(NROW,1), X(1), IND(1)                         ORV 50
C DOUBLE PRECISION XP(1)                                     ORV 60
C IF NUMBER OF ROWS IS GREATER THAN NUMBER OF COLUMNS COMPUTE A'A ORV 70
C OTHERWISE AA'                                           ORV 80
C IF (N.GT.NCOL) GO TO 10
10  L2P=1                                         ORV 100
    MP=N                                         ORV 110
    GO TO 20                                       ORV 120
10  L2P=2                                         ORV 130
    MP=NCOL                                       ORV 140
20  CALL MXTPX (A,NROW,N,NCOL,X,L2P,NASIZE,XP)           ORV 150
    IC=1                                         ORV 160
    IND(1)=0                                      ORV 170
    IND(2)=0                                      ORV 180
    DO 80 I=1,MP                                    ORV 190
    DO 80 J=1,MP                                    ORV 200
    IF (I.EQ.J) GO TO 40                          ORV 210
    IF (X(IC)) 30,80,30                           ORV 220
30  IF (ABS(X(IC))-1.E-7) 60,60,90                ORV 230
40  IF (X(IC)-1.0) 50,80,50                      ORV 240
50  IF (ABS(X(IC)-1.0)-1.E-7) 60,60,70            ORV 250
60  IND(2)=1                                      ORV 260
    GO TO 80                                       ORV 270
70  IND(1)=2                                      ORV 280
80  IC=IC+1                                       ORV 290
    GO TO 100                                      ORV 300
90  IND(1)=2                                      ORV 310
    IND(2)=2                                      ORV 320
    GO TO 150                                      ORV 330
100 IF (IND(1).EQ.0.AND.IND(2).EQ.1) IND(1)=1          ORV 340
    IF (N.EQ.NCOL) GO TO 150                      ORV 350
C SET UP INDICATORS FOR RECTANGULAR MATRICES           ORV 360
    GO TO (110,130), L2P                           ORV 370
110 IF (IND(1).EQ.1) GO TO 120                      ORV 380
    IND(1)=3                                       ORV 390
    IND(2)=3                                       ORV 400
    GO TO 150                                      ORV 410
120 IND(1)=-3                                      ORV 420
    IND(2)=-3                                      ORV 430
    GO TO 150                                      ORV 440
130 IF (IND(1).EQ.1) GO TO 140                      ORV 450
    IND(1)=4                                       ORV 460
    IND(2)=4                                       ORV 470
    GO TO 150                                      ORV 480
C *
    IND(1)=0 EXACT ORTHOGONAL                      ORV 490
    IND(1)=1 RELATIVE (1.E-7) ORTHOGONAL           ORV 500
    IND(1)=2 NON-ORTHOGONAL                        ORV 510
C INDICATORS FOR RECTANGULAR MATRICES               ORV 520
    IND(1)=-3 RELATIVE ORTHOGONAL ROWWISE         ORV 530
    IND(1)=3 EXACT ORTHOGONAL ROWWISE              ORV 540
    IND(1)=-4 RELATIVE ORTHOGONAL COLUMNWISE       ORV 550
    IND(1)=4 EXACT ORTHOGONAL COLUMNWISE           ORV 560
    IND(2)=-1 DIAGONAL TERMS ARE 1.0 SE           ORV 570
    IND(2)=0 EXACT ORTHOGONAL NORMALIZED          ORV 580
C

```

C	IND(2)=1 RELATIVE ORTHOGONAL NORMALIZED	ORV 600
C	IND(2)=2 NON-ORTHOGONAL	ORV 610
C	IND(2)=-3 RELATIVE ROWWISE (NORMALIZED)	ORV 620
C	IND(2)=3 EXACT ROWWISE (NORMALIZED)	ORV 630
C	IND(2)=-4 RELATIVE COLUMNWISE (NORMALIZED)	ORV 640
C	IND(2)=4 EXACT COLUMNWISE (NORMALIZED)	ORV 650
C	*	ORV 660
C	*	ORV 670
140	IND(1)=-4	ORV 680
	IND(2)=-4	ORV 690
150	RETURN	ORV 700
	END	ORV 710

```

SUBROUTINE ORTPLT( ND19,ND2,N,SSQ,ND3,IB,IXA,IWS)          ORP 10
C VERSION 5.00      ORTPLT      5/15/70                  ORP 20
C THIS PROGRAM IS USED BY ORTHO TO GENERATE PLOTS          ORP 30
C WRITTEN BY S PEAVY      10/11/69                  ORP 40
C DIMENSION IU(1), IB(1)                  ORP 50
C EQUIVALENCE (IU,A)                  ORP 60
C *****
C COMMON/BLOCKD/IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,   ORP 80
1NARGS,VWXYZ(8),NERROR                  ORP 90
COMMON/BLOCRC/NRC,RC(12600)                  ORP 100
COMMON/HEADER/NOCARD(80),ITLE(60,6),LNCNT,IPRINT,NPAGE,IPUNCH    ORP 110
COMMON/SCRAT/NS,NS2,A(13500)                  ORP 120
COMMON/FMAT/IFMTX(6),IOSWT,IFMTS(6),LHEAD(96)      ORP 130
DIMENSION ARGS(100)                  ORP 140
EQUIVALENCE (ARGS(1),RC(12501))      ORP 150
COMMON /ABCDEF/L(48)                  ORP 160
COMMON/CONSTS/PI,E,HALFPI,DEG,RAD,XALOG      ORP 170
C *****
C IW=IWS                  ORP 180
C IWST=1                  ORP 182
C IF(KIND(2).EQ.0) GO TO 18310      ORP 184
C IWST=2                  ORP 186
C WT=ARGS(2)                  ORP 187
C
18310 IND4 = ND19+1                  ORP 188
IND2=ND2+1                  ORP 190
NZW=N                  ORP 200
DO 18320 I=1,N                  ORP 205
GO TO (18312,18314),IWST      ORP 210
C
18312 WT=RC(IW)                  ORP 211
IW=IW+1                  ORP 212
C
18314 IF(WT.NE.0.0) GO TO 18316      ORP 213
IU (IND4)=27                  ORP 214
NZW=NZW-1                  ORP 215
GO TO 18318                  ORP 216
C
18316 Z=A(IND4)/FSQRT(SSQ/WT-A(IND2)**2)      ORP 218
IZ=Z /.3                  ORP 220
IF (Z.GT.0.0.AND.AMOD(Z,.3).NE.0.0)IZ=IZ+1      ORP 230
IU(IND4)=IZ+13                  ORP 240
IF(IU(IND4).LE.0) IU(IND4)=1      ORP 250
IF (IU(IND4).GT.26) IU(IND4)=26      ORP 260
C
18318 IND2=IND2+1                  ORP 270
C
18320 IND4=IND4+1                  ORP 280
IND3=ND3+1                  ORP 290
YMAX=A(IND3)                  ORP 300
YMIN=A(IND3)                  ORP 310
DO 18340 I=2,N                  ORP 320
IND3=IND3+1                  ORP 330
IF(YMIN.LE.A(IND3)) GO TO 18330      ORP 340
YMIN=A(IND3)                  ORP 350
GO TO 18340                  ORP 360
C
18330 IF(YMAX.LT.A(IND3)) YMAX=A(IND3)      ORP 370
C
18340 CONTINUE                  ORP 380
YMM=ABS (YMAX-YMIN)/50.      ORP 390
YMX=FLOAT(N-1)/50.      ORP 400
CALL PAGE(0)                  ORP 410
IPLOT=1                  ORP 420
WRITE(IPRINT,18350)      ORP 430
C
18350 FORMAT(15X,36HSTANDARDIZED RESIDUALS VS ROW NUMBER,22X,   ORP 440
1 42HSTANDARDIZED RESIDUALS VS PREDICTED VALUES)      ORP 450
C                                         ORP 460

```

```

18355 WRITE(IPRINT,18360) (L(39),I=1,88) ORP 470
18360 FORMAT(7X,2(1H+,9A1),1H+,4A1,1HX,4A1,2(1H+9A1),1H+,10X,2(1H+,9A1)) ORP 480
   1 ,1H+,4A1,1HX,4A1,2(1H+,9A1),1H+) ORP 490
   YYPR=3.75 ORP 500
   LINE=26 ORP 510
18390 DO 20050 I=1,5 ORP 520
   DO 18400 IJI=1,102 ORP 530
18400 IB(IJI)=L(45) ORP 540
   GO TO (18410,19000),IPLOT ORP 550
18410 IND3=ND3+1 ORP 560
   IND4=ND19+1 ORP 570
   DO 18430 IJI=1,N ORP 580
   IF(IU(IND4).NE.LINE) GO TO 18420 ORP 590
   IZ=FLOAT(IJI-1)/YMX +.5 ORP 600
   IZ=IZ+1 ORP 610
   IF(IZ.LE.0) IZ=1 ORP 620
   IF(IZ.GT.51) IZ=51 ORP 630
   IB(IZ)=L(41) ORP 640
   IZ=(A(IND3)-YMIN)/YMM ORP 650
   IZ=IZ+1 ORP 660
   IF(IZ.LE.0) IZ=1 ORP 670
   IF(IZ.GT.51) IZ=51 ORP 680
   IB(IZ+51)=L(41) ORP 690
18420 IND4=IND4+1 ORP 700
18430 IND3=IND3+1 ORP 710
   GO TO 20000 ORP 720
19000 IND4=ND19+1 ORP 730
   IX=IXA ORP 740
   DO 19010 IJI=1,N ORP 750
   IF(IU(IND4).NE.LINE) GO TO 19005 ORP 760
   IZ=(RC(IX)-XMIN)/XMM ORP 770
   IZ=IZ+1 ORP 780
   IF(IZ.LE.0) IZ=1 ORP 790
   IF(IZ.GT.51) IZ=51 ORP 800
   IB(IZ)=L(41) ORP 810
   RATIO=(AN-GAMMA)/FDEN ORP 820
   YMM=4.91*(RATIO**.14-(1.-RATIO)**.14) ORP 830
   AN=AN-1.0 ORP 840
   IF(AN.LT.2. .AND. NZW.LE.10) GAMMA = 1./3. ORP 850
   IZ=YMM/.1 ORP 860
   IZ=IZ+26 ORP 870
   IF(IZ.LE.0) IZ=1 ORP 880
   IF(IZ.GT.51) IZ=51 ORP 890
   IB(IZ+51)=L(41) ORP 900
19005 IX=IX+1 ORP 910
19010 IND4=IND4+1 ORP 920
20000 IF(I-1) 20010,20010,20030 ORP 930
20010 WRITE(IPRINT,20020)YYPR,(IB(IJI),IJI=1,51),YYPR,(IB(IJI),IJI=52,
   1 102) ORP 940
   ORP 950
20020 FORMAT(1X,F5.2,1H+,51A1,1H+,3X,F5.2,1H+,51A1,1H+) ORP 960
   GO TO 20045 ORP 970
20030 WRITE(IPRINT,20040) (IB(IJI),IJI=1,102) ORP 980
20040 FORMAT(6X,1H-,51A1,1H-,8X,1H-,51A1,1H-) ORP 990
20045 LINE=LINE-1 ORP1000
   IF(LINE.EQ.0) GO TO 20060 ORP1010
20050 CONTINUE ORP1020
   YYPR=YYPR-1.5 ORP1030
   GO TO 18390 ORP1040
20060 WRITE(IPRINT,18360) (L(39),I=1,88) ORP1050

```

```

GO TO (20070,21100 ),IPLOT          ORP1060
20070 YMM=YMX*25.0 +1.0             ORP1070
YMMY= (YMAX-YMIN)/2.+YMIN          ORP1080
WRITE(IPRINT,20080) YMM,N,YMIN,YMMY,YMAX ORP1090
20080 FORMAT(6X,3H1.0,18X,F9.4,16X,I5,2H.0 ,1PE15.4,E26.4,10X,E10.4) ORP1100
      WRITE (IPRINT,20090)               ORP1110
20090 FORMAT (1H )                  ORP1120
      IPLOT=2                         ORP1130
      IX=IXA                         ORP1140
      XMAX=RC(IX)                    ORP1150
      XMIN=RC(IX)                    ORP1160
      DO 21000 I=2,N                 ORP1170
      IX=IX+1                        ORP1180
      IF(RC(IX).GT.XMAX) XMAX=RC(IX) ORP1190
      IF(XMIN.GT.RC(IX)) XMIN=RC(IX) ORP1200
21000 CONTINUE                      ORP1210
      XMM = ABS (XMAX-XMIN)/50.       ORP1230
      GAMMA=PI/8.0                   ORP1240
      AN=NZW                         ORP1245
      FDEN=AN-2.*GAMMA+1.0           ORP1250
      WRITE(IPRINT,21010) (LHEAD(I),I=13,24) ORP1270
21010 FORMAT(14X,26HSTANDARDIZED RESIDUALS VS ,12A1, 21X, ORP1280
      1 42HPROBABILITY PLOT OF STANDARDIZED RESIDUALS) ORP1290
      GO TO 18355                   ORP1300
21100 YMMY=(XMAX-XMIN) /2.+XMIN     ORP1310
      WRITE(IPRINT,21110) XMIN,YMMY,XMAX ORP1320
21110 FORMAT(1PE13.4,14X, E12.4, 8X, E12.4,7X,4H-2.5,22X,3H0.0,22X, ORP1330
      1 3H2.5)                       ORP1340
      RETURN                         ORP1350
      END                            ORP1360

```

```

SUBROUTINE OUTPUT                               OUT   10
C      VERSION 5.00     OUTPUT    5/15/70          OUT   20
COMMON /BLOCKA/  MODE,M,KARD(83),KARG,ARG,ARG2,NEWCD(80),KRDEND  OUT   30
COMMON /BLOCKB/  NSTMT,NSTMTX,NSTMTH,NCOM,LCOM,IOVFL,COM(2000)  OUT   40
COMMON /BLOCRC/  NRC,RC(12600)                 OUT   50
COMMON /BLOCKD/  IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NOUT 60
IARGS,VWXYZ(8),NERROR                         OUT   70
DIMENSION ARGS(100)                           OUT   80
EQUIVALENCE (ARGS(1),RC(12501))              OUT   90
COMMON /BLOCKC/  KIO,INUNIT,ISCRAT,KBDOUT,KRDKNT,LLIST        OUT  100
C
C      WRITE RECORD ON SCRATCH UNIT            OUT  110
C
C      IF (NERROR.EQ.0.AND.LLIST.EQ.0) GO TO 10    OUT  140
IF (MODE.EQ.3) GO TO 20                      OUT  150
WRITE (ISCRAT,30) NEWCD                      OUT  160
10     RETURN                                    OUT  170
20     I=NSTMT/10                                OUT  180
WRITE (ISCRAT,40) I,NEWCD                    OUT  190
GO TO 10                                     OUT  200
C
30     FORMAT (4X,80A1)                          OUT  210
40     FORMAT (1H+,I3,80A1)                      OUT  220
      END                                       OUT  230
                                                OUT  240

```

```

C SUBROUTINE PACK (NWORD,MWORD,NO,IP)          PAC 10
C VERSION 5.00      PACK      5/15/70          PAC 20
C WRITTEN BY S PEAVY 9/17/69          PAC 30
C NWORD   CONTAINS CHARACTERS TO BE PACKED OR UNPACKED    PAC 40
C MWORD   THE PACKED CHARACTERS IN CODED FORM (SEE BELOW) OR THE    PAC 50
C           UNPACKED CHARACTERS          PAC 60
C NO      NO OF CHARACTERS TO BE PACKED OR UNPACKED    PAC 70
C IP      IP=0   PACK          PAC 80
C IP      IP=1   UNPACK        PAC 90
C           PAC 100
C THE CHARACTERS ARE PACKED IN A CODED FORM. EACH CHARACTER HAS BEENPAC 110
C ASSIGNED A VALUE IN OMCONV. THIS VALUE IS 1 LESS THAN THE          PAC 120
C SUBSCRIPT OF L (IN LABELED COMMON ABCDEF) FOR THAT PARTICULAR          PAC 130
C CHARACTER. THESE VALUES ARE STORED IN KARD. THE VALUES OF THE          PAC 140
C CHARACTERS ARE PACKED AS FOLLOWS          PAC 150
C           MWORD(I)=(KARD(K)+1)*2**16+(KARD(K-1)+1)*2**8+KARD(K-2)+1PAC 160
C           PAC 170
C COMMON /ABCDEF/ L(48)          PAC 180
C DIMENSION NWORD(1), MWORD(1)          PAC 190
C KB=1          PAC 200
C KA=1          PAC 210
C IF (IP.EQ.1) GO TO 30          PAC 220
C PACK          PAC 230
10 MWORD(KA)=0          PAC 240
DO 20 I=1,3          PAC 250
MWORD(KA)=MWORD(KA)*256+NWORD(KB)+1          PAC 260
KB=KB+1          PAC 270
IF (KB.GT.NO) GO TO 22          PAC 280
20 CONTINUE          PAC 290
KA=KA+1          PAC 300
GO TO 10          PAC 310
22 ICE=MOD(NO,3)          PAC 311
IF (ICE.EQ.0) RETURN          PAC 312
ICE=3-ICE          PAC 313
24 MWORD(KA)=MWORD(KA)*256+45          PAC 314
ICE=ICE-1          PAC 315
IF (ICE.EQ.0) RETURN          PAC 316
GO TO 24          PAC 317
C UNPACK          PAC 320
30 ICA=NWORD(KB)          PAC 330
ICD=65536          PAC 340
DO 40 I=1,3          PAC 350
ICB=ICA/ICD          PAC 360
IF (ICB.EQ.0) GO TO 40          PAC 370
MWORD(KA)=L(ICB)          PAC 380
KA=KA+1          PAC 390
IF (KA.GT.NO) RETURN          PAC 400
ICA=ICA-ICB*ICD          PAC 410
40 ICD=ICD/256          PAC 420
KB=KB+1          PAC 430
GO TO 30          PAC 440
END          PAC 450

```

C	SUBROUTINE PAGE (J)	PAG 10
C	VERSION 5.00 PAGE 5/15/70	PAG 20
C	BRING UP A NEW PAGE AND PRINT OMNITAB CARD AND PAGE NUMBER	PAG 30
C	THEN, IF J = 0, DONE	PAG 40
C	J = 1, PRINT TITLE1	PAG 50
C	J = 2, PRINT TITLE1, 2	PAG 60
C	ETC. FOR J = 3, 4	PAG 70
C	THIS ROUTINE ASSUMES THAT THE EXECUTIVE SYSTEM LEAVES	PAG 80
C	THE PRINTER FORM AT THE TOP OF THE FIRST BLANK PAGE.	PAG 90
C	COMMON/HEADER/NOCARD(80),ITLE(60,6),LNCNT,IPRINT,NPAGE,IPUNCH	PAG 100
C	NPAGE=NPAGE+1	PAG 110
C	WRITE (IPRINT,20) NOCARD,NPAGE	PAG 150
C	IF (J.LE.0.OR.J.GT.4) GO TO 10	PAG 160
C	WRITE (IPRINT,30) ((ITLE(I,II),I=1,60),II=1,J)	PAG 180
10	RETURN	PAG 190
C		PAG 200
C	FORMAT (1H1,19X,80A1,10X,4HPAGE,I4)	PAG 210
20	FORMAT (1X,120A1/1X,120A1)	PAG 220
30	END	PAG 230
		PAG 240

```

SUBROUTINE PDMOTE                                     PDM 10
C VERSION 5.00      PDMOTE      5/15/70             PDM 20
COMMON /BLOCRC/ NRC,RC(12600)                      PDM 30
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NPDM 40
IARGS,VWXYZ(8),NERROR                            PDM 50
DIMENSION ARGS(100)                               PDM 60
EQUIVALENCE (ARGS(1),RC(12501))                  PDM 70
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG              PDM 80
L2=L2-10                                         PDM 90
C                                                 PDM 100
C L2 : 0 FOR PROMOTE,    1 FOR DEMOTE   ( L2 ORIGINALLY 10, 11 ) PDM 110
C                                                 PDM 120
C IF (MOD(NARGS,2).NE.0) GO TO 30                PDM 130
I=10                                              PDM 140
10 CALL ERROR (I)                                PDM 150
20 RETURN                                         PDM 160
30 IF(KIND(1).EQ.1) GO TO 35                    PDM 163
NR = IARGS (1)                                    PDM 170
IARGS(1)=1                                       PDM 180
CALL CHKCOLUMN (I)                             PDM 190
IF (I.EQ.0) GO TO 40                           PDM 200
35 I = 20                                         PDM 210
GO TO 10                                         PDM 220
C                                                 PDM 230
C IF NUMBER OF ROWS TO BE MOVED IS NEGATIVE, FLIP INSTRUCTIONS. PDM 240
C I.E. PROMOTE -6 IS THE SAME AS DEMOTE 6          PDM 250
40 IF (NR.GE.0) GO TO 50                         PDM 260
L2=1-L2                                         PDM 270
NR=-NR                                         PDM 280
50 NARGS=NARGS-1                                 PDM 290
C                                                 PDM 300
C CHECK DISTANCE OF MOVE                         PDM 310
C                                                 PDM 320
IF (L2.EQ.0) GO TO 70                           PDM 330
IF (NR+NRMAX.LE.NROW) GO TO 100                 PDM 340
CALL ERROR (231)                                PDM 345
NRMAX=NROW-NR                                    PDM 350
IF (NRMAX) 20,20,100                            PDM 35I
70 NDIFF = NRMAX-NR                            PDM 355
IF (NDIFF) 81,82,82                            PDM 360
81 CALL ERROR (230)                                PDM 365
NDIFF = 0                                         PDM 370
NR = NRMAX                                         PDM 375
82 IF (NARGS.GT.0) GO TO 100                     PDM 380
J = IARGS (1)-1                                  PDM 385
DO 95 I1 = 1,NCOL                                PDM 390
K1 = J+1                                         PDM 395
IF (NDIFF.EQ.0) GO TO 86                         PDM 400
K2 = K1 + NR                                     PDM 410
DO 85 I2 = 1,NDIFF                                PDM 420
RC(K1) = RC(K2)                                   PDM 425
K1 = K1+1                                         PDM 430
85 K2 = K2+1                                         PDM 435
86 DO 90 I3 = 1,NR                                PDM 440
RC(K1) = 0.0                                       PDM 445
90 K1 = K1 +1                                      PDM 450
95 J = J + NROW                                     PDM 460
GO TO 20                                         PDM 470
100 LIMIT=NARGS                                 PDM 480

```

```

IF (LIMIT.EQ.0) LIMIT=2*NCOL          PDM 490
IF (NERROR.NE.0) GO TO 20            PDM 500
IF (NRMAX.NE.0) GO TO 110           PDM 510
GO TO 10                            PDM 520
C                                     PDM 530
C                                     START PROMOTING OR DEMOTING
C                                     PDM 540
C
110 DO 180 I=1,LIMIT,2             PDM 560
    IF (NARGS.NE.0) GO TO 120        PDM 570
    K1=IARGS(1)                   PDM 580
    K2=K1                         PDM 590
    IARGS(1)=IARGS(1)+NROW         PDM 600
    GO TO 130                      PDM 610
120 K1=IARGS(I+1)                 PDM 620
    K2=IARGS(I+2)                 PDM 630
130 IF (L2.EQ.0) GO TO 150        PDM 640
C                                     PDM 650
C                                     DEMOTE COL AT K1 TO COL AT K2
C                                     PDM 660
C
    K1=K1+NRMAX                  PDM 670
    K2=K2+NRMAX+NR                PDM 680
    DO 140 J=1,NRMAX              PDM 690
    K1=K1-1                        PDM 700
    K2=K2-1                        PDM 710
140 RC(K2)=RC(K1)                 PDM 720
    GO TO 180                      PDM 730
C                                     PDM 740
C                                     PROMOTE COL AT K1 TO COL AT K2
C                                     PDM 750
C
150 JJ=NRMAX-NR                  PDM 760
    IF (JJ.EQ.0) GO TO 165        PDM 770
    K1=K1+NR                        PDM 780
    DO 160 J=1,JJ                  PDM 785
    RC(K2)=RC(K1)                 PDM 790
    K1=K1+1                         PDM 800
    K2=K2+1                         PDM 810
160 K2=K2+1                      PDM 820
C                                     PDM 830
C                                     IF PROMOTE ARRAY, FILL REST OF COLUMN WITH ZEROES.
C                                     PDM 840
C
    IF (NARGS.NE.0) GO TO 180      PDM 850
165 JJ=JJ+1                       PDM 860
    DO 170 J=JJ,NRMAX              PDM 870
    RC(K2)=0.                      PDM 880
    PDM 890
170 K2=K2+1                      PDM 900
180 CONTINUE                      PDM 910
    IF (L2.NE.0) NRMAX=NRMAX+NR   PDM 920
    GO TO 20                        PDM 930
    END                            PDM 940
                                PDM 950

```

```

SUBROUTINE PHYCON (NAME)          PHY 10
C VERSION 5.00      PHYCON      5/15/70    PHY 20
COMMON /BLOCKA/ MODE,M,KARD(83),KARG,ARG,ARG2,NEWCD(80),KRDEND   PHY 30
COMMON/PCONST/JPC,P(40),N(40)    PHY 40
C REMOVE                         PHY 50
C
C PHYSICAL CONSTANT LIST          PHY 60
C
C ENTRIES ARE IN PAIRS, FIRST MKS VALUE, THEN CGS (ELECTROMAGNETIC) PHY 70
C
C
C PI      PI                      PHY 80
C E      BASE OF NATURAL LOGS     PHY 90
C C      SPEED OF LIGHT IN VACUUM PHY 100
C Q      ELEMENTARY CHARGE       PHY 110
C N      AVOGADRO CONSTANT      PHY 120
C ME     ELECTRON REST MASS     PHY 130
C MP     PROTON REST MASS       PHY 140
C F      FARADAY CONSTANT       PHY 150
C H      PLANCK CONSTANT        PHY 160
C ALPHA  FINE STRUCTURE CONSTANT PHY 170
C QME    CHARGE TO MASS RATIO FOR ELECTRON PHY 180
C RINF   RYDBERG CONSTANT       PHY 190
C GAMMA  GYROMAGNETIC RATIO OF PROTON (CORRECTED FOR H2O) PHY 200
C MUB    BOHR MAGNETON          PHY 210
C R      GAS CONSTANT           PHY 220
C K      BOLTZMANN CONSTANT     PHY 230
C CONE   FIRST RADIATION CONSTANT PHY 240
C CTWO   SECOND RADIATION CONSTANT PHY 250
C SIGMA  STEPHAN-BOLTZMANN CONSTANT PHY 260
C G      GRAVITATIONAL CONSTANT PHY 270
C
C
C IF NAME .LE. 0, NAME = INDEX FROM MKS,CGS  0 = CGS, -1 = MKS  PHY 280
C
C J=JPC                          PHY 290
C IF (NAME.GT.0) GO TO 10         PHY 300
C JPC=NAME                        PHY 310
C RETURN                          PHY 320
C
10 DO 20 I=1,20                  PHY 330
C IF (NAME.EQ.N(I)) GO TO 30      PHY 340
C
20 CONTINUE                       PHY 350
C ARG=0.                           PHY 355
C RETURN                          PHY 360
C
30 I=I+I+J                        PHY 370
C ARG=P(I)                         PHY 380
C RETURN                          PHY 390
C END                            PHY 400
C

```

```

C      BLOCK DATA PHYSIC          PHC 10
C      VERSION 5.00   PHYSIC      5/15/70    PHC 20
C      BLOCK DATA PHYSICAL CONSTANTS   PHC 30
C      (THEIR VALUES AND NUMBER REPRESENTATION)  PHC 40
C      BLOCK DATA          PHC 50
C      COMMON/PCONST/JPC,P(40),N(40)    PHC 60
C          PCONST DEFINES PHYSICAL CONSTANT VALUES    PHC 70
C          SI UNITS           CGS UNITS    PHC 80
C          SEE BELOW FOR FURTHER COMMENTS    PHC 90
C          PI     P( 1)=3.1415926535    P( 2)=3.1415926535    PHC 100
C          E      P( 3)=2.718281828459    P( 4)=2.718281828459    PHC 110
C          C      P( 5)=2.997925E8     P( 6)=2.997925E10    PHC 120
C          Q      P( 7)=1.60210E-19    P( 8)=1.60210E-20    PHC 130
C          N      P( 9)=6.02252E23    P(10)=6.02252E23    PHC 140
C          ME     P(11)=9.1091E-31    P(12)=9.1091E-28    PHC 150
C          MP     P(13)=1.67252E-27    P(14)=1.67252E-24    PHC 160
C          F      P(15)=9.64870E4     P(16)=9648.70    PHC 170
C          H      P(17)=6.6256E-34     P(18)=6.6256E-27    PHC 180
C          ALPHA  P(19)=7.29720E-3     P(20)=7.29720E-3    PHC 190
C          QME    P(21)=1.758796E11    P(22)=17587960.    PHC 200
C          RINF   P(23)=10973731.    P(24)=109737.31    PHC 210
C          GAMMA  P(25)=2.67519E8     P(26)=26751.9    PHC 220
C          MUB    P(27)=9.2732E-24    P(28)=9.2732E-21    PHC 230
C          R      P(29)=8.3143     P(30)=8.3143E7    PHC 240
C          K      P(31)=1.38054E-23    P(32)=1.38054E-16   PHC 250
C          CONE   P(33)=3.7415E-16    P(34)=3.7415E-5    PHC 260
C          CTWO   P(35)=1.43879E-2     P(36)=1.43879    PHC 270
C          SIGMA  P(37)=5.6697E-8     P(38)=5.6697E-5    PHC 280
C          G      P(39)=6.670E-11    P(40)=6.670E-8    PHC 290
C          DATA P(1),P(2),P(3),P(4),P(5),P(6),P(7),P(8),P(9),P(10)/    PHC 300
C          12*3.1415926535,2*2.718281828459,2.997925E8,2.997925E10,    PHC 310
C          2 1.60210E-19,1.60210E-20,2*6.02252E23/    PHC 320
C          DATA P(11),P(12),P(13),P(14),P(15),P(16),P(17),P(18),P(19),P(20)/    PHC 330
C          1 9.1091E-31,9.1091E-28,1.67252E-27,1.67252E-24,9.64870E4,9648.70,    PHC 340
C          2 6.6256E-34,6.6256E-27,2*7.29720E-3/    PHC 350
C          DATA P(21),P(22),P(23),P(24),P(25),P(26),P(27),P(28),P(29),P(30)/    PHC 360
C          1 1.758796E11,17587960.,10973731.,109737.31,2.67519E8,26751.9,    PHC 370
C          2 9.2732E-24,9.2732E-21,8.3143,8.3143E7/    PHC 380
C          DATA P(31),P(32),P(33),P(34),P(35),P(36),P(37),P(38),P(39),P(40)/    PHC 390
C          1 1.38054E-23,1.38054E-16,3.7415E-16,3.7415E-5,1.43879E-2,1.43879,    PHC 400
C          2 5.6697E-8,5.6697E-5,6.670E-11,6.670E-8/    PHC 410
C          DATA N(1),N(2),N(3),N(4),N(5),N(6),N(7),N(8),N(9),N(10)/    PHC 420
C          1 11907,3645,2187,12393,10206,9612,9909,4374,5832,1069/    PHC 430
C          DATA N(11),N(12),N(13),N(14),N(15),N(16),N(17),N(18),N(19),N(20)/    PHC 440
C          1 12749,13379,5143,10046,13122,8019,2606,2750,14101,5103/    PHC 450
C          PHYSICAL CONSTANTS INTEGER REPRESENTATION    PHC 460
C          N (1)= 11907= PI      (PI)    PHC 470
C          N (2)= 3645= E       (BASE OF NATURAL LOGS)    PHC 480
C          N (3)= 2187= C       (SPEED OF LIGHT IN VACUUM)    PHC 490
C          N (4)= 12393= Q      (ELEMENTARY CHARGE)    PHC 500
C          N (5)= 10206= N      (AVOGADRO CONSTANT)    PHC 510
C          N (6)= 9612= ME     (ELECTRON REST MASS)    PHC 520
C          N (7)= 9909= MP     (PROTON REST MASS)    PHC 530
C          N (8)= 4374= F      (FARADAY CONSTANT)    PHC 540
C          N (9)= 5832= H      (PLANCK CONSTANT)    PHC 550
C          N(10)= 1069= ALPHA  (FINE STRUCTURE CONSTANT)    PHC 560
C          N(11)= 12749= QME   (CHARGE TO MASS RATIO FOR ELECTRON)    PHC 570
C          N(12)= 13379= RINF  (RYDBERG CONSTANT)    PHC 580
C          N(13)= 5103= C      (RYDBERG CONSTANT)    PHC 590

```

C N(13)=	5143=	GAMMA	(GYROMAGNETIC RATIO OF PROTON-CORRECTED FOR H20)	PHC 600
C N(14)=	10046=	MUB	(BOHR MAGNETON)	PHC 610
C N(15)=	13122=	R	(GAS CONSTANT)	PHC 620
C N(16)=	8019=	K	(BOLTZMANN CONSTANT)	PHC 630
C N(17)=	2606=	CONE	(FIRST RADIATION CONSTANT)	PHC 640
C N(18)=	2750=	CTWO	(SECOND RADIATION CONSTANT)	PHC 650
C N(19)=	14101=	SIGMA	(STEPHAN-BOLTZMANN CONSTANT)	PHC 660
C N(20)=	5103=	G	(GRAVITATIONAL CONSTANT)	PHC 670
END				

SUBROUTINE PLOT PLO 10  
 C VERSION 5.00 PLOT 5/15/70 PLO 20  
 C S PEAVY 1/18/68 PLO 30  
 C THIS ROUTINE PLOTS MAX. OF 5 CURVES. IF MORE THEN ONE POINT FALLS PLO 40  
 C ON THE SAME POSITION A TALLY IS KEPT AND THE NUMBER IS PRINTED. PLO 50  
 C THE USER MAY PROVIDE THE BOUNDS ON THE X,Y COORDINATES. PLO 60  
 C IF BOUNDS ARE PROVIDED, THEY MUST APPEAR IN PAIRS AS READ NOS. IF A PLO 70  
 C PAIR OF REAL NOS ARE EQUAL THE PROGRAM ASSUMES THERE ARE NO BOUNDS PLO 80  
 C COMMANDS FOR USING THIS PLOT ARE AS FOLLOWS PLO 90  
 C FOR THE AXIS THAT PAIR REPRESENTS AND THE BOUNDS WILL BE CALCULAT-PLO 100  
 C ED. PLO 110  
 C COMMANDS FOR USING PLOT ARE AS FOLLOWS PLO 120  
 C I PLOT Y +++,+++,.... X +++ PLO 130  
 C II PLOT Y +++,+++,....,(YMIN,YMAX) X +++ (XMIN,XMAX) PLO 140  
 C III PLOT Y +++,+++,....,(YMIN,YMAX) X ++1 PLO 150  
 C IV PLOT Y +++,+++,.... V +++ (XMIN,XMAX) PLO 160  
 C V PLOT Y +++,+++,.... X (XMIN,XMAX) (YMIN,YMAX) PLO 170  
 C PLO 180  
 C ERRORS PLO 190  
 C I WHEN TYPE II COMMAND IS USED THERE MUST BE TWO PAIRS OF REAL PLO 200  
 C NOS. OTHERWISE THE FOLLOWING MESSAGE IS PRINTED PLO 210  
 C ' Y BOUNDS ARE NOT SET UP CORRECTLY' PLO 220  
 C I IF BOUNDS ARE PROVIDED, THEN THERE MUST BE FOUR REAL NOS. PLO 230  
 C II IF A SINGLE REAL NO. APPEARS AHEAD OF COLUMN NOS., THE FOLLOW-PLO 240  
 C ING MESSAGE WILL BE PRINTED AND NO PLOTTING WILL TAKE PLACE PLO 250  
 C ' Y BOUNDS ARE NOT SET UP CORRECTLY' PLO 260  
 C III IF A PLOT COMMAND ENDS WITH ONE REAL NO, THE FOLLOWING MESSAGE PLO 270  
 C WILL BE PRINTED AND PLOTTING WILL BE TERMINATED PLO 280  
 C ' X BOUNDS ARE NOT SET UP CORRECTLY' PLO 290  
 C IV IF MORE THEN 5 PLOTS ARE REQUESTED PER GRAPH, NO GRAPH WILL BE PLO 300  
 C PRODUCED AND FOLLOWING MESSAGE WILL BE PRINTED. PLO 310  
 C ' MORE THEN 5 PLOTS WERE REQUESTED PER GRAPH' PLO 320  
 COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG PLO 330  
 COMMON /BLOCRC/ NRC,RC(12600) PLO 340  
 COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NPLO 350  
 IARGS,VWXYZ(8),NERROR PLO 360  
 DIMENSION ARGS(100) PLO 370  
 EQUIVALENCE (ARGS(1),RC(12501)) PLO 380  
 COMMON /SCRAT/ NS,NS2,A(13500) PLO 390  
 COMMON /HEADER/ NOCARD(80),ITLE(60,6),LNCT,IPRINT,NPAGE,IPUNCH PLO 400  
 COMMON /FMAT/ IFMTX(6),IOSWT,IFMTS(6),LHEAD(96) PLO 410  
 DIMENSION TIT(60), TITX(60) PLO 420  
 EQUIVALENCE (TIT,ITLE(1,6)), (TITX,ITLE(1,5)) PLO 430  
 DIMENSION X(1), KCCL(6), PRINT(101), XP(6), BOOL(5), IDGT(9) PLO 440  
 EQUIVALENCE (RC(1),X(1)), (PRINT,A) PLO 450  
 INTEGER PRINT,BLANK PLO 460  
 EQUIVALENCE (X0,XMIN), (X1,XMAX), (Y0,YMIN), (Y1,YMAX) PLO 470  
 DIMENSION IH(12,8), IPR(101) PLO 480  
 EQUIVALENCE (LHEAD,IH), (IPR,A(200)) PLO 490  
 INTEGER BOOL PLO 500  
 DATA BOOL(1),BOOL(2),BOOL(3),BOOL(4),BOOL(5)/1H.,1H\*,1H+,1H,,1H-/ ,PLO 510  
 1BLANK/1H / PLO 520  
 DATA IDGT(1),IDGT(2),IDGT(3),IDGT(4),IDGT(5),IDGT(6),IDGT(7),IDGT(PLO 530  
 18),IDGT(9)/1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1HX/ PLO 540  
 C INITIAL SWITCHES PLO 550  
 DATA IXPR/1HX/ PLO 560  
 IF (NRMAX.GT.0) GO TO 10 PLO 570  
 CALL ERROR (9) PLO 580  
 RETURN PLO 590

```

10  ISWT=1          PLO 600
    ISWT1=0         PLO 610
    XUP=1.E+35      PLO 620
    XDOWN=-1.E+35   PLO 630
    YUP=1.E+35      PLO 640
    YDOWN=-1.E+35   PLO 650
    NCN=0           PLO 660
    IPT=100          PLO 670
    IPTX=101         PLO 680
    IF (L2.NE.6) GO TO 20
    IPT=60           PLO 690
    IPTX=61           PLO 700
20  IF (NARGS.EQ.2) GO TO 50          PLO 710
    IF (KIND(NARGS).EQ.0) GO TO 40
    IF (KIND(NARGS)-KIND(NARGS-1).NE.0) GO TO 680
C   X OR Y BOUNDS ARE PROVIDED      PLO 720
    IF (KIND(NARGS-2).EQ.0) GO TO 30
    IF (KIND(NARGS-3).EQ.0) GO TO 680
    ISWT=5           PLO 730
    YUP=ARGS(NARGS)
    YDOWN=ARGS(NARGS-1)          PLO 740
    XUP=ARGS(NARGS-2)          PLO 750
    XDOWN=ARGS(NARGS-3)          PLO 760
    NARGS=NARGS-4           PLO 770
    GO TO 50           PLO 780
C   X BOUNDS ARE PROVIDED          PLO 790
30  ISWT=3           PLO 800
    XUP=ARGS(NARGS)
    XDOWN=ARGS(NARGS-1)          PLO 810
    NARGS=NARGS-2           PLO 820
    IF (NARGS.EQ.2) GO TO 50
C   CHECK TO SEE IF THERE ARE Y BOUNDS PLO 830
40  IF (KIND(NARGS-1)-KIND(NARGS-2).NE.0) GO TO 670
    IF (KIND(NARGS-1).EQ.0) GO TO 50
C   Y LIMITS ARE PROVIDED          PLO 840
    ISWT=ISWT+1          PLO 850
    YUP=ARGS(NARGS-1)
    YDOWN=ARGS(NARGS-2)          PLO 860
    IARGS(NARGS-2)=IARGS(NARGS)
    KIND(NARGS-2)=0          PLO 870
    NARGS=NARGS-2           PLO 880
    DO 60 I=1,NARGS          PLO 890
    KCCL(I)=IARGS(I)
    M=NARGS-1           PLO 900
    IF (NARGS.GT.6) GO TO 710
    CALL CHKCOL (J)
    IF (J.GT.0) GO TO 690
C   NO ERROR FOUND IN COLUMN NOS.
    IF (NERROR.GE.1) RETURN
C   SEARCH FOR MAX AND MIN ON AXIS, IF BOUNDS ARE NOT PROVIDED,
C   OTHERWISE TALLY NO OF POINTS THAT FALL OUTSIDE OF BOUNDS .
    IF (XUP.GE.XDOWN) GO TO 70          PLO 910
    XAP=XDOWN
    XAN=XUP
    GO TO 80
70  XAP=XUP
    XAN=XDOWN
    IF (YUP.GE.YDOWN) GO TO 90          PLO 920
    YAP=YDOWN

```

	YAN=YUP	PL01190
	GO TO 100	PL01200
90	YAP=YUP	PL01210
	YAN=YDOWN	PL01220
100	K1=IARGS(NARGS)	PL01230
	K2=K1-1+NRMAX	PL01240
	IF (ISWT-2) 110,720,150	PL01250
110	X1=X(K1)	PL01260
	X0=X1	PL01270
	DO 130 I=K1,K2	PL01280
	IF (X1.GE.X(I)) GO TO 120	PL01290
	X1=X(I)	PL01300
	GO TO 130	PL01310
120	IF (X0.LE.X(I)) GO TO 130	PL01320
	X0=X(I)	PL01330
130	CONTINUE	PL01340
140	XAP=X1	PL01350
	XAN=X0	PL01360
150	GO TO (170,270,160,270,270), ISWT	PL01370
160	KEY=2	PL01380
	GO TO 180	PL01390
170	KEY=1	PL01400
180	DO 260 J=1,M.	PL01410
	K1=IARGS(NARGS)	PL01420
	K3=IARGS(J)	PL01430
	K4=K3-1+NRMAX	PL01440
	IF (J.GT.1) GO TO 190	PL01450
	Y1=X(K3)	PL01460
	Y0=Y1	PL01470
	KY=1	PL01480
190	GO TO (200,220), KEY	PL01490
200	DO 210 I=K3,K4	PL01500
	IF (Y1.LT.X(I)) Y1=X(I)	PL01510
	IF (Y0.GT.X(I)) Y0=X(I)	PL01520
210	CONTINUE	PL01530
	GO TO 260	PL01540
220	DO 250 I=K3,K4	PL01550
	IF (X(K1).GE.XAN.AND.X(K1).LE.XAP) GO TO (230,240), KY	PL01560
	GO TO 250	PL01570
230	Y1=X(I)	PL01580
	Y0=X(I)	PL01590
	KY=2	PL01600
	GO TO 250	PL01610
240	IF (Y1.LT.X(I)) Y1=X(I)	PL01620
	IF (Y0.GT.X(I)) Y0=X(I)	PL01630
250	K1=K1+1	PL01640
260	CONTINUE	PL01650
	YAP=Y1	PL01660
	YAN=Y0	PL01670
	IF (ISWT.EQ.1) GO TO 800	PL01680
	GO TO 280	PL01690
270	Y1=YUP	PL01700
	Y0=YDOWN	PL01710
	ISWT1=1	PL01720
	IF (ISWT.EQ.2) GO TO 770	PL01730
280	X1=XUP	PL01740
	X0=XDOWN	PL01750
	GO TO 770	PL01760
C	DETERMINE X AND Y INCREMENTS FOR PLOT	PL01770

```

290  YDELTA=(YMAX-YMIN)/50.          PL01780
      K1=IARGS(NARGS)                PL01790
      XDELTA=(XMAX-XMIN)/FLOAT(IPT)   PL01800
      YL=YMAX-YDELTA/2.              PL01810
      YT=YMAX                         PL01820
      IF (ISWT.LE.1) GO TO 820        PL01830
      IF (L2.EQ.6) GO TO 300          PL01840
      WRITE (IPRINT,1050) NTOT,NCN    PL01850
      GO TO 820                      PL01860
300  WRITE (IPRINT,940) NTOT,NCN    PL01870
      GO TO 820                      PL01880
310  KYTL=1                         PL01890
      IF (YMAX.LT.YMIN) KYTL=2       PL01900
      KXTL=1                         PL01910
      IF (XMAX.LT.XMIN) KXTL=2       PL01920
      ITB=1                           PL01930
C   THE I LOOP CONTROLS THE 5 DIVISIONS OF THE Y ORDINATE  PL01940
      DO 620 I=1,6                  PL01950
      L=1                            PL01960
C   THE J LOOP IS FOR EACH LINE OF PRINT WITHIN THE DIVISIONS  PL01970
      DO 620 J=1,10                 PL01980
C   BLANK OUT PRINT BUFFER LINE.  PL01990
      DO 320 K=1,IPTX               PL02000
320  PRINT(K)=BLANK                 PL02010
C   THE KK INDEX IS FOR EACH CURVE.  KK LESS THAN 6.        PL02020
      DO 500 KK=1,M                 PL02030
      K3=IARGS(KK)                  PL02040
      K4=K3-1+NRMAX                 PL02050
      K5=K1                          PL02060
C   THIS DETERMINES IF Y(K) VALUE IS ON THE PRESENT PRINT LINE  PL02070
      DO 490 K=K3,K4                PL02080
      GO TO (330,350), KYTL         PL02090
330  IF (X(K)-YT) 340,340,490     PL02100
340  IF (X(K)-YL) 490,490,370     PL02110
350  IF (X(K)-YL) 360,360,490     PL02120
360  IF (X(K)-YT) 490,490,370     PL02130
C   YES. Y(K) BELONGS ON THIS PRINT LINE                     PL02140
C   THEREFORE DETERMIND WHERE ALL THE X(K5) FALL ON THE X-AXIS  PL02150
370  XL=XMIN                         PL02160
      XT=XMIN+XDELTA/2.             PL02170
      DO 480 KA=1,IPTX              PL02180
      GO TO (400,380), KXTL         PL02190
380  IF (X(K5)-XT) 470,470,390     PL02200
390  IF (X(K5)-XL) 420,420,470     PL02210
400  IF (X(K5)-XL) 470,410,410     PL02220
410  IF (X(K5)-XT) 420,470,470     PL02230
420  IF (PRINT(KA)-BLANK) 440,430,440  PL02240
430  PRINT(KA)=BOOL(KK)            PL02250
      GO TO 490                      PL02260
C   IF MORE THEN ONE POINT FALLS ON THE PRINT POSITION, TALLY THE NO.  PL02270
C   OF POINTS.                    PL02280
440  DO 450 KKK=1,9                 PL02290
      IF (PRINT(KA)-IDGT(KKK)) 450,460,450  PL02300
450  CONTINUE                        PL02310
      PRINT(KA)=IDGT(1)              PL02320
      GO TO 490                      PL02330
460  IF (PRINT(KA).NE.IDGT(9)) PRINT(KA)=IDGT(KKK+1)  PL02340
      GO TO 490                      PL02350
470  XL=XT                          PL02360

```

```

480 XT=XT+XDELTA          PL02370
490 K5=K5+1                 PL02380
500 CONTINUE                PL02390
      YP=YT*YL              PL02400
      YT=YL                  PL02410
      YL=YL-YDELTA           PL02420
      GO TO (510,550), L     PL02430
510 IF (I-5) 520,520,630    PL02440
520 L=2                     PL02450
      YS=YT+YDELTA/2.        PL02460
C THIS PATH IS EXECUTED ONCE IN EVERY DIVISION OF THE Y-AXIS. EVERY PL02470
C TENTH LINE, STARTING WITH ZERO LINE                         PL02480
      IF (L2.EQ.6) GO TO 570          PL02490
      IF (YP) 530,530,540          PL02500
530 WRITE (IPRINT,960) TIT(ITB),YS,PRINT          PL02510
      GO TO 620                  PL02520
540 WRITE (IPRINT,950) TIT(ITB),YS,PRINT          PL02530
      GO TO 620                  PL02540
550 IF (L2.EQ.6) GO TO 590          PL02550
      IF (YP) 560,560,610          PL02560
C PRINTS LINE          PL02570
560 WRITE (IPRINT,970) TIT(ITB),PRINT          PL02580
      GO TO 620                  PL02590
570 IF(YP.GT.0.0) GO TO 580          PL02600
      WRITE (IPRINT,980) TIT(ITB),YS,(PRINT(K),K=1,IPTX) PL02610
      GO TO 620                  PL02620
580 WRITE (IPRINT,990) TIT(ITB),YS,(PRINT(K),K=1,IPTX) PL02630
      GO TO 620                  PL02640
590 IF(YP.GT.0.0) GO TO 600          PL02650
      WRITE (IPRINT,1000) TIT(ITB),(PRINT(K),K=1,IPTX) PL02660
      GO TO 620                  PL02670
600 WRITE (IPRINT,1010) TIT(ITB),(PRINT(K),K=1,IPTX) PL02680
      GO TO 620                  PL02690
610 WRITE (IPRINT,1020) TIT(ITB),PRINT          PL02700
620 ITB=ITB+1                 PL02710
630 IF (L2.EQ.6) GO TO 910          PL02720
      IF (YP) 640,640,650          PL02730
640 WRITE (IPRINT,960) TIT(51),YMIN,PRINT          PL02740
      GO TO 660                  PL02750
650 WRITE (IPRINT,950) TIT(51),YMIN,PRINT          PL02760
C LAST LINE OF PRINT OUT PLUS X VALUES ALONG X-AXIS.          PL02770
660 WRITE (IPRINT,1060) IPR          PL02780
      WRITE (IPRINT,1030) XP          PL02790
      WRITE (IPRINT,1070) TITX          PL02800
      RETURN          PL02810
C THIS PRINTS OUT THAT 'Y BOUNDS ARE NOT SET UP CORRECTLY'. PL02820
670 CONTINUE                PL02830
C THIS PRINTS OUT THAT 'X BOUNDS ARE NOT SET UP CORRECTLY'. PL02840
680 CONTINUE                PL02850
C THIS PRINT 'COL. NOS. APPEAR AS ARGUMENTS'.          PL02860
690 CALL ERROR (20)          PL02870
700 NERROR=NERROR-1          PL02880
      RETURN          PL02890
C THIS PRINTS THAT 'MORE THEN 5 PLOTS WERE REQUISTED PER GRAPH'. PL02900
710 CALL ERROR (10)          PL02910
      GO TO 700          PL02920
720 KEY=1                     PL02930
      DO 760 IK=1,M          PL02940
      IKK=IARGS(IK)          PL02950

```

	DO 750 I=K1,K2	PL02960
	IF (X(IKK).GE.YAN.AND.X(IKK).LE.YAP) GO TO (730,740), KEY	PL02970
	GO TO 750	PL02980
730	X1=X(I)	PL02990
	X0=X1	PL03000
	KEY=2	PL03010
	GO TO 750	PL03020
740	IF (X1.LT.X(I)) X1=X(I)	PL03030
	IF (X0.GT.X(I)) X0=X(I)	PL03040
750	IKK=IKK+1	PL03050
760	CONTINUE	PL03060
	IF (KEY.EQ.2) GO TO 140	PL03070
	X0=XDOWN	PL03080
	X1=XUP	PL03090
	GO TO 140	PL03100
770	DO 790 J=1,M	PL03110
	K1=IARGS(NARGS)	PL03120
	K3=IARGS(J)	PL03130
	K4=K3-1+NRMAX	PL03140
	DO 790 I=K3,K4	PL03150
	IF (X(I).GT.YAP.OR.X(I).LT.YAN) GO TO 780	PL03160
	IF (X(K1).LE.XAP.AND.X(K1).GE.XAN) GO TO 790	PL03170
780	NCN=NCN+1	PL03180
790	K1=K1+1	PL03190
	NTOT=M*NRMAX-NCN	PL03200
C	DETERMINE TYPE OF HEADINGS TO BE PRINTED	PL03210
800	CALL HEADS (KCCL,NARGS,0,1)	PL03220
	K=4	PL03230
	IF (L2.EQ.6) K=1	PL03240
	CALL PAGE (K)	PL03250
	IF (L2.EQ.6) GO TO 810	PL03260
	WRITE (IPRINT,1040) (IH(I,NARGS),I=1,12),((IH(I,J),I=1,12),BOOL(J)	PL03270
	1,J=1,M)	PL03280
	GO TO 290	PL03290
810	WRITE (IPRINT,1080) (IH(I,NARGS),I=1,12),((IH(I,J),I=1,12),BOOL(J)	PL03300
	1,J=1,M)	PL03310
	GO TO 290	PL03320
820	XP(1)=XMIN	PL03330
	XP(6)=XMAX	PL03340
	XR=20.*XDELTA	PL03350
	DO 830 I=2,5	PL03360
830	XP(I)=XP(I-1)+XR	PL03370
	DO 840 J=1,100	PL03380
840	IPR(J)=BOOL(5)	PL03390
	DO 850 I=1,101,10	PL03400
850	IPR(I)=BOOL(3)	PL03410
	IF (XMIN*XMAX.GE.0.0) GO TO 900	PL03420
	J=0	PL03430
	DO 860 I=2,5	PL03440
	IF (XP(I-1)*XP(I)) 870,890,860	PL03450
860	CONTINUE	PL03460
	N=IPTX	PL03470
	GO TO 890	PL03480
870	XXP=XP(I-1)+XDELTA	PL03490
	DO 880 J=1,20	PL03500
	IF (XP(I-1)*XXP.LE.0.0) GO TO 890	PL03510
880	XXP=XXP+XDELTA	PL03520
	J=20	PL03530
890	N=(I-2)*20+J	PL03540

```

IPR(N)=IXPR PL03550
900  WRITE (IPRINT,1060) (IPR(K),K=1,IPTX) PL03560
      GO TO 310 PL03570
910  IF(YP.GT.0.0) GO TO 920 PL03580
      WRITE (IPRINT,980) TIT(51),YMIN,(PRINT(K),K=1,IPTX) PL03590
      GO TO 930 PL03600
920  WRITE (IPRINT,990) TIT(51),YMIN,(PRINT(K),K=1,IPTX) PL03610
930  WRITE (IPRINT,1060) (IPR(K),K=1,IPTX) PL03620
      WRITE (IPRINT,1030) (XP(K),K=1,4) PL03630
      WRITE (IPRINT,1090) TITX PL03640
      RETURN PL03650
C PL03660
940  FORMAT (21H NO. OF PTS. PLOTTED ,I5,33H NO. NOT PLOTTED (OUT OF BOPPL03670
1UNDS) ,I5) PL03680
950  FORMAT (1X,A1,1PE11.4,1H+,101A1,1H+) PL03690
960  FORMAT (1X,A1,1PE11.4,1HX,101A1,1HX) PL03700
970  FORMAT (1X,A1,11X,1HX,101A1,1HX) PL03710
980  FORMAT (1X,A1,1PE11.4,1HX,61A1,1HX) PL03720
990  FORMAT (1X,A1,1PE11.4,1H+,61A1,1H+) PL03730
1000 FORMAT (1X,A1,11X,1HX,61A1,1HX) PL03740
1010 FORMAT (1X,A1,11X,1H-,61A1,1H-) PL03750
1020 FORMAT (1X,A1,11X,1H-,101A1,1H-) PL03760
1030 FORMAT (6(7X,1PE13.4)) PL03770
1040 FORMAT (6H ABS-,12A1,6H,ORD-,5(12A1,2H (,A1,3H), )) PL03780
1050 FORMAT (29H TOTAL NO. OF PTS. PLOTTED IS,I5,60H AND NO. NOT PLOTTEPL03790
1D BECAUSE THEY FALL OUTSIDE OF BOUNDS IS,I5) PL03800
1060 FORMAT (14X,101A1) PL03810
1070 FORMAT (34X,60A1) PL03820
1080 FORMAT (6H ABS-,12A1/6H ORD-,5(12A1,2H (,A1,3H), )) PL03830
1090 FORMAT (14X,60A1) PL03840
      END PL03850

```

```

C SUBROUTINE PREPAK (N,IND,IA,LOC,LH)          PRE 10
C VERSION 5.00      PREPAK    5/15/70          PRE 20
C     S. PEAVY     8/ 5/69          PRE 30
C THIS SUBROUTINE DOES THE FOLLOWING:          PRE 40
C N=1  PACK FORMAT IN IFMT          PRE 50
C N=2  PACK HEAD   IN IHEAD. IF MORE THAN 50 HEADINGS ARE STORED, PRE 60
C           STACK OF HEADINGS IS PUSHED DOWN AND BOTTOM ONES      PRE 70
C           DISCARDED          PRE 80
C N=3  CLEAR VARIABLES IFMT AND IHEAD          PRE 90
C N=4  PICK UP PROPER FORMAT NO. IN IA AND STORE IN LH          PRE 100
C N=5  UNPACKS THE HEADING OF LOC. INTO LH IA A1 LEFT JUSTIFIED PRE 110
C IND INDICATOR. IF IND =0 CALL TO PREPAK WAS O.K.          PRE 120
C           IF IND =1 A FLAG THAT RESULTS WERE NOT OBTAINED      PRE 130
C           PRE 140
C IA   COL NUMBER FOR THE HEADING TO BE PACKED OR FORMAT DESIRED PRE 150
C           PRE 160
C LOC  LOC CONTAINS THE COLUMN NUMBER WHOSE HEADING THE SUBROUTINE PRE 170
C           IS TRYING TO FIND.          PRE 180
C           PRE 190
C LH   IS WHERE THE HEADING WILL BE STORED AS A1 LEFT JUSTIFIED PRE 200
C           AFTER IT IS UNPACKED, IF THE TITLE IS FOUND.          PRE 210
C           OR          PRE 220
C           WHERE FORMAT WILL BE STORED IF N=4          PRE 230
COMMON /ABCDEF/ L(48)          PRE 240
COMMON /BLOCKA/ MODE,M,KARD(83),KARG,ARG,ARG2,NEWCD(80),KRDEND PRE 250
COMMON /PKSWT/ IHCNT,IHTP          PRE 260
COMMON /BLOCKC/ KIO,INUNIT,ISCRAT,KBDOUT,KRDKN,LLIST          PRE 270
COMMON /SCRAT/ NS,NS2,A(13500)          PRE 280
DIMENSION IAA(80)          PRE 290
EQUIVALENCE (A,IAA)          PRE 300
DIMENSION LH(1)          PRE 310
C ****PRE 320
C           PRE 330
C THE VARIABLE IFMT CONTAINS THE INFORMATION FOR 6 FORMATS          PRE 340
C (I E. FORMAT A THUR F). THE MAXIMUM LENGTH FOR EACH FORMAT IS PRE 350
C 72 CHARACTERS INCLUDING LEFT AND RIGHT PARENTASIS. IF FORMATS ARE PRE 360
C PACKED DIFFERENTLY THEN STATED BELOW , THE DIMENSION SIZE OF PRE 370
C THE FIRST (12) CONSTANT MUST BE CHANGED TO BE EQUAL OR GREATER PRE 380
C THAN 72/(NO. OF CHARACTERS PER WORD)+M. SEE NOTE BELOW FOR M VALUEPRE 390
C CAUTION: FORMATS MUST BE PACKED IF NH CONVERSION IS PERMITTED      PRE 400
C           PRE 410
C THE VARIABLE IHEAD (5,LA) CONTAINS THE HEADINGS FOR LA COLUMNS. PRE 420
C MAXIMUN NO. OF CHARACTERS PER HEADING IS 12.          PRE 430
C           PRE 440
C DIMENSION IFMT(12,6), IHEAD(5,50)          PRE 450
C           PRE 460
C IF THE VARIABLES IFMT(II,6),IHEAD(5,LA) ARE REDIMENSIONED SO THAT PRE 470
C     II DOES NOT = 12          PRE 480
C AND LA DOES NOT = 50          PRE 490
C THEN THE FOLLOWING DATA STATEMENT MUST BE CHANGED          PRE 500
C           PRE 510
C DATA II/12/,LA/50/          PRE 520
C           PRE 530
C FORMAT 90 MUST BE CHANGED IF MORE OR FEWER CHARACTERS CAN BE      PRE 540
C PACKED INTO A WORD. 90 FORMAT (IIAK) WHERE II IS DEFINED ABOVE ANDPRE 550
C K =12/(CHARACTERS PER WORD)+M.          PRE 560
C     WHERE M=0 IF 12/(NO. OF CHAR/WORD) HAS NO REMAINDER          PRE 570
C     AND M=1 IF 12/(NO. OF CHAR/WORD) HAD A REMAINDER          PRE 580
C ****PRE 590

```

```

IF (N.GT.5) GO TO 60          PRE 600
IND=0                         PRE 610
GO TO (10,70,150,180,200), N  PRE 620
C THIS PART OF PROGRAM PACKS AND STORES FORMAT  PRE 630
10 K=KARD(M)                  PRE 640
15 M=M+1                      PRE 650
   IF (KARD(M).NE.41) IF (KARD(M)-46) 15,60,15  PRE 655
   KK=1                         PRE 660
   KA=0                         PRE 670
   MA=M+1                      PRE 680
   KR=KRDEND+3                 PRE 690
   DO 20 I=MA,KR               PRE 700
   IF (KARD(I).EQ.41) KK=KK+1  PRE 710
   IF (KARD(I).NE.42) GO TO 20  PRE 720
   KA=KA+1                     PRE 730
   IF (KA.EQ.KK) GO TO 30     PRE 740
20  CONTINUE                   PRE 750
   GO TO 60                     PRE 760
30  MB=M-2                     PRE 770
   IM=I-M+1                    PRE 780
   DO 40 JA=1,80               PRE 790
40  IAA(JA)=L(45)              PRE 800
   DO 50 JA=1,IM               PRE 810
   IAA(JA)=NEWCD(MB)          PRE 820
50  MB=MB+1                    PRE 830
   WRITE (ISCRAT,240) (IAA(JA),JA=1,80)        PRE 840
   BACKSPACE ISCRAT           PRE 850
   READ (ISCRAT,230) (IFMT(JA,K-9),JA=1,II)    PRE 860
   BACKSPACE ISCRAT           PRE 870
   RETURN                      PRE 880
60  IND=IND+1                  PRE 890
   RETURN                      PRE 900
C THIS PACKS HEADS            PRE 910
70  IF (IHCNT.GE.IHTP) GO TO 140  PRE 920
   IF (IHCNT.EQ.0) GO TO 110    PRE 930
C IHEAD(1,I)=COL NUMBER FOR THAT HEADING  PRE 940
   DO 80 I=1,IHCNT             PRE 950
   IF (IA.EQ.IHEAD(1,I)) GO TO 130  PRE 960
80  CONTINUE                   PRE 970
   KB=IHCNT                   PRE 980
90  DO 100 I=1,KB              PRE 990
   KA=KB-I+2                  PRE1000
   DO 100 K=1,5                PRE1010
100 IHEAD(K,KA)=IHEAD(K,KA-1)  PRE1020
110 IHCNT=IHCNT+1              PRE1030
120 IHEAD(1,1)=IA              PRE1040
   ICHAR=12                    PRE1042
   DO 122 I=2,5                PRE1044
C THE FOLLOWING CONSTANT IS (45*256+45)*256+45  PRE1046
122 IHEAD (I,1)=2960685      PRE1050
   IF(M+12.GT.KRDEND+3) ICHAR=KRDEND+2-M  PRE1052
   CALL PACK(KARD(M+1),IHEAD(2,1),ICHAR,0)  PRE1054
   RETURN                      PRE1060
130 IF (I.EQ.1) GO TO 120      PRE1070
   KB=I-1                      PRE1080
   GO TO 90                     PRE1090
140 KB=IHTP-1                  PRE1100
   CALL ERROR (229)             PRE1105
   GO TO 90                     PRE1110

```

```

C      N=3   CLEAR IFMT, IHEAD                         PRE1120
150    DO 160  I=1,6                                     PRE1130
      DO 160  IA=1,II                                    PRE1140
160    IFMT(IA,I)=0                                     PRE1150
      DO 170  I=1,LA                                    PRE1160
      DO 170  IA=1,5                                     PRE1170
170    IHEAD(IA,I)=0                                    PRE1180
      RETURN                                           PRE1190
C      FIND PROPER FORMAT                            PRE1200
180    IF (IA.LT.2.OR.IA.GT.7) GO TO 60               PRE1210
      IF (IFMT(1,IA-1).EQ.0) GO TO 60               PRE1220
      DO 190  I=1,II                                    PRE1230
190    LH(I)=IFMT(I,IA-1)                           PRE1240
      RETURN                                           PRE1250
C      SEARCH FOR HEADING AND UNPACK                  PRE1260
C      IF HEADING IS FOUND IND=0, OTHERWISE 1.        PRE1270
200    DO 210  I=1,IHCNT                           PRE1280
      IF (LOC.EQ.IHEAD(1,I)) GO TO 220              PRE1290
210    CONTINUE                                         PRE1300
C      NO HEADING FOUND                           PRE1310
      GO TO 60                                         PRE1320
220    CALL PACK (IHEAD(2,I),LH,12,1)                PRE1330
      RETURN                                           PRE1340
C
230    FORMAT (12A6)                                 PRE1350
240    FORMAT (80A1)                                 PRE1360
      END                                              PRE1370
                                                PRE1380

```

```

SUBROUTINE PRINTX                               PRI 10
C      VERSION 5.00      PRINTX      5/15/70      PRI 20
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG          PRI 30
COMMON /BLOCRC/ NRC,RC(12600)                 PRI 40
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NPRI 50
IARGS,VWXYZ(8),NERROR                      PRI 60
DIMENSION ARGS(100)                         PRI 70
EQUIVALENCE (ARGS(1),RC(12501))             PRI 80
COMMON /HEADER/ NOCARD(80),ITLE(60,6),LNCNT,IPRINT,NPAGE,IPUNCH  PRI 90
COMMON /FMAT/ IFMTX(6),IOSWT,IFMTS(6),LHEAD(96)  PRI 100
COMMON /KFMT/ KFMT(100)                      PRI 110
C
C THIS SUBROUTINE IS CALLED TO EXECUTE THE FOLLOWING      PRI 120
C     PRINT // COLS ++,++,++,++ ,ETC (PRINT WITH FORMAT//)  PRI 130
C     PRINT    COLS ++,++,++,++   USE RPRINT IF ALL ARGS ARE INTEGER  PRI 140
C                                     UNLESS IOSWT IS ON           PRI 150
C     PRINT (USING ARGS AS IN RPRINT) ALWAYS USE RPRINT  PRI 160
C
C     L1=2 PRINT                                         PRI 170
C     L1=8 NPRINT                                       PRI 180
C
IF (NARGS.NE.0) GO TO 45                      PRI 183
10 CALL ERROR(205)                            PRI 190
20 RETURN                                     PRI 200
30 CALL ERROR (222)                           PRI 210
40 IF (NPAGE.EQ.0 .AND.L1.EQ.8) CALL PAGE(0)  PRI 220
CALL RPRINT                                    PRI 225
RETURN                                         PRI 230
45 IF (L2.EQ.1) IF (IOSWT) 40,40,230        PRI 240
CALL PREPAK (4,IND,L2,IND,KFMT)              PRI 249
IF (IND.NE.0) GO TO 30                      PRI 250
IP=1                                         PRI 260
50 CALL CHKCOLUMN (I)                        PRI 270
IF (I.NE.0) GO TO 10                         PRI 280
IF (NERROR.NE.0) RETURN                      PRI 290
IB=0                                         PRI 300
IA=1                                         PRI 305
ICP=0                                         PRI 310
GO TO (60,70), IP                           PRI 315
60 IB=NARGS                                    PRI 320
GO TO 100                                     PRI 330
70 IBB=NARGS                                   PRI 340
80 IF (IBB.GT.8) GO TO 90                     PRI 350
IB=IBB+IB                                     PRI 360
IC=IBB                                         PRI 370
IBB=0                                         PRI 380
GO TO 100                                     PRI 390
90 IBB=IBB-8                                   PRI 400
IB=8+IB                                      PRI 410
IC=8                                         PRI 420
100 LL=NRMAX                                    PRI 430
110 IF (LL.GT.51) GO TO 120                   PRI 440
J=LL                                         PRI 450
LL=0                                         PRI 460
GO TO 130                                     PRI 470
120 LL=LL-50                                    PRI 480
J=50                                         PRI 490
130 IF (L1.EQ.8) IF(NPAGE) 155,260,155       PRI 500
CALL PAGE (4)                                 PRI 510
GO TO (150,140), IP                           PRI 520
                                         PRI 530
                                         PRI 540

```

```

140 CALL HEADS (KFMT(IA),IC,ICP,0)          PRI 550
150 WRITE (IPRINT,250)                      PRI 560
155 DO 200 M=1,J                           PRI 570
    DO 160 I=IA,IB                         PRI 580
    K=IARGS(I)                            PRI 590
    IARGS(I)=IARGS(I)+1                  PRI 600
160  ARGS(I)=RC(K)                         PRI 610
    GO TO (170,180), IP                   PRI 620
170  WRITE (IPRINT,KFMT) (ARGS(I),I=1,NARGS) PRI 630
    GO TO 200                            PRI 640
180  WRITE (IPRINT,IFMTX) (ARGS(I),I=IA,IB) PRI 650
190  IF (MOD(M,10).EQ.0) WRITE (IPRINT,250) PRI 660
200  CONTINUE                             PRI 670
    ICP=1                                PRI 680
    IF (LL) 210,210,110                  PRI 690
210  GO TO (20,220), IP                   PRI 700
220  IF (IBB.EQ.0) GO TO 20              PRI 710
    IF (L1.EQ.8) WRITE(IPRINT,250)        PRI 713
    IA=IB+1                            PRI 715
    ICP=0                                PRI 720
    GO TO 80                            PRI 730
C   USE STANDARD OR SPECIFIED FORMAT      PRI 820
230  IP=2                                 PRI 830
    DO 240 I=1,NARGS                    PRI 840
240  KFMT(I)=IARGS(I)                  PRI 850
    GO TO 50                            PRI 860
C
250  FORMAT (1X)                         PRI 870
260  CALL PAGE (0)                      PRI 880
    GO TO 155                           PRI 890
END                                     PRI 900
                                         PRI 910

```

```

C      SUBROUTINE PROB (VNU1,VNU2,F,Q)          PRB   10
      VERSION 5.00      PROB      5/15/70        PRB   20
      DOUBLE PRECISION FDSIN,FDCOS,FDEXP,FDLOG    PRB   30
      DOUBLE PRECISION C,A,X,W,ONE,B,TA,TB,G       PRB   40
      DATA C/.6366197723675814D0/,EP/1.E-5/,ONE/1.D0/,TWO/2.0/,ONEP/1./,PRB 50
      1P5/.5/
      NU1=VNU1+EP                                PRB   60
      NU2=VNU2+EP                                PRB   70
      V1=NU1                                     PRB   80
      V2=NU2                                     PRB  100
      IF (ABS(V1-VNU1).GT.EP) GO TO 200         PRB  110
      IF (ABS(V2-VNU2).GT.EP) GO TO 200         PRB  120
      IF (F) 5,6,10                               PRB  130
      5     F=0.0                                    PRB  140
C     ERROR , SET F=0 SINCE F LESS THEN 0,      PRB  150
      CALL ERROR (206)                           PRB  160
      6     Q=1.0                                    PRB  162
      RETURN                                     PRB  164
      10    IF (NU1.LT.0.OR.NU2.LT.0) GO TO 190    PRB  170
      20    MNU1=MOD(NU1,2)                         PRB  180
      MNU2=MOD(NU2,2)                         PRB  190
      IF (MNU2.NE.0) GO TO 70                   PRB  200
      I1=N2/2-1                                 PRB  210
      X=V2/(V2+V1*F)                           PRB  220
      V4=V1/TWO                                 PRB  230
      I4=NU1                                     PRB  240
      30    A=ONE                                    PRB  250
      IF (I1.EQ.0) GO TO 50                     PRB  260
      W=A                                       PRB  270
      DO 40 I=1,I1                            PRB  280
      T=I                                       PRB  290
      W=((V4+T-ONEP)/T)*X*W                  PRB  300
      40    A=A+W                                 PRB  310
      50    Q=A*FDEXP(V4*FDLOG(ONE-X))           PRB  320
      IF (I4.EQ.NU1) Q=ONEP-Q                 PRB  330
      60    IF (Q.LT.0.) Q=0                      PRB  340
      IF (Q.GT.ONEP) Q=ONEP                   PRB  350
      RETURN                                     PRB  360
      70    IF (MNU1.NE.0) GO TO 80              PRB  370
      I1=N1/2-1                                 PRB  380
      X=ONEP-V2/(V2+V1*F)                     PRB  390
      V4=V2/TWO                                PRB  400
      I4=NU2                                     PRB  410
      GO TO 30                                  PRB  420
      80    IF (NU2.NE.1) GO TO 130             PRB  430
      IF (NU1.NE.1) GO TO 90                   PRB  440
      Q=C*ATAN(ONEP/FSQRT(F))                PRB  450
      GO TO 60                                  PRB  460
      90    X=ATAN(FSQRT(V2/(V1*F)))           PRB  470
      I1=(NU1-3)/2                             PRB  480
      IS=1                                      PRB  490
      100   TB=FDSIN(X)                         PRB  500
      A=FDCOS(X)                                PRB  510
      IF (I1.EQ.0) GO TO 120                   PRB  520
      TA=A**2                                 PRB  530
      W=A                                       PRB  540
      DO 110 I=1,I1                            PRB  550
      V3=I                                     PRB  560
      W=V3/(V3+P5)*TA*W                        PRB  570

```

```

110 A=A+W PRB 580
120 A=C*(X+TB*A) PRB 590
Q=A PRB 600
GO TO (60,140), IS PRB 610
130 X=ATAN(FSQRT(V1*F/V2)) PRB 620
I1=(NU2-3)/2 PRB 630
IS=2 PRB 640
GO TO 100 PRB 650
140 IF (NU1.NE.1) GO TO 150 PRB 660
Q=ONE-A PRB 670
GO TO 60 PRB 680
150 I1=(NU1-3)/2 PRB 690
B=ONE PRB 700
IF (I1.EQ.0) GO TO 170 PRB 710
W=B PRB 720
DO 160 I=1,I1 PRB 730
V3=I PRB 740
W=(V2+TWO*V3-ONEP)/(TWO*V3+ONEP)*TB**2*W PRB 750
160 B=B+W PRB 760
170 G=C PRB 770
I1=(NU2-1)/2 PRB 780
DO 180 I=1,I1 PRB 790
V3=I PRB 800
180 G=(TWO*V3)/(TWO*V3-ONEP)*G PRB 810
W=1.0D0 PRB 820
X=FDCOS(X) PRB 830
IF (MOD(NU2,2).GT.0) W=DSIGN(W,X) PRB 840
Q=ONE-A+G*TB*W*FDEXP(FLOAT(NU2)*FDLOG(DABS(X)))*B PRB 850
GO TO 60 PRB 860
C PRINT ' EITHER NU1 OR NU2 IS LESS THEN 1 ' PRB 870
190 CALL ERROR (207) PRB 880
RETURN PRB 890
C PRINT ' EITHER NU1 OR NU2 IS NOT A INTEGER PROGRAM USES LARGEST PRB 900
C INTEGER CONTAINED ' PRB 910
200 CALL ERROR (208) PRB 920
GO TO 20 PRB 930
END PRB 940

```

```

SUBROUTINE PROCHK (A,NROW,N,NCOL,IVEC,X,NSIZE) PRK 10
C VERSION 5.00      PROCHK      5/15/70 PRK 20
C SUBROUTINE PROCHK      R VARNER      5/ 7/68 PRK 30
C R VARNER SUBROUTINE TO CHECK FOR DIAGONAL ,NORMAL,SYMMETRIC, PRK 40
C SKEW-SYMMETRIC AND ORTHOGONAL MATRIX PRK 50
C DIMENSION A(NROW,1), IVEC(1) PRK 60
C SET ALL INDICATORS TO NO CONDITION PRK 70
DO 10 I=1,5 PRK 80
10 IVEC(I)=2 PRK 90
C TEST TO SEE IF WE HAVE A DIAGONAL MATRIX PRK 100
C IF YES IVEC(1)=0   IF NO IVEC(1)=2 PRK 110
DO 30 I=1,N PRK 120
DO 30 J=1,N PRK 130
IF (I-J) 20,30,20 PRK 140
20 IF (A(I,J)) 40,30,40 PRK 150
30 CONTINUE PRK 160
IVEC(1)=0 PRK 170
IVEC(2)=0 PRK 180
IVEC(3)=0 PRK 190
GO TO 50 PRK 200
40 IVEC(1)=2 PRK 210
C CHECK FOR SUMMTRY PRK 220
CALL SYMV (A,NROW,N,IVEC(2)) PRK 230
C CHECK FOR SKEW SYMMETRY PRK 240
CALL SKSYMV (A,NROW,N,IRV) PRK 250
IF (IRV.GE.3) IVEC(2)=IRV PRK 260
IF (IVEC(2).EQ.2) GO TO 50 PRK 270
IVEC(3)=0 PRK 280
C CHECK FOR ORTHOGONAL MATRIX PRK 290
C IF A IS ORTHOGONAL IVEC(4)=0  OTHERWISE IVEC(4)=2 PRK 300
50 CALL ORTHRIV (A,NROW,N,NCOL,IVEC(4),X,NSIZE,X) PRK 310
RETURN PRK 320
END PRK 330

```

```

SUBROUTINE PROROW                               PRO 10
C VERSION 5.00      PROROW      5/15/70          PRO 20
C PROGRAMMED BY CARLA MESSINA MAY,1967          PRO 30
C L2 = 1, ROWSUM      L2 = 2, PRODUCT          PRO 40
COMMON /BLOCRC/ NRC,RC(12600)                  PRO 50
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NPRO 60
IARGS,VWXYZ(8),NERROR                         PRO 70
DIMENSION ARGS(100)                           PRO 80
EQUIVALENCE (ARGS(1),RC(12501))              PRO 90
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG          PRO 100
COMMON /SCRAT/ NS,NS2,A(13500)                PRO 110
IF (NARGS-3) 10,40,40                          PRO 120
10 IF (NARGS.EQ.1.AND.KIND(1).EQ.0) GO TO 230  PRO 130
K=10
20 CALL ERROR (K)                            PRO 140
30 RETURN                                     PRO 150
40 CALL CHKCOLUMN (J)                        PRO 170
IF (J) 50,60,50                                PRO 180
50 K=3                                         PRO 190
GO TO 20                                       PRO 200
60 IF (NRMAX) 70,70,80                          PRO 210
70 K=9                                         PRO 220
GO TO 20                                       PRO 230
80 IF (NERROR.NE.0) RETURN                     PRO 240
DO 100 I=1, NRMAX                            PRO 250
A(I)=0.0                                      PRO 260
GO TO (100,90), L2                           PRO 270
90 A(I)=1.0                                     PRO 280
100 CONTINUE                                    PRO 290
IF (NARGS-4) 110,190,190                      PRO 300
110 IF (IARGS(1)-IARGS(2)) 120,120,50          PRO 310
120 K=IARGS(1)                                  PRO 320
DO 150 I=1, NRMAX                            PRO 330
J=K+I-1                                      PRO 340
GO TO (130,140), L2                           PRO 350
130 A(I)=A(I)+RC(J)                           PRO 360
GO TO 150                                     PRO 370
140 A(I)=A(I)*RC(J)                           PRO 380
150 CONTINUE                                    PRO 390
IF (IARGS(1)+NROW-IARGS(2)) 160,160,170      PRO 400
160 IARGS(1)=IARGS(1)+NROW                      PRO 410
GO TO 120                                     PRO 420
170 K=IARGS(NARGS)                            PRO 430
DO 180 I=1, NRMAX                            PRO 440
J=K+I-1                                      PRO 450
180 RC(J)=A(I)                                PRO 460
GO TO 30                                      PRO 470
190 II=NARGS-1                                PRO 480
DO 220 L=1,II                                 PRO 490
K=IARGS(L)                                    PRO 500
DO 220 I=1, NRMAX                            PRO 510
J=K+I-1                                      PRO 520
GO TO (200,210), L2                           PRO 530
200 A(I)=A(I)+RC(J)                           PRO 540
GO TO 220                                     PRO 550
210 A(I)=A(I)*RC(J)                           PRO 560
220 CONTINUE                                    PRO 570
GO TO 170                                     PRO 580
230 CALL ADRESS (1,J)                          PRO 590

```

```
IF (J.LE.0) CALL ERROR (3)          PRO 600
DO 250 I=1,NRMAX                  PRO 610
IR=I                               PRO 620
SUM=0.                             PRO 630
DO 240 K=1,NCOL                   PRO 640
SUM=SUM+RC(IR)                    PRO 650
240 IR=IR+NROW                     PRO 660
RC(J)=SUM                          PRO 670
250 J=J+1                           PRO 680
RETURN                            PRO 690
END                                PRO 700
```

```

C SUBROUTINE PUNCH                                     PUN 10
C VERSION 5.00      PUNCH      5/15/70                 PUN 20
C                                                 PUN 30
C THE COMMAND PUNCH MAY BE USED IN THE FOLLOWING WAYS   PUN 40
C     PUNCH COL ++,++,++,++ (4 COLUMN LIMIT)           PUN 50
C     PUNCH // COL ++,++,++,++,ECT (ACCORDIND TO FORMAT //) PUN 60
C THIS SUBROUTINE IS USED TO EXECUTE WRITE TAPE          PUN 70
C     WRITE TAPE T FROM COL ++,++,++,++ (4 COLUMN LIMIT) PUN 80
C     WRITE TAPE T // FROM COL ++,++, ETC (USE FOLMAT//)  PUN 90
C                                                 PUN 100
C COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG                  PUN 110
C COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NPUN 120
1ARGS,VWXYZ(8),NERROR                                PUN 130
COMMON /BLOCRC/ NRC,RC(12600)                         PUN 140
DIMENSION ARGS(100)                                  PUN 150
EQUIVALENCE (ARGS(1),RC(12501))                      PUN 160
COMMON /KFMT/ KFMT(100)                               PUN 170
COMMON /FMAT/ IFMTX(6),IOSWT,IFMTS(6),LHEAD(96)       PUN 180
COMMON /TAPE/ NAME4(2),NTPCT,IPUNCP,INUNIP,LITP        PUN 190
COMMON /HEADER/ NOCARD(80),ITLE(60,6),LNCNT,IPRINT,NPAGE,IPUNCH PUN 200
C                                                 PUN 210
C L1=3 PUNCH L1=47 WRITE TAPE                         PUN 220
C                                                 PUN 230
IX=1                                              PUN 240
IF (L1.EQ.47) IX=2                                    PUN 250
IF (NARGS.NE.0) GO TO 20                            PUN 260
10 CALL ERROR (205)                                 PUN 270
RETURN                                            PUN 280
C                                                 PUN 290
C IF L2=1 ONLY 4 COLUMNS CAN BE PRINTED             PUN 300
C                                                 PUN 310
20 IF(L2.NE.1) GO TO 25                            PUN 320
NARGS=MIN0(NARGS,4)                                PUN 330
GO TO 30                                           PUN 340
25 CALL PREPAK (4,IND,L2,IND,KFMT)                 PUN 345
IF (IND.NE.0) GO TO 90                            PUN 350
30 CALL CHKCOL (I)                                PUN 360
IF (I.NE.0) GO TO 10                            PUN 370
IF (NERROR.NE.0) GO TO (110,100), IX            PUN 380
DO 60 I=1,NRMAX                                 PUN 390
DO 40 J=1,NARGS                                PUN 400
K=IARGS(J)                                       PUN 410
IARGS(J)=K+1                                    PUN 420
40 ARGS(J)=RC(K)                                PUN 430
IF (L2.NE.1) GO TO 50                            PUN 440
WRITE (IPUNCH,IFMTX) (ARGS(K),K=1,NARGS)         PUN 450
GO TO 60                                           PUN 460
50 WRITE (IPUNCH,KFMT) (ARGS(K),K=1,NARGS)         PUN 470
60 CONTINUE                                         PUN 480
IF (L1.NE.47) RETURN                           PUN 490
DO 70 J=1,NARGS                                PUN 500
70 ARGS(J)=0.0                                  PUN 510
IF (L2.NE.1) GO TO 80                            PUN 520
WRITE (IPUNCH,IFMTX) (ARGS(K),K=1,NARGS)         PUN 530
GO TO 100                                         PUN 540
80 WRITE (IPUNCH,KFMT) (ARGS(K),K=1,NARGS)         PUN 550
GO TO 100                                         PUN 560
90 CALL ERROR (222)                                PUN 570
L2=1                                              PUN 580

```

GO TO 30	PUN 590
100 IPUNCH=IPUNCP	PUN 600
110 RETURN	PUN 610
END	PUN 620

```

SUBROUTINE PVTRI (A,NROW,N,INDU,INDB)          PVT 10
C VERSION 5.00      PVTRI      5/15/70          PVT 20
C
C TO DETERMINE IF A IS AN UPPER OR LOWER TRIANGULAR MATRIX PVT 40
C S PEAVY FOR UNIVAC 1108    2/ 7/68          PVT 50
C
C A MATRIX TO BE CHECKED          PVT 60
C NROW- DIMENSION SIZE OF A          PVT 70
C N- PRESENT SIZE OF A          PVT 80
C INDU INDICATOR  INDU=0,UPPER TRIANGLE=0, INDU=1, UPPER TRIANGLE NOPVT 100
C           INDB=0,BOTTOM TRIAL =0, INDB=1, BOTTOM NOT ZEPVT 110
C
C DIMENSION A(NROW,NROW)          PVT 120
C INDU=1          PVT 130
C INDB=1          PVT 140
C NN=N-1          PVT 150
C DO 50 I=1,NN          PVT 160
C II=I+1          PVT 170
C DO 50 J=II,N          PVT 180
C GO TO (10,20), INDU          PVT 190
C 10 IF (A(I,J).NE.0.) INDU=2          PVT 200
C 20 GO TO (30,40), INDB          PVT 210
C 30 IF (A(J,I).NE.0.) INDB=2          PVT 220
C 40 IF (INDU.EQ.2.AND.INDB.EQ.2) GO TO 60          PVT 230
C 50 CONTINUE          PVT 240
C 60 INDU=INDU-1          PVT 250
C INDB=INDB-1          PVT 260
C RETURN          PVT 270
C END          PVT 280
C                      PVT 290

```

```

C      SUBROUTINE RANKO (N,X,H,R,T)          RKO 10
C      VERSION 45.0    RANKO     3/ 6/70        RKO 20
C      DIMENSION X(1), H(1), R(1)            RKO 30
C      *****
C      PUTS RANK OF N X'S IN VECTOR R. VECTOR H IS USED FOR STORAGE.   RKO 50
C      X,H AND R MUST BE DIMENSIONED N OR GREATER.                   RKO 60
C      STORES CORRECTION FOR TIES IN T = (1/12)*SUM(T-1)*T*(T+1).    RKO 70
C      T=0 MEANS NO TIES.                                         RKO 80
C      WRITTEN BY DAVID HOGBEN, SEL, NBS.    4/9/69.                 RKO 90
C      MOVE X TO R AND PUT I IN H           RKO 120
10     DO 20 I=1,N                         RKO 130
      H(I)=I                           RKO 140
20     R(I)=X(I)                         RKO 150
C      SORT X IN R, CARRY ALONG I IN H TO OBTAIN HIERARCHY IN H.    RKO 160
C      SORT USES 'PUSH-DOWN' METHOD. SEE ORGANICK, PAGE 84.       RKO 170
      K1=N-1                          RKO 180
      DO 30 I=1,K1                     RKO 190
      K2=N-I                          RKO 200
      DO 30 J=1,K2                     RKO 210
      IF (R(J).LE.R(J+1)) GO TO 30    RKO 220
      W=R(J)                           RKO 230
      R(J)=R(J+1)                      RKO 240
      R(J+1)=W                         RKO 250
      W=H(J)                           RKO 260
      H(J)=H(J+1)                      RKO 270
      H(J+1)=W                         RKO 280
30     CONTINUE                         RKO 290
C      REPLACE R(I) BY I*.                  RKO 300
C      LET K BE SUCH THAT R(I)=R(I-J+1),J=1,K. THEN I* = I-(K-1)/2. RKO 310
      K=1                             RKO 320
      T=0                            RKO 330
      DO 70 I=2,N                      RKO 340
      IF (R(I)-R(I-1)) 50,40,50       RKO 350
40     K=K+1                          RKO 360
      GO TO 70                         RKO 370
50     DO 60 J=1,K                      RKO 380
      IJ=I-J                          RKO 390
      R(IJ)=FLOAT(I-1)-FLOAT(K-1)/2.  RKO 400
      IF (K.GT.1) T=T+(FLOAT(K-1)*FLOAT(K)*FLOAT(K+1))/12.0       RKO 410
      K=1                            RKO 420
70     CONTINUE                         RKO 430
      T = T + (FLOAT(K-1)*FLOAT(K)*FLOAT(K+1))/12.0                RKO 440
      DO 80 I=1,K                      RKO 450
      K2=N+1-I                        RKO 460
80     R(K2)=FLOAT(N)-FLOAT(K-1)/2.0  RKO 470
C      SORT H CARRY ALONG R TO OBTAIN RANKS IN R                  RKO 480
      DO 90 I=1,K1                     RKO 490
      K2=N-I                          RKO 500
      DO 90 J=1,K2                     RKO 510
      IF (H(J).LE.H(J+1)) GO TO 90    RKO 520
      W=H(J)                           RKO 530
      H(J)=H(J+1)                      RKO 540
      H(J+1)=W                         RKO 550
      W=R(J)                           RKO 560
      R(J)=R(J+1)                      RKO 570
      R(J+1)=W                         RKO 580
90     CONTINUE                         RKO 590
      RETURN                           RKO 600
      END                            RKO 610

```

```

SUBROUTINE RANKS                                     RAS 10
C   VERSION 5.00      RANKS      5/15/70          RAS 20
C   *****          RAS 30
C   VERSION 3.05 COMMON                           RAS 40
C   COMMON /BLOCRC/ NRC,RC(12600)                  RAS 50
C   COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NRAS 60
C   IARGS,VWXYZ(8),NERROR                         RAS 70
C   DIMENSION ARGS(100)                            RAS 80
C   EQUIVALENCE (ARGS(1),RC(12501))               RAS 90
C   COMMON /SCRAT/ NS,NS2,A(13500)                 RAS 100
C   COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG           RAS 110
C   *****          RAS 120
C   RANKS OF VALUES IN COLUMN ++ PUT IN COLUMN ++. TIES ARE ALLOWED. RAS 130
C   ADJUSTMENT T=(1/12)*SUM(T-1)*T*(T+1) FOR FURTHER CALCULATIONS IS RAS 140
C   PUT IN ROW (NRMAX+1) IF NRMAX LT NROW.          RAS 150
C   *****          RAS 160
10    IF (NARGS.EQ.2) GO TO 20                      RAS 170
      CALL ERROR (10)                                RAS 180
      RETURN                                         RAS 190
20    CALL ADRESS (1,J1)                            RAS 200
      CALL ADRESS (2,J2)                            RAS 210
      IF (J1.GT.0.AND.J2.GT.0) GO TO 30            RAS 220
      CALL ERROR (3)                                RAS 230
      RETURN                                         RAS 240
30    IF (NRMAX.GT.0) GO TO 40                      RAS 250
      CALL ERROR (9)                                RAS 260
      RETURN                                         RAS 270
40    IF (NERROR.NE.0) RETURN                      RAS 280
      CALL RANKX (NRMAX,RC(J1),A(2),RC(J2),A(1)) RAS 290
      IF (NRMAX.GE.NROW) RETURN                     RAS 300
      JANR=J2+NRMAX                               RAS 305
      RC(JANR)=A(1)                                RAS 310
      RETURN                                         RAS 320
      END                                            RAS 330

```

```

SUBROUTINE RANKX (N,X,H,R,T)                                RAX 10
C VERSION 5.00      RANKX      5/15/70                      RAX 20
C DIMENSION X(1), H(1), R(1)                                RAX 30
C *****
C PUTS RANK OF N X'S IN VECTOR R. VECTOR H IS USED FOR STORAGE.   RAX 50
C X,H AND R MUST BE DIMENSIONED N OR GREATER.                  RAX 60
C STORES CORRECTION FOR TIES IN T = (1/12)*SUM(T-1)*T*(T+1).    RAX 70
C T=0 MEANS NO TIES.                                         RAX 80
C WRITTEN BY DAVID HOGBEN, SEL, NBS.   4/9/69.                 RAX 90
C COMPUTATION OF T CORRECTED 8/26/69                         RAX 95
C *****
C MOVE X TO R AND PUT I IN H                                RAX 100
10 DO 20 I=1,N                                              RAX 120
H(I)=I                                         RAX 130
20 R(I)=X(I)                                              RAX 140
C SORT X IN R, CARRY ALONG I IN H TO OBTAIN HIERARCHY IN H.   RAX 150
C SORT USES 'PUSH-DOWN' METHOD. SEE ORGANICK, PAGE 84.        RAX 160
K1=N-1                                              RAX 170
DO 30 I=1,K1                                         RAX 180
K2=N-I                                              RAX 190
DO 30 J=1,K2                                         RAX 200
IF (R(J).LE.R(J+1)) GO TO 30                         RAX 210
W=R(J)                                              RAX 220
R(J)=R(J+1)                                         RAX 230
R(J+1)=W                                         RAX 240
W=H(J)                                              RAX 250
H(J)=H(J+1)                                         RAX 260
H(J+1)=W                                         RAX 270
30 CONTINUE                                         RAX 280
C REPLACE R(I) BY I*.                                     RAX 290
C LET K BE SUCH THAT R(I)=R(I-J+1),J=1,K. THEN I* = I-(K-1)/2. RAX 300
K=1                                              RAX 310
T=0                                              RAX 320
DO 70 I=2,N                                         RAX 330
IF (R(I)-R(I-1)) 50,40,50                         RAX 340
40 K=K+1                                         RAX 350
GO TO 70                                         RAX 360
50 DO 60 J=1,K                                         RAX 370
IJ=I-J                                         RAX 375
60 R(IJ)=FLOAT(I-1)-FLOAT(K-1)/2.                   RAX 380
IF (K.GT.1) T=T+(FLOAT(K-1)*FLOAT(K)*FLOAT(K+1))/12.0 RAX 390
K=1                                              RAX 400
70 CONTINUE                                         RAX 410
T = T + (FLOAT(K-1)*FLOAT(K)*FLOAT(K+1))/12.0       RAX 415
DO 80 I=1,K                                         RAX 420
K2=N+1-I                                         RAX 430
80 R(K2)=FLOAT(N)-FLOAT(K-1)/2.0                    RAX 440
C SORT H CARRY ALONG R TO OBTAIN RANKS IN R          RAX 450
DO 90 I=1,K1                                         RAX 460
K2=N-I                                         RAX 470
DO 90 J=1,K2                                         RAX 480
IF (H(J).LE.H(J+1)) GO TO 90                         RAX 490
W=H(J)                                         RAX 500
H(J)=H(J+1)                                         RAX 510
H(J+1)=W                                         RAX 520
W=R(J)                                         RAX 530
R(J)=R(J+1)                                         RAX 540
R(J+1)=W                                         RAX 550
90 CONTINUE                                         RAX 560
RETURN                                         RAX 570
END                                         RAX 580

```

```

C SUBROUTINE RCSUM (A,NROW,N,K,R)          RCS 10
C VERSION 5.00      RCSUM      5/15/70      RCS 20
C WRITTEN BY S PEAVY 11/22/67      RCS 30
C
C WHERE:
C     A IS LOCATION OF MATRIX TO BE SUMMED ROW AND COLUMN WISE RCS 60
C     NROW SIZE OF A IN DIMENSION STATEMENT A(NROW,NROW)      RCS 70
C     N NO OF ROWS IN A                                         RCS 80
C     K NO OF COLS IN A                                         RCS 90
C     R RESULTS.                                              RCS 100
C         R(1).. R(K)  COL SUMS                               RCS 110
C         R(K+1).. R(K+N) ROW SUMS                           RCS 120
C         R(K+N+2)   GRAND SUM                            RCS 130
C         R(K+N+1)   S A(I,J) FOR ALL I,J.                  RCS 140
C         R(K+N+3)   S A(I,J)**2 FOR ALL I,J.                RCS 150
C         R(K+N+4)   SUM OF ABSOLUTE VALUES OF ALL A(I,J)    RCS 160
C
C DIMENSION A(NROW,NROW), R(N)                  RCS 170
C L=1                                         RCS 180
C ASUM=0.                                     RCS 190
C DO 20 J=1,K                                 RCS 200
C SUM=0.                                     RCS 210
C DO 10 I=1,N                                 RCS 220
C ASUM=ASUM+ABS(A(I,J))                     RCS 230
C 10 SUM=SUM+A(I,J)                         RCS 240
C R(L)=SUM                                    RCS 250
C 20 L=L+1                                    RCS 260
C S=0.0                                       RCS 270
C SS=0.0                                      RCS 280
C DO 40 I=1,N                                 RCS 290
C SUM=0.0                                     RCS 300
C DO 30 J=1,K                                 RCS 310
C SUM=SUM+A(I,J)                           RCS 320
C 30 SS=SS+A(I,J)**2                         RCS 330
C S=S+SUM                                    RCS 340
C R(L)=SUM                                    RCS 350
C 40 L=L+1                                    RCS 360
C R(L)=S                                     RCS 370
C R(L+2)=SS                                  RCS 380
C R(L+1)=S                                  RCS 390
C R(L+3)=ASUM                                RCS 400
C RETURN                                     RCS 410
C END                                         RCS 420
C                                         RCS 430

```

```

C      SUBROUTINE READQ                               REQ   10
      VERSION 5.00      READQ      5/15/70          REQ   20
      COMMON /BLOCRC/ NRC,RC(12600)                  REQ   30
      COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NREQ 40
      IARGS,VWXYZ(8),NERROR                         REQ   50
      DIMENSION ARGS(100)                           REQ   60
      EQUIVALENCE (ARGS(1),RC(12501))              REQ   70
      COMMON /QRS/ NDROW,IFLAG,J,NNARG             REQ   80
      COMMON /BLOCKA/ MODE,M,KARD(83),KARG,ARG,ARG2,NEWCD(80),KRDEND  REQ   90
      COMMON /BLOCKC/ KIO,INUNIT,ISCRAT,KBDOUT,KRDKN,LLIST        REQ  100
      COMMON /TAPE/ NAME4(2),NTPCT,IPUNCP,INUNIP,L1TP            REQ  110
      COMMON/KFMT/KFMT(100)                          REQ  115
      IF (IFLAG.NE.0) GO TO 80                      REQ  120
      IF (J.LT.NROW) GO TO 10                      REQ  130
      IFLAG=1                                       REQ  140
      CALL ERROR (201)                            REQ  150
      GO TO 40                                     REQ  160
C      NNARG CONTAINS NARGS OF READ COMMAND        REQ  170
C      KFMT(1)THRU KFMT(NNAGR) CONTAINS ADDRESSES OF COLUMN TOPS  REQ  180
C      THESE CORRECTIONS ARE NEEDED FOR TAPE OPERATIONS    REQ  190
      10     IF (L1TP.NE.45) GO TO 50                REQ  200
      DO 30 I=1,NNARG                                REQ  210
      K=KFMT(I)+J                                    REQ  220
      IF (KIND(I).EQ.0) GO TO 20                    REQ  230
      IF (ARGS(I).NE.0.) GO TO 50                   REQ  240
      GO TO 30                                     REQ  250
      20     IF (IARGS(I).NE.0) GO TO 50                REQ  260
      30     CONTINUE                                 REQ  270
      40     INUNIT=INUNIP                           REQ  280
      MODE=1                                       REQ  290
      GO TO 80                                     REQ  300
      50     IF (NARGS.GE.NNARG) GO TO 55                REQ  305
      NNS=NARGS+1                                  REQ  307
      DO 52 I=NNS,NNARG                            REQ  310
      KIND(I)=0                                    REQ  315
      52     IARGS(I)=0                                REQ  320
      55     DO 70 I=1,NNARG                           REQ  325
      *****                                         ****REQ  330
      C      K=KFMT(I)+J                            REQ  340
      IF (KIND(I).EQ.0) GO TO 60                    REQ  350
      RC(K)=ARGS(I)                                REQ  360
      GO TO 70                                     REQ  370
      60     RC(K)=IARGS(I)                           REQ  380
      70     CONTINUE                                REQ  390
      C      J IS CARD COUNT. IT COUNTS FROM ZERO.    REQ  400
      J=J+1                                       REQ  410
      NRMAX=MAX0(NRMAX,J)                          REQ  420
      C      THESE CORRECTIONS ARE NEEDED FOR TAPE OPERATIONS  REQ  430
      IF (L1TP.NE.46) GO TO 80                    REQ  440
      NTPCT=NTPCT-1                                REQ  450
      IF (NTPCT.EQ.0) GO TO 40                    REQ  460
      *****                                         ****REQ  470
      80     RETURN                                 REQ  480
      END                                     REQ  490

```

```

SUBROUTINE READX                                     REX 10
C VERSION 5.00      READX      5/15/70             REX 20
COMMON /BLOCKA/ MODE,M,KARD(83),KARG,ARG,ARG2,NEWCD(80),KRDEND REX 30
COMMON /BLOCRC/ NRC,RC(12600)                      REX 40
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NREX 50
1ARGS,VWXYZ(8),NERROR                           REX 60
DIMENSION ARGS(100)                            REX 70
EQUIVALENCE (ARGS(1),RC(12501))                REX 80
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG            REX 90
COMMON /QRS/ NDROW,IFLAG,J,NNARG               REX 100
COMMON /BLOCKC/ KIO,INUNIT,ISCRAT,KBDOUT,KRDKNT,LLIST REX 110
C THE FOLLOWING CARD IS NEEDED ONLY FOR TAPE OPERATIONS REX 120
COMMON /TAPE/ NAME4(2),NTPCT,IPUNCP,INUNIP,L1TP       REX 130
C *****
C COMMON/KFMT/KFMT(100)                         REX 140
C IF (L2.NE.1) GO TO 90                          REX 150
C ISRFLG=0                                      REX 170
C IF (NARGS.GT.0) GO TO 20                        REX 180
10 CALL ERROR (10)                                REX 200
GO TO 70                                         REX 210
20 MODE=2                                         REX 220
CALL CHKCOL (I)                                 REX 230
IF(I.EQ.0) GO TO 40                           REX 240
30 CALL ERROR (3)                                REX 250
GO TO 70                                         REX 260
40 IF (NERROR.NE.0) GO TO 70                     REX 270
DO 50 I=1,NARGS                                REX 280
KFMT(I)=IARGS(I)                               REX 290
IARGS(I)=0                                      REX 300
50 ARGS(I)=0.                                    REX 310
IFLAG=0                                         REX 320
J=0      *                                       REX 330
NNARG=NARGS                                     REX 340
GO TO 80                                         REX 350
60 MODE=2                                         REX 360
70 IFLAG=1                                       REX 370
C THE FOLLOWING CARDS ARE NEEDED ONLY FOR TAPE OPERATIONS REX 380
C IF (LITP.EQ.46.OR.LITP.EQ.45) GO TO 200        REX 390
C *****
C 80 RETURN                                       REX 410
C
C          FORMATTED READ                         REX 435
C          READ X N C C C C                      REX 440
C
C          N = NUMBER OF CARDS TO READ. IF N = 0, READ UNTIL A REX 460
C          BLANK CARD IS FOUND                    REX 470
C          X IS THE FORMAT IDENTIFIER, A,B,C,D,E,F REX 480
C
C 90 IF(NARGS.LE.1) GO TO 10                      REX 490
C          SETUP FORMAT                         REX 500
C          CALL PREPAK(4,IND,L2,I,KFMT)           REX 510
C          IF (IND.NE.0) CALL ERROR (27)          REX 520
C          IF (NERROR.NE.0) GO TO 60              REX 525
C          CHECK AND CONVERT ARGUMENTS          REX 530
C          DO 100 I=2,NARGS                      REX 540
C          CALL ADRESS (I,IARGS(I))              REX 550
C          IF(IARGS(I).LE.0) GO TO 185            REX 560
100 CONTINUE                                     REX 565
C          IF(IARGS(1)) 30,110,120              REX 570

```

```

110  N=NRC                                REX 580
     GO TO 130                             REX 590
120  N=IARGS(1)                            REX 600
130  DO 170 I=1,N                           REX 610
     READ (INUNIT,KFMT) (ARGS(J),J=2,NARGS) REX 620
C       CHECK IF LOOKING FOR BLANK CARD    REX 630
     IF (IARGS(1).NE.0) GO TO 150          REX 640
     DO 140 J=2,NARGS                      REX 650
     IF (ARGS(J).NE.0.) GO TO 150          REX 655
140  CONTINUE                               REX 657
C       BLANK CARD FOUND, TERMINATE READ.   REX 660
     GO TO 180                             REX 670
C       IF THERE IS TOO MUCH DATA, DO NOT ENTER EXCESS REX 680
150  IF (I.GT.NROW) GO TO 170              REX 690
     DO 160 J=2,NARGS                      REX 700
     K=IARGS(J)                            REX 710
     IARGS(J)=K+1                          REX 720
160  RC(K)=ARGS(J)                        REX 730
170  CONTINUE                               REX 740
     I=N+1                                 REX 750
180  I=I-1                                 REX 760
     NRMAX=MAX0(NRMAX,MIN0(I,NROW))      REX 770
     WRITE (ISCRAT,210) I                  REX 780
     IF (I.GT.NROW) CALL ERROR (201)        REX 790
C       THE FOLLOWING CARDS ARE NEEDED ONLY FOR TAPE OPERATIONS REX 800
C       THE CARD WAS GO TO 80             REX 810
     GO TO 190                            REX 820
C       ***** REX 830
185  CALL ERROR (11)                         REX 840
C       THE FOLLOWING CARD IS NEEDED ONLY FOR TAPE OPERATIONS REX 850
C       THE CARD WAS GO TO 80             REX 860
190  IF(L1TP.NE.45.AND.L1TP.NE.46) RETURN  REX 870
200  INUNIT=INUNIP                          REX 880
     MODE=1                               REX 890
     RETURN                               REX 900
C       ***** REX 910
C
210  FORMAT (5X,I4,33H DATA CARD(S) READ BUT NOT LISTED, 42X) REX 930
     END                                  REX 940

```

```

SUBROUTINE REPINC (IJSWT)
C      VERSION 5.00      REPINC      5/15/70          REP   10
C      WRITTEN BY R VARNER    4/ 9/68          REP   20
C      COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG          REP   30
C      COMMON /BLOCKB/ NSTMT,NSTMTH,NCOM,LCOM,IOVFL,COM(2000)  REP   40
C      COMMON /BLOCKX/ INDEX(6,8),LEVEL          REP   50
C      COMMON /BLOCRC/ NRC,RC(12600)          REP   60
C      COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NREP  REP   70
C      IARGS,VWXYZ(8),NERROR          REP   80
C      DIMENSION ARGS(100)          REP   90
C      EQUIVALENCE (ARGS(1),RC(12501))          REP  100
C
C      IJSWT=1 COMMAND IS REPEAT  INITIALIZE THINGS          REP  110
C      IJSWT=2 IN REPEAT MODE          REP  120
C      IJSWT=3 COMMAND IS INCREMENT OR RESTORE          REP  130
C
C      L2=6 INCREMENT          REP  140
C      L2=8 RESTORE          REP  150
C
C      GO TO (350,490,10), IJSWT          REP  160
10     IF (L2.EQ.6) GO TO 20          REP  170
      T=0.0          REP  180
      GO TO 30          REP  190
20     T=1.0          REP  200
30     IF (NARGS.GE.2) GO TO 50          REP  210
40     K=10          REP  220
      GO TO 320          REP  230
C
C      GET STATEMENT NUMBER. CAN BE FLOATING OR INTEGER.          REP  240
C
50     IF (KIND(1).EQ.0) GO TO 60          REP  250
      J=10.*ARGS(1)+.5          REP  260
      GO TO 70          REP  270
60     J=10*IARGS(1)          REP  280
70     IF (J.GT.NSTMTH) GO TO 80          REP  290
      J=LOCATE(J)          REP  300
C
C      J HAS LOCATION OF COMMAND TO BE MODIFIED          REP  310
C
80     IF (J.GT.0) GO TO 90          REP  320
      K=13          REP  330
      GO TO 320          REP  340
C
C      JJ IS FIRST LOCATION OF THE NEXT STORED COMMAND.          REP  350
C
90     JJ=J+IFIX(COM(J+1))          REP  360
C
C      CHECK THAT COMMAND HAS THE PROPER NUMBER OF ARGUMENTS          REP  370
C
      IF (NARGS-1.NE.MOD(IFIX(COM(J+2)),64)) GO TO 40          REP  380
      J=J+3          REP  390
C
C      SKIP OVER HEADER          REP  400
C
C
C      CHECK IF THIS COMMAND IS STORED. IF SO, PULL OUT INTO ARGTAB.          REP  410
C      (ALL BUT FIRST ARG WHICH IS STATEMENT NUMBER)          REP  420
C
      IF (LEVEL.EQ.0) GO TO 110          REP  430

```

```

K=2*NARGS          REP 600
DO 100 I=2,K       REP 610
ARGTAB(I)=COM(I2+4) REP 620
100   I2=I2+1      REP 630
C                                         REP 640
C   I2 IS LOCATION OF THIS COMMAND        REP 650
C                                         REP 660
110   I=2+KIND(1)      REP 670
C                                         REP 680
C   PERFORM INCREMENT OR RESTORE.  PICK UP ARGUMENT FROM    REP 690
C   COMMAND TO BE MODIFIED AND EXAMINE IT.                  REP 700
C                                         REP 710
120   IF (COM(J)) 250,130,200      REP 720
C                                         REP 730
C   FLOATING POINT CONST.                REP 740
C                                         REP 750
130   IF (ARGTAB(I)) 140,160,310      REP 760
C                                         REP 770
C   INCR. FL. PT. CONST. BY 'STATEMENT'      REP 780
140   IF(ARGTAB(I).EQ.(-1.)) GO TO 310      REP 790
     CALL XPND (ARGTAB(I),K,Y,KND)
     IF (K.LT.0) GO TO 220
     IF (KND.EQ.0) GO TO 310
     COM(J+1)=T*COM(J+1)+Y
     J=J+2
150   I=I+K+1      REP 850
     GO TO 190      REP 860
160   COM(J+1)=T*COM(J+1)+ARGTAB(I+1)      REP 870
170   J=J+2      REP 880
180   I=I+2      REP 890
190   IF (J-JJ) 120,330,330      REP 900
C                                         REP 910
C   COLUMN NUMBER                    REP 920
C                                         REP 930
200   IF (ARGTAB(I)) 210,310,230      REP 940
C                                         REP 950
C   INTEGER CONST MODIFIED BY :STATEMENT:      REP 960
C                                         REP 970
210   IF(ARGTAB(I).EQ.(-1.)) GO TO 310      REP 980
     CALL XPND (ARGTAB(I),K,Y,KND)
     IF (K.GE.0) IF (KND) 310,225,310      REP 990
220   K=-K      REP1000
     GO TO 320      REP1010
225   COM(J)=T*COM(J)+Y      REP1020
     J=J+1      REP1030
     GO TO 150      REP1040
230   COM(J)=T*(COM(J)-8192.)+ARGTAB(I)      REP1050
     IF (COM(J)) 300,300,240      REP1060
240   J=J+1      REP1070
     I=I+1      REP1080
     GO TO 190      REP1090
C                                         REP1100
C   VARIABLE *REFERENCE*                REP1110
C   NRMAX,V,W,X,Y,Z CAN ONLY BE INCREMENTED, BY 0 OR 0.      REP1120
C   WHETHER 0 OR 0. INCREMENTS :X: OR 'X' IS IMMATERIAL.      REP1130
C                                         REP1140
250   IF(COM(J).LT.(-16.)) GO TO 260      REP1150
     IF(COM(J).EQ.(-1.)) GO TO 340      REP1160
     J=J+1      REP1170

```

```

GO TO 180                               REP1180
C                                         REP1190
C                                         *ROW,COL* REFERENCE.      REP1200
C                                         REP1210
260  IF (ARGTAB(I)+16.0) 270,310,310      REP1220
270  COM(J)=T*(COM(J)+8208.)+ARGTAB(I)    REP1230
     IF(COM(J).GT.(-16.)) GO TO 310       REP1240
     IF (COM(J+1)*ARGTAB(I+1)) 310,310,280  REP1250
280  Y=T*(ABS(COM(J+1))-8192.)+ABS(ARGTAB(I+1))  REP1260
     IF (Y) 310,310,290                   REP1270
290  COM(J+1)=SIGN(Y,COM(J+1))           REP1280
     GO TO 170                           REP1290
300  K=18                                REP1300
     GO TO 320                           REP1310
310  K=20                                REP1320
320  CALL ERROR (K)                      REP1330
330  RETURN                               REP1340
C                                         REP1350
C                                         *** (=THRU) IGNORE. INCREM. OR RESTORE MAY OR MAY NOT
C                                         HAVE CORRESPONDING ***      REP1360
C                                         REP1370
C                                         REP1380
340  IF(ARGTAB(I).EQ.(-1.))   I=I+1      REP1390
     J=J+1                               REP1400
     GO TO 190                           REP1410
C                                         REP1420
C                                         REP1430
C                                         NESTED PERFORMS UP TO EIGHT LEVELS ARE ALLOWED.      REP1440
C                                         CURRENT LEVEL IS STORED IN LEVEL          REP1450
C                                         REP1460
C                                         INDEX(1,LEVEL) CONTAINS LOCATION OF COMMAND AT ARG1 (FIRST)      REP1470
C                                         INDEX(2,LEVEL) CONTAINS RUNNING INDEX FROM ARG1 TO ARG2      REP1480
C                                         INDEX(3,LEVEL) CONTAINS LOCATION OF COMMAND AT ARG2(LAST)      REP1490
C                                         INDEX(4,LEVEL) CONTAINS THIRD ARG (REPEAT COUNT)      REP1500
C                                         INDEX(5,LEVEL) CONTAINS CURRENT LEVEL COUNTER (1 TO ARG3)      REP1510
C                                         INDEX(6,LEVEL) CONTAINS STATEMENT NUMBER OF STATEMENT CURRENTLY      REP1520
C                                         BEING EXECUTED.          REP1530
C                                         REP1540
350  IF (NARGS-3) 360,400,390             REP1550
360  IF (NARGS-1) 390,370,380             REP1560
C                                         REP1570
C                                         SECOND ARG MISSING,MAKE SAME AS FIRST ARG      REP1580
C                                         REP1590
370  IARGS(2)=IARGS(1)                  REP1600
     KIND(2)=KIND(1)                    REP1610
C                                         REP1620
C                                         THIRD ARG MISSING, SET TO INTEGER 1      REP1630
C                                         REP1640
380  IARGS(3)=1                         REP1650
     KIND(3)=0                         REP1660
     GO TO 410                         REP1670
390  CALL ERROR (10)                   REP1680
     GO TO 500                         REP1690
400  IF (KIND(3).EQ.0.AND.IARGS(3).GT.0) GO TO 410      REP1700
     CALL ERROR (3)                    REP1710
     GO TO 500                         REP1720
410  DO 450 I=1,2                     REP1730
     IF (KIND(I).EQ.0) GO TO 420      REP1740
     IARGS(I)=10.*ARGS(I)+.5        REP1750
     GO TO 430                         REP1760

```

```

420 IARGS(I)=10*IARGS(I) REP1770
430 IF (IARGS(I).GT.NSTMTH) GO TO 440 REP1780
IARGS(I)=LOCATE(IARGS(I)) REP1790
IF (IARGS(I).GT.0) GO TO 450 REP1800
440 CALL ERROR (13) REP1810
GO TO 500 REP1820
450 CONTINUE REP1830
IF (LEVEL.LT.8) GO TO 460 REP1840
CALL ERROR (19) REP1850
GO TO 500 REP1860
460 IF (IARGS(2).LT.IARGS(1)) CALL ERROR (3) REP1870
IF (NERROR.NE.0) GO TO 500 REP1880
LEVEL=LEVEL+1 REP1890
INDEX(1,LEVEL)=IARGS(1) REP1900
INDEX(3,LEVEL)=IARGS(2) REP1910
INDEX(4,LEVEL)=IARGS(3) REP1920
INDEX(5,LEVEL)=0 REP1930
C REP1940
C OUTER LOOP REP1950
C REP1960
470 INDEX(5,LEVEL)=INDEX(5,LEVEL)+1 REP1970
IF (INDEX(5,LEVEL).LE.INDEX(4,LEVEL)) GO TO 480 REP1980
C REP1990
C FINISHED OUTER LOOP, REDUCE LEVEL BY 1 REP2000
C REP2010
LEVEL=LEVEL-1 REP2020
IF (LEVEL.GT.0) GO TO 490 REP2030
RETURN REP2040
480 INDEX(2,LEVEL)=INDEX(1,LEVEL) REP2050
490 I2=INDEX(2,LEVEL) REP2060
IF (I2.GT.INDEX(3,LEVEL)) GO TO 470 REP2070
INDEX(6,LEVEL)=COM(I2) REP2080
K=COM(I2+1) REP2090
INDEX(2,LEVEL)=INDEX(2,LEVEL)+K REP2100
L2=COM(I2+2) REP2110
L1=L2/64 REP2120
NARGS=L2-64*L1 REP2130
L2=L1/64 REP2140
L1=L1-64*L2 REP2150
CALL EXPAND (K-2,COM(I2+3)) REP2160
RETURN REP2170
500 IJSWT=-IJSWT REP2180
RETURN REP2190
END REP2200

```

```

SUBROUTINE RESET                               RES   10
C      VERSION 5.00     RESET      5/15/70      RES   20
COMMON /BLOCRC/ NRC,RC(12600)                 RES   30
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NRES 40
IARGS,VWXYZ(8),NERROR                         RES   50
DIMENSION ARGS(100)                           RES   60
EQUIVALENCE (ARGS(1),RC(12501))              RES   70
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG          RES   80
IF(NARGS.EQ.1) IF(L2-2) 25,20,40             RES   85
K=10                                         RES   90
10    CALL ERROR (K)                          RES  100
20    RETURN                                     RES  110
C      RESET NRMAX                           RES  120
25    IF (KIND(1).NE.0) IARGS(1)=ARGS(1)       RES  130
      IF (IARGS(1).GE.0.AND.IARGS(1).LE.NROW) GO TO 30
      K=3                                         RES  150
      GO TO 10                                    RES  160
30    NRMAX=IARGS(1)                          RES  170
      GO TO 20                                    RES  180
C      RESET V,W,X,Y,Z                         RES  190
40    IF (KIND(1).EQ.0) ARGS(1)=IARGS(1)       RES  200
      VWXYZ(L2-2)=ARGS(1)                      RES  210
      GO TO 20                                    RES  220
      END                                         RES  230

```

```

C SUBROUTINE RFORMAT (X,N,NS,NW,ND,NX,XVAL,ARRAY,NB,NC) RFO 10
C VERSION 5.00 RFORMAT 5/15/70 RFO 20
C WRITTEN BY DAVID HOGBEN, SEL, NBS. 4/18/69. RFO 30
C *****
C REWRITE AND COMBINING OF FXFARG(2/1/69) AND FXFORM(2/7/69). RFO 40
C FLOATING FORMAT 1PE NW.NS-1 IS GIVEN IF N LT 0 AND NX LT 0. RFO 50
C FIXED FORMAT F NW.ND IS GIVEN IF N GE 0 AND NX LT 0. SET NS=8. RFO 60
C PERIOD NOT PRINTED IF NDECS=0 RFO 70
C FXFARG SET *** XVAL=X (OR ANY REAL VARIABLE) RFO 80
C ARRAY=A (OR ANY DIMENSIONED VECTOR) RFO 90
C NBLANK=0 (OR ANY OTHER INTEGER) RFO 100
C INPUT *** X, N, NSIGDS, NWMAX. RFO 110
C OUTPUT *** NWIDTH, NDECS. RFO 120
C FXFORM SET *** X=X(OR ANY REAL VARIABLE) RFO 130
C N=1 (OR ANY INTEGER) RFO 140
C NWMAX=0 RFO 150
C INPUT *** XVAL, NSIGDS, NWIDTH, NDECS, NBLANK. RFO 160
C OUTPUT *** ARRAY RFO 170
C *****
C NUMBER IS CENTERED IF NC=1 AND NOT CENTERED IF NC=0(RIGHT JUST'D) RFO 180
C THIS REVISION WRITTEN 10/09/69 RFO 190
C COMMON /ABCDEF/ L(48) RFO 195
C DIMENSION X(1), ARRAY(1), C(10) RFO 200
C EQUIVALENCE (C(1),L(1)), (BLANK,L(45)), (PERIOD,L(38)), (CPLUS,L(4RFO 210
10 10)), (CMINUS,L(39)), (CASTER,L(41)) RFO 220
C DOUBLE PRECISION Z RFO 230
C NSIGDS=MIN0(8,NS) RFO 240
C NSIGDS=MAX0(1,NSIGDS) RFO 250
C IF (NX) 50,60,20 RFO 260
C *****
C RFO 270
C FXFARG SUBROUTINE RFO 280
C INPUT FOR THE SUBROUTINE FXFORM. NWMAX IS THE MAXIMUM ALLOWABLE RFO 290
C VALUE OF NWIDTH. X MUST BE DIMENSIONED AND N IS ITS LENGTH. RFO 300
C NWMAX MUST BE GREATER THAN NSIGDS PLUS FOUR RFO 310
C NWIDTH = MIN(MMAX-MMIN+NSIGDS+2,NWMAX) RFO 320
C NDECS=NSIGDS-MMIN-1, NDECS=MIN0(MAX0(NDECS,NSIGDS+2),NWMAX-3), IF RFO 330
C NWIDTH EXCEEDS NWMAX. RFO 340
C IF NSIGDS GT 8, IT IS SET = TO 8, IF LT 1 SET = TO 1. RFO 350
C REFERENCE *** SEL NOTE N-68-3, SEPTEMBER, 1968. RFO 360
C WRITTEN BY DAVID HOGBEN, SEL, NBS. 2/ 1/69. RFO 370
C *****
C RFO 380
C 20 NWMAX=MAX0(NSIGDS+5,NX) RFO 390
C Y=ABS(X(1)) RFO 400
C IF (Y.LE.0.) Y=1.0 RFO 410
C Y1=Y RFO 420
C Y2=Y RFO 430
C IF (N.LT.1) N=1 RFO 440
C DO 30 I=1,N RFO 450
C Y=ABS(X(I)) RFO 460
C IF (Y.LE.0.) Y=1.0 RFO 470
C IF (Y.LT.Y1) Y1=Y RFO 480
C IF (Y.GT.Y2) Y2=Y RFO 490
C CONTINUE RFO 495
C MMIN=FLOG10(Y1) RFO 500
C IF (Y1.LT.1.) MMIN=MMIN-1 RFO 510
C MMAX=FLOG10(Y2) RFO 520
C IF (Y2.LT.1.) MMAX=MMAX-1 RFO 530
C Z=Y1 RFO 540
C LL=Z*10.D0** (NSIGDS-MMIN)+5.0D0 RFO 550

```

```

IF (LL.GE.10**NSIGDS+1) MMIN=MMIN+1 RFO 560
IF (LL.LT.10**NSIGDS) MMIN=MMIN-1 RFO 570
Z=Y2 RFO 580
LL=Z*10.D0**NSIGDS-MMAX)+5.0D0 RFO 590
IF (LL.GE.10**NSIGDS+1) MMAX=MMAX+1 RFO 600
IF (LL.LT.10**NSIGDS) MMAX=MMAX-1 RFO 610
NDECS=NSIGDS-MMIN-1 RFO 620
NDECS=MAX0(0,NDECS) RFO 630
NWIDTH=MMAX+3+NDECS RFO 640
IF (MMAX.LT.0) NWIDTH=NDECS+2 RFO 650
IF (NWIDTH.LE.NWMAX) GO TO 40 RFO 660
NDECS=MAX0(NDECS,NSIGDS+2) RFO 670
NDECS=MIN0(NDECS,NWMAX-3) RFO 680
NWIDTH=NWMAX RFO 690
40 IF (NDECS.LT.0) NDECS=0 RFO 700
ND=NDECS RFO 710
NW=NWIDTH RFO 720
RETURN RFO 730
C **** RFO 740
C FXFORM SUBROUTINE RFO 750
C SUBROUTINE FXFORM ALLOWS PRINTING OF REAL NUMBERS X IN FIXED RFO 760
C FORMAT WITH DECIMAL POINT IN CONSTANT POSITION. NWIDTH = WIDTH OF RFO 770
C FIELD, NDECS = NUMBER OF PLACES TO RIGHT OF DECIMAL POINT, X IS RFO 780
C THE NUMBER, ARRAY IS THE VECTOR WHERE X IS RETURNED TO BE PRINTED RFO 790
C ACCORDING TO A FORMAT, NSIGDS = NUMBER OF DIGITS GIVEN WITH BLANKS RFO 800
C ON THE RIGHT, NBLANK IS THE NUMBER OF BLANKS TO BE PUT ON THE RFO 810
C LEFT OF THE FIELD. IF X IS TOO LARGE OR TOO SMALL IT IS RETURNED RFO 820
C AS A FLOATING POINT NUMBER. ARRAY MUST BE DIMENSIONED. RFO 830
C A PERIOD IS NOT GIVEN IF X IS GREATER THAN OR EQUAL TO 10**NSIGDS RFO 840
C ZERO IS WRITTEN 0. RFO 850
C IF NDECS=NWIDTH, BLANKS ARE RETURNED RFO 860
C SUBROUTINE MAY BE USED IN CONJUNCTION WITH SUBROUTINE FXFARGS. RFO 870
C IF NSIGDS GT 8, IT IS SET = TO 8, IF LT 1 SET = TO 1. RFO 880
C NWIDTH IS ADJUSTED IF NECESSARY SO THAT IT IS GE NDECS+2, RFO 890
C GE NSIGDS+2 IF FIXED AND GE NSIGDS+5 IF FLOATING RFO 900
C REFERENCE *** SEL NOTE N-68-3, SEPTEMBER, 1968. RFO 910
C WRITTEN BY DAVID HOBGEN, SEL, NBS. 2/7/69. RFO 920
C **** RFO 930
50 NWIDTH=NW RFO 940
GO TO 70 RFO 950
60 NWIDTH=MAX0(NW,NSIGDS+2) RFO 960
70 NDECS=MAX0(0,ND) RFO 970
IF (NWIDTH.LT.NDECS) NWIDTH=NDECS+2 RFO 980
NB=MAX0(0,NB) RFO 985
IF (NC.NE.0) NC=1 RFO 990
NBLANK=NB-(NB/2)*NC RFO 995
MF=0 RFO 1000
Y=ABS(XVAL) RFO 1010
80 NDIFF=NWIDTH-NDECS RFO 1020
NWMAX=NWIDTH+NBLANK RFO 1030
NPONE=NDIFF+NBLANK RFO 1040
IEND=NWIDTH+NB RFO 1045
DO 90 I=1,IEND RFO 1050
90 ARRAY(I)=BLANK RFO 1060
IF (NDECS.EQ.NWIDTH) RETURN RFO 1070
IF (Y.GT.0.) GO TO 110 RFO 1080
C XVAL=0. IS SPECIAL CASE UNLESS FIXED OR FLOATING RFO 1090
IF (NX.GE.0) GO TO 100 RFO 1100
C FIXED RFO 1110

```

```

M=0                               RF01120
GO TO 130                         RF01130
100 IF (NDECS.NE.0) ARRAY(NPONE)=PERIOD
                                RF01140
ARRAY(NPONE-1)=C(1)                RF01150
RETURN                            RF01160
110 M=FLOG10(Y)                   RF01170
IF (Y.LT.1.) M=M-1                 RF01180
Z=Y                               RF01190
Z = Z*10.D0** (NSIGDS-M)          RF01193
X1 = Z                            RF01196
LL1 = X1                           RF01200
X2 = Z - DBLE(X1)                 RF01205
LL2 = X2                           RF01210
LL = LL1 + LL2 + 5                RF01215
IF (LL.LT.10** (NSIGDS+1)) GO TO 120
M=M+1                            RF01220
LL=LL/10                          RF01230
GO TO 130                         RF01240
RF01250
120 IF (LL.GE.10**NSIGDS) GO TO 130
M=M-1                            RF01260
LL=10*LL                          RF01270
RF01280
130 IF (NX.EQ.0) GO TO 170
IF (N.LT.0) GO TO 180
C FIXED                           RF01290
RF01300
IF (M.LT.NDIFF-2) GO TO 150
IF (M.EQ.NDIFF-2.AND.XVAL.GE.0.) GO TO 150
NSIGDS=MAX0(0,NWIDTH-5)           RF01340
IF (NSIGDS.GT.0) GO TO 180
RF01350
C PUT IN ASTERISKS               RF01360
DO 140 I=1,NWIDTH
IRVSP=I+NBLANK                    RF01370
RF01375
140 ARRAY(IRVSP)=CASTER          RF01380
RETURN                            RF01390
RF01400
150 NSIGDS=MIN0(8,NDECS+M+1)
NSIGDS=MAX0(0,NSIGDS)
LL=(LL-5)/(10** (8-NSIGDS))+5    RF01410
IF (NSIGDS.GT.0) GO TO 170
RF01420
IF (XVAL.LT.0.0) ARRAY(NPONE-1)=CMINUS
DO 160 I=NPONE,NWMAX             RF01430
RF01440
RF01450
160 ARRAY(I)=C(1)
ARRAY(NPONE)=PERIOD              RF01460
RF01470
IF (NDECS+1.EQ.(-M).AND.LL.GT.10) ARRAY(NWMAX)=C(2)
RETURN                            RF01480
RF01490
RF01500
170 MREAL=0
IF (M.GE.NSIGDS-1-NDECS.AND.M.LT.NDIFF-2) GO TO 190
IF (M.EQ.NDIFF-2.AND.XVAL.GT.0.) GO TO 190
C FLOATING                         RF01510
RF01520
RF01530
180 MREAL=M
M=0                               RF01540
RF01550
MF=1                               RF01560
RF01570
Y=Y*10.**(-MREAL)
190 IF (M.LT.NSIGDS.AND.NDECS.NE.0) ARRAY(NPONE)=PERIOD
NINT=NPONE-1-M
IF (M.LT.0) NINT=NINT+1
NEND=NINT+NSIGDS-1
IF (M.GE.0.AND.M.LT.NSIGDS-1) NEND=NEND+1
DO 200 J=NINT,NEND
I=NEND+NINT-J
IF (I.EQ.NPONE) GO TO 200
RF01580
RF01590
RF01600
RF01610
RF01620
RF01630
RF01640
RF01650

```

LL=LL/10	RF01660
NN=MOD(LL,10)	RF01670
ARRAY(I)=C(NN+1)	RF01680
200 CONTINUE	RF01690
IF (MF.EQ.0) GO TO 220	RF01700
C PUT IN EXPONENT FOR FLOATING POINT NUMBER	RF01710
IF (NWIDTH.GE.NSIGDS+5) GO TO 210	RF01720
NWIDTH=NSIGDS+5	RF01730
GO TO 80	RF01740
210 IF (MREAL.LT.0) ARRAY(NEND+1)=CMINUS	RF01750
IF (MREAL.GE.0) ARRAY(NEND+1)=CPLUS	RF01760
MREALA=IABS(MREAL)	RF01770
M1=MREALA/10	RF01780
M2=MOD(MREALA,10)	RF01790
ARRAY(NEND+2)=C(M1+1)	RF01800
ARRAY(NEND+3)=C(M2+1)	RF01810
220 IF (XVAL.LT.0..AND.M.GE.0) ARRAY(NINT-1)=CMINUS	RF01820
IF (XVAL.LT.0..AND.M.LT.0) ARRAY(NPONE-1)=CMINUS	RF01830
IF (M.GE.(-1)) GO TO 240	RF01840
I1=NPONE+1	RF01850
I2=NINT-1	RF01860
DO 230 I=I1,I2	RF01870
230 ARRAY(I)=C(1)	RF01880
RETURN	RF01890
C PUT IN NON-SIGNIFICANT ZEROS	RF01900
240 IF (M.LT.NSIGDS.OR.MF.NE.0) RETURN	RF01910
I1=NINT+NSIGDS	RF01920
I2=NPONE-1	RF01930
DO 250 I=I1,I2	RF01940
250 ARRAY(I)=C(1)	RF01950
RETURN	RF01960
END	RF01970

```

SUBROUTINE RNDOWN          RND  10
C VERSION 5.00      RNDOWN      5/15/70      RND  20
COMMON /BLOCKC/ KIO,INUNIT,ISCRAT,KBDOUT,KRDKNT,LLIST      RND  30
COMMON /BLOCKX/ INDEX(6,8),LEVEL      RND  40
C
C IF AN ERROR IS MADE IN A STORED STATEMENT, THIS ROUTINE PRINTS      RND  50
C OUT EXACTLY WHEN AND WHERE IT OCCURRED.      RND  60
C
C A=FLOAT(INDEX(6,LEVEL))/10.      RND  90
WRITE (ISCRAT,50) A      RND 100
N=LEVEL-1      RND 110
10 IF (N) 40,30,20      RND 120
20 A=FLOAT(INDEX(6,N))/10.      RND 130
WRITE (ISCRAT,60) INDEX(5,N+1),INDEX(4,N+1),A      RND 140
N=N-1      RND 150
GO TO 10      RND 160
30 WRITE (ISCRAT,70) INDEX(5,1),INDEX(4,1)      RND 170
40 RETURN      RND 180
C
50 FORMAT (31H IN COMMAND AT STATEMENT NUMBER,F6.1,47X)      RND 190
60 FORMAT (10H CYCLE NO.,I4,3H OF,I4,24H OF PERFORM AT STATEMENT,F6.1RNDRND 210
1,23X)      RND 220
70 FORMAT (10H CYCLE NO.,I4,3H OF,I4,31H OF EXTERNAL PERFORM STATEMENRND 230
1T.,32X)      RND 240
END      RND 250

```

```

SUBROUTINE RNJBK (RNO,NSTART,NFINSH)      RNJ  10
C VERSION 5.00      RNJBK      5/15/70      RNJ  20
DATA M,FLM/8192,8192./      RNJ  30
C 8192=2**13      RNJ  40
C JB KRUSKAL (1969) ACM, 12, 93-94.      RNJ  50
C ****      RNJ  60
C RETURNS RANDOM NUMBER (0,1) IN RNJBK.      RNJ  70
C SET NSTART=1 TO START AT BEGINNING, OTHERWISE PREVIOUS NFINSH.      RNJ  80
C RETURNS NFINSH FOR LAST NUMBER GENERATED.      RNJ  90
C WRITTEN BY DAVID HOGGEN, SEL, NBS. 3/24/69.      RNJ 100
C ****      RNJ 110
K=NSTART      RNJ 120
DO 10 I=1,3      RNJ 130
10 K=MOD(5*K,M)      RNJ 140
RNO=FLOAT(K)/FLM      RNJ 150
NFINSH=K      RNJ 160
RETURN      RNJ 170
END      RNJ 180

```

```

SUBROUTINE RPRINT                                     RPR 10
C VERSION 5.00      RPRINT      5/15/70           RPR 20
C **** RPRINT COLS ++, ++, ... ++ (MAXIMUM OF 50 COLS) (NO OF SD IS 8) RPR 40
C RPRINT COLS ++, ++, ... ++ WITH ** SIGNIFICANT DIGITS (49 COL MAX)RPR 50
C RPRINT COLS ++ .. ++ WITH ** SD, ... (NWMAX=13, NWIDTH+NBLANK=15)RPR 60
C                                         RPR 70
C RPRINT ** COLS, COL++ WITH , SD ... (NWMAX=22, NBLANK=3)          RPR 80
C RPRINT ** COLS, ++ WITH , SD AND NWMAX = , , ... (NBLANK=3)        RPR 90
C RPRINT ** COLS, ++ WITH , SD NWMAX , NBLANK , , ...                 RPR 100
C ****                                         RPR 110
C FLOATING 1PEW.D IS OBTAINED IN LAST TWO OPTIONS IF NWMAX=0.          RPR 120
C     W=NSIGDS+5 AND D=NSIGDS-1                                         RPR 130
C FIXED FW.D IS OBTAINED IN LAST TWO OPTIONS IF NWMAX LT 0.            RPR 140
C     W=-NWMAX AND D=NSIGDS                                         RPR 150
C IW IS OBTAINED IN LAST TWO OPTIONS IF NSIGDS=0 AND NWMAX=-W.          RPR 160
C ****                                         RPR 170
C REPLACES TRAILING ZEROS BY BLANKS IF COUNT LT NRMAX                  RPR 180
C IF COLUMN CONTAINS ALL ZEROS AND FIRST ARGUMENT IS A COLUMN NUMBERRPR 190
C NO COLUMN HEADING IS GIVEN                                         RPR 200
C IF FIRST ARGUMENT IS NOT A COLUMN NUMBER (1) NO HEADING IF WIDTH RPR 210
C LESS THAN 6, (2) COL NO IF 6 LE WIDTH LT 12, (0) COLUMN XXX. IFRPR 220
C WIDTH GREATER THAN OR EQUAL TO 12.                                     RPR 230
C IF NUMBER IS FLOATED ASTERISK IS PUT TO RIGHT OF FIELD.             RPR 240
C NUMBERS ARE PRINTED IN BLOCKS OF 5 IF NRMAX IS LESS THAN 49.          RPR 250
C WRITTEN BY DAVID HOGBEN, SEL, NBS.   4/17/69.                         RPR 260
C ****                                         RPR 270
C COMMON /ABCDEF/ L(48)                                              RPR 280
C COMMON /BLOCRC/ NRC,RC(12600)                                         RPR 290
C COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NRPR 300
C IARGS,VWXYZ(8),NERROR                                         RPR 310
C DIMENSION ARGS(100)                                              RPR 320
C EQUIVALENCE (ARGS(1),RC(12501))                                     RPR 330
C COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG                                RPR 340
C COMMON /SCRAT/ NS,NS2,A(13500)                                         RPR 350
C COMMON /HEADER/ NOCARD(80),ITLE(60,6),LNCNT,IPRINT,NPAGE,IPUNCH       RPR 360
C COMMON /FMAT/ IFMTX(6),IOSWT,IFMTS(6),LHEAD(96)                      RPR 370
C ****                                         RPR 380
C DIMENSION NWIDTH(100), NDECS(100), NBLANK(100), IRGS(100), NCOUNT(RPR 390
C 1100), NWMAX(100), NSIGDS(100), AL(48), NF(100), NWM(100)             RPR 400
C EQUIVALENCE (NWIDTH(1),A(1001)), (NDECS(1),A(1101)), (NBLANK(1),A(RPR 410
C 11201)), (IRGS(1),A(1301)), (NCOUNT(1),A(1401)), (NWMAX(1),A(1601))RPR 420
C 2, (NSIGDS(1),A(1701)), (AL(1),L(1)), (NF(1),A(1801)), (NWM(1),A(19RPR 430
C 301))                                         RPR 440
C ****                                         RPR 450
C IRGS(I) NEEDED FOR HEADS BECAUSE CHKCOLUMN IS USED                   RPR 460
C LINES 20 TO 70 DO ERROR CHECKING                                     RPR 470
C IF (L1.NE.6) GO TO 20                                         RPR 480
C NARGS=NARGS-1                                         RPR 490
C IF (KIND(1).EQ.1) GO TO 50                                         RPR 500
C IF (IARGS(1).LE.0.OR.IARGS(1).GT.NROW) GO TO 60                     RPR 510
C NRJ=IARGS(1)-1                                         RPR 520
C IF (NARGS.LT.1) GO TO 30                                         RPR 530
C DO 10 I=1,NARGS                                         RPR 540
C KIND(I)=KIND(I+1)                                         RPR 550
C IARGS(I)=IARGS(I+1)                                         RPR 560
10  ARGS(I)=ARGS(I+1)                                         RPR 570
20  IF (NARGS.GT.0) GO TO 40                                         RPR 580
30  CALL ERROR (10)                                         RPR 590

```

```

GO TO 400                                RPR 600
40  IF (NRMAX.GT.0) GO TO 70              RPR 610
CALL ERROR (9)                            RPR 620
GO TO 400                                RPR 630
50  CALL ERROR (3)                          RPR 640
GO TO 400                                RPR 650
60  CALL ERROR (11)                         RPR 660
GO TO 400                                RPR 670
C   ALL ARGUMENTS ARE INTEGERS.           RPR 680
70  NPAR=1                                 RPR 690
IF (KIND(1).EQ.1.OR.KIND(NARGS).EQ.1) GO TO 90 RPR 700
DO 80 I=1,NARGS                           RPR 710
NSIGDS(I)=8                               RPR 720
NF(I)=1                                   RPR 730
NWM(I)=0                                  RPR 740
NWMAX(I)=13                             RPR 750
80  IRGS(I)=IARGS(I)                      RPR 760
CALL CHKCOLUMN (J)                        RPR 770
IF (J.NE.0) GO TO 50                      RPR 780
NA=NARGS                                 RPR 790
GO TO 160                                RPR 800
C   LAST ARGUMENT IS NOT AN INTEGER, NSIGDS IS GIVEN. RPR 810
90  IF (KIND(1).EQ.1) GO TO 130            RPR 820
LL=0                                     RPR 830
DO 100 I=1,NARGS                          RPR 840
IF (KIND(I).EQ.0) GO TO 100               RPR 850
LL=LL+1                                 RPR 860
ARGS(LL)=ARGS(I)                         RPR 870
100 CONTINUE                               RPR 880
NL=0                                     RPR 890
DO 120 I=1,NARGS                          RPR 900
IF (KIND(I).EQ.0) GO TO 110               RPR 910
NL=NL+1                                 RPR 920
GO TO 120                                RPR 930
110 I2=I-NL                               RPR 940
IARGS(I2)=IARGS(I)                        RPR 950
NSIGDS(I2)=ARGS(NL+1)                     RPR 960
NWMAX(I2)=13                            RPR 970
NF(I2)=1                                 RPR 980
NWM(I2)=0                               RPR 990
IRGS(I2)=IARGS(I2)                        RPR1000
120  KIND(I2)=0                            RPR1010
NA=NARGS-NL                            RPR1020
NARGS=NA                                RPR1030
CALL CHKCOLUMN (J)                        RPR1040
IF (J.NE.0) GO TO 50                      RPR1050
GO TO 160                                RPR1060
C   FIRST ARGUMENT IS NOT AN INTEGER, PARAMETERS ARE GIVEN. RPR1070
130  IF (ARGS(1).LE.0.) GO TO 50            RPR1080
IRVSP=ARGS(1)                            RPR1085
NPAR=(NARGS-1)/IRVSP                      RPR1090
IF (NPAR.NE.2.AND.NPAR.NE.3.AND.NPAR.NE.4) GO TO 30 RPR1100
A(1)=ABS(FLOAT(NPAR)*ARGS(1)+1.-FLOAT(NARGS)) RPR1110
IF (A(1).GT.0.0) GO TO 30                RPR1120
NA=ARGS(1)                                RPR1130
DO 150 I=1,NA                            RPR1140
ISUB=2+NPAR*(I-1)                         RPR1150
IRGS(I)=IARGS(1SUB)                       RPR1160
IARGS(I)=IARGS(ISUB)                      RPR1170

```

```

CALL ADDRESS (ISUB,IARGS(I)) RPR1180
IF (IARGS(I).LE.0) GO TO 50 RPR1190
NSIGDS(I)=IARGS(ISUB+1) RPR1200
NWMAX(I)=IARGS(ISUB+2)*(NPAR/3)+22*(1-NPAR/3) RPR1210
NF(I)=1 RPR1220
NWM(I)=0 RPR1230
IF (NWMAX(I).GT.0) GO TO 150 RPR1240
IF (NWMAX(I).LT.0) GO TO 140 RPR1250
C FLOATING RPR1260
NWIDTH(I)=NSIGDS(I)+5 RPR1270
NDECS(I)=NSIGDS(I)+2 RPR1280
NF(I)=-1 RPR1290
NWM(I)=-1 RPR1300
GO TO 150 RPR1310
C FIXED RPR1320
140 NWIDTH(I)=-NWMAX(I) RPR1330
NDECS(I)=NSIGDS(I) RPR1340
NSIGDS(I)=8 RPR1350
NWM(I)=-1 RPR1360
150 NBLANK(I)=IARGS(ISUB+3)*(NPAR/4)+3*(1-NPAR/4) RPR1370
IF (NPAR.EQ.4.AND.NBLANK(1).LT.1) NBLANK(1)=1 RPR1375
C LINES 160 TO 240 INITIALIZE AND CALL FXFARG RPR1380
160 IF (NERROR.NE.0) GO TO 400 RPR1390
IF (L1.NE.6) GO TO 210 RPR1400
I1=1 RPR1405
I2=0 RPR1410
165 LL=1 RPR1415
I2=MIN0(8,NA)+I2 RPR1420
NA=NA-8 RPR1425
DO 200 I=I1,I2 RPR1430
K=IARGS(I)+NRJ RPR1440
IF (NWMAX(I).LE.0) GO TO 170 RPR1450
CALL RFORMAT (RC(K),1,NSIGDS(I),NWIDTH(I),NDECS(I),NWMAX(I),A(1),A(RPR1460
11),0,0) RPR1470
170 IF (NPAR.EQ.1) NBLANK(I)=15-NWIDTH(I) RPR1480
CALL RFORMAT (A,NF(I),NSIGDS(I),NWIDTH(I),NDECS(I),NWM(I),RC(K),A(LRPR1490
1L),NBLANK(I),0) RPR1500
LL=LL+NWIDTH(I)+NBLANK(I) RPR1510
IF (NWIDTH(I).LT.NWMAX(I).OR.NWMAX(I).LE.0) GO TO 200 RPR1520
IF (NBLANK(I).EQ.0) GO TO 200 RPR1530
I5=LL-NDECS(I)+NSIGDS(I)-1 RPR1540
IF (A(I5)-AL(39)) 180,190,180 RPR1550
180 IF (A(I5)-AL(40)) 200,190,200 RPR1560
190 K=LL-NWIDTH(I)-1 RPR1570
A(K)=AL(41) RPR1580
200 CONTINUE RPR1590
NL=MIN0(LL-1,120) RPR1600
WRITE (IPRINT,410) (A(I),I=2,NL) RPR1610
IF (NA.LE.0) GO TO 400 RPR1615
I1=I2+1 RPR1620
GO TO 165 RPR1625
210 I1=1 RPR1630
DO 240 I=1,NA RPR1640
K=IARGS(I) RPR1650
C DETERMINE COUNT OF COL I RPR1660
NCOUNT(I)=NRMAX RPR1670
DO 220 J=1,NRMAX RPR1680
K1=K+NRMAX-J RPR1690
IF (ABS(RC(K1)).GT.0.) GO TO 230 RPR1700

```

```

220 NCOUNT(I)=NCOUNT(I)-1 RPR1710
230 IF (NCOUNT(I).GT.NRMAX-3) NCOUNT(I)=NRMAX RPR1720
    IF (NCOUNT(I).EQ.0) NWIDTH(I)=NWMAX(I) RPR1730
    IF (NCOUNT(I).EQ.0) GO TO 240 RPR1740
    IF (NWMAX(I).LE.0) GO TO 240 RPR1750
    CALL RFORMAT (RC(K),NCOUNT(I),NSIGDS(I),NWIDHT(I),NDECS(I),NWMAX(I)) RPR1760
1,A(1),A(1),0,0) RPR1770
240 IF (NPAR.EQ.1) NBLANK(I)=15-NWIDHT(I) RPR1780
C LINES 250 TO 390 CALL FXFORM AND PRINT IN READABLE FORMAT RPR1790
250 IF (L1.NE.8) CALL PAGE (4) RPR1800
    I4=0 RPR1810
    DO 260 I=I1,NA RPR1820
    I4=I4+NWIDHT(I)+NBLANK(I) RPR1830
    IF (I4.LE.120) GO TO 260 RPR1840
    I4=I-I1 RPR1850
    GO TO 270 RPR1860
260 IF (I.EQ.NA) I4=NA-I1+1 RPR1870
270 I2=I4+I1-1 RPR1880
    NARGS=I2-I1+1 RPR1890
C FROM HERE TO 290 PUTS IN COL HEADING IF FIRST ARG NOT A COL NO. RPR1900
    IF (NPAR.EQ.1) GO TO 290 RPR1910
    CALL RFORMAT (A,1,1,119,119,0,1.0,A(1),0,0) RPR1920
    LL=1 RPR1930
    DO 280 I=I1,I2 RPR1940
    LL=LL+NWIDHT(I)+NBLANK(I) RPR1950
    IF (NWIDHT(I).LT.6.OR.NCOUNT(I).EQ.0) GO TO 280 RPR1960
    A(200)=IRGS(I) RPR1970
    I5=FLOG10(A(200))+1.0 RPR1980
    CALL RFORMAT (A,1,I5,6,0,0,A(200),A(LL-6),0,0) RPR1990
    A(LL-1)=AL(45) RPR2000
    IF (NWIDHT(I).LT.12) GO TO 280 RPR2010
    A(LL-12)=AL(13) RPR2020
    A(LL-11)=AL(25) RPR2030
    A(LL-10)=AL(22) RPR2040
    A(LL-9)=AL(31) RPR2050
    A(LL-8)=AL(23) RPR2060
    A(LL-7)=AL(24) RPR2070
280 CONTINUE RPR2080
    WRITE (IPRINT,410) (A(I),I=2,LL) RPR2090
    GO TO 320 RPR2100
290 IF (L1.EQ.8) GO TO 325 RPR2110
    CALL HEADS (IRGS(I1),I4,0,1) RPR2120
    DO 310 I=I1,I2 RPR2130
    IF (NCOUNT(I).GT.0) GO TO 310 RPR2140
    I5=12*(I-I1)+1 RPR2150
    DO 300 I6=1,12 RPR2160
    LHEAD(I5)=L(45) RPR2170
300 I5=I5+1 RPR2180
310 CONTINUE RPR2190
    I5=12*I4 RPR2200
    WRITE (IPRINT,420) (LHEAD(I6),I6=1,I5) RPR2210
320 WRITE (IPRINT,410) RPR2220
325 DO 390 J=1,NRMAX RPR2230
    LL=1 RPR2240
    DO 350 I=I1,I2 RPR2250
    K=IARGS(I)+J-1 RPR2260
C PRINT BLANKS IF NCOUNT(I) LT NRMAX RPR2270
    IF (J.GT.NCOUNT(I)) NDECS(I)=NWIDHT(I) RPR2280
    CALL RFORMAT (A,NF(I),NSIGDS(I),NWIDHT(I),NDECS(I),NWM(I),RC(K),A(L)) RPR2290

```

```

1L),NBLANK(I),0) RPR2300
LL=LL+NWIDTH(I)+NBLANK(I) RPR2310
IF (NWIDTH(I).LT.NWMAX(I).OR.NWMAX(I).LE.0) GO TO 350 RPR2320
IF (NBLANK(I).EQ.0) GO TO 350 RPR2330
C PUT IN ASTERISK IF READABLE RETURNS FLOATING RPR2340
I5=LL-NDECS(I)+NSIGDS(I)-1 RPR2350
IF (A(I5)-AL(39)) 330,340,330 RPR2360
330 IF (A(I5)-AL(40)) 350,340,350 RPR2370
340 K=LL-NWIDTH(I)-1 RPR2380
A(K)=AL(41) RPR2390
350 CONTINUE RPR2400
C I=2 COMPENSATES FOR IX RPR2410
NL=LL-1 RPR2420
WRITE (IPRINT,410) (A(I),I=2,NL) RPR2430
C PRINT IN BLOCKS OF FIVE RPR2440
IF (J.EQ.NRMAX) GO TO 355 RPR2445
IF (MOD(J,5).EQ.0.AND.NRMAX.LE.48) WRITE (IPRINT,410) RPR2450
IF (MOD(J,10).EQ.0.AND.NRMAX.GT.48) WRITE (IPRINT,410) RPR2460
C CALL NEW PAGE IF NRMAX GT 50 RPR2470
355 IF(MOD(J,50).NE.0) GO TO 390 RPR2480
IF (J.EQ.NRMAX) GO TO 390 RPR2485
IF (L1.NE.8) CALL PAGE (4) RPR2490
IF(L1.EQ.8) GO TO 390 RPR2500
I5=12*I4 RPR2510
IF (NPAR.GT.1) GO TO 360 RPR2520
WRITE (IPRINT,420) (LHEAD(I6),I6=1,I5) RPR2530
GO TO 380 RPR2540
C FROM HERE TO 380 PUTS IN COL HEADING IF FIRST ARG NOT A COL NO. RPR2550
360 CALL RFORMAT (A,1,1,119,119,0,1.0,A(1),0,0) RPR2560
LL=1 RPR2570
DO 370 I=I1,I2 RPR2580
LL=LL+NWIDTH(I)+NBLANK(I) RPR2590
IF (NWIDTH(I).LT.6.OR.NCOUNT(I).EQ.0) GO TO 370 RPR2600
A(200)=IRGS(I) RPR2610
I5=FLOG10(A(200))+1.0 RPR2620
CALL RFORMAT (A,1,I5,6,0,0,A(200),A(LL-6),0,0) RPR2630
A(LL-1)=AL(45) RPR2640
IF (NWIDTH(I).LT.12) GO TO 370 RPR2650
A(LL-12)=AL(13) RPR2660
A(LL-11)=AL(25) RPR2670
A(LL-10)=AL(22) RPR2680
A(LL-9)=AL(31) RPR2690
A(LL-8)=AL(23) RPR2700
A(LL-7)=AL(24) RPR2710
370 CONTINUE RPR2720
WRITE (IPRINT,410) (A(I),I=2,LL) RPR2730
380 WRITE (IPRINT,410) RPR2740
390 CONTINUE RPR2750
C ADJUST FOR MORE THAN 8 COLUMNS RPR2760
IF (I2.EQ.NA) GO TO 400 RPR2770
IF(L1.NE.8) GO TO 145 RPR2772
WRITE(IPRINT, 410) RPR2778
145 I1=I2+1 RPR2780
GO TO 250 RPR2790
400 RETURN RPR2810
C RPR2820
410 FORMAT (1X,119A1) RPR2830
420 FORMAT (8(3X,12A1)) RPR2840
END RPR2850

```

```

SUBROUTINE SELECT                               SEL 10
C   VERSION 5.00      SELECT      5/15/70      SEL 20
COMMON /BLOCRC/ NRC,RC(12600)                 SEL 30
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NSEL 40
IARGS,VWXYZ(8),NERROR                         SEL 50
DIMENSION ARGS(100)                           SEL 60
EQUIVALENCE (ARGS(1),RC(12501))              SEL 70
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG          SEL 80
COMMON /SCRAT/ NS,NS2,A(13500)                SEL 90
C                                         SEL 100
C   ITYPE=1      SELECT IN COL ++ VALUES APPROX COL ++ TO WITHIN **, SEL 110
C               STORE IN COL ++
C   ITYPE=1      SELECT IN COL ++ VALUES APPROX COL ++ TO WITHIN **, SEL 130
C               STORE IN COL ++ TO COL ++
C   ITYPE=1      SELECT IN COL ++ VALUES APPROX COL ++ TO WITHIN **, SEL 150
C               STORE ++ TO ++, STORE NUMBER FALLING WITHIN TOL IN COSEL 160
C                                         SEL 170
C   ITYPE=2      SEARCH IN COL ++ FOR NUMBERS IN ++, TRANSFER CORRESP VSEL 180
C               FROM ++ INTO ++, ++ INTO ++, ETC           SEL 190
C                                         SEL 200
C   ITYPE=3      CENSOR COL ++ FOR $$, REPLACING BY $$, STORE IN COL ++SEL 210
C                                         SEL 220
C   ITPE=5      MATCH COLUMN (C) WITH (E), EXTRACT (E), PUT IN COLUMN (C)SEL 225
C WRITTEN BY DAVID HOGBEN SEL, NBS.    2/28/70. (CENSOR REVISED) SEL 230
C                                         SEL 235
GO TO (10,100,130,40,130), L2                  SEL 240
10  IF (KIND(3)) 50,20,50                      SEL 250
20  K=3                                         SEL 260
30  CALL ERROR (K)                           SEL 270
40  RETURN                                     SEL 280
50  IARGS(3)=IARGS(2)                         SEL 290
KIND(3)=0                                     SEL 300
IF (NARGS-4) 60,70,80                         SEL 310
60  K=10                                       SEL 320
GO TO 30                                      SEL 330
65  K=11                                       SEL 335
GO TO 30                                      SEL 337
70  IARGS(5)=IARGS(4)                         SEL 340
NARGS=NARGS+1                                 SEL 350
KIND(5)=KIND(4)                             SEL 360
80  IF (NARGS-6) 90,90,60                      SEL 370
90  IF (IARGS(4)-IARGS(5)) 230,230,20        SEL 380
100 IF (NARGS-4) 60,110,110                   SEL 390
110 IF (2*(NARGS/2)-NARGS) 60,120,60         SEL 400
120 CALL CHKCOL (J)                          SEL 410
IF (J) 20,200,20                            SEL 420
130 IF (NARGS-4) 60,140,60                      SEL 430
140 CALL ADRESS (1,I1)                        SEL 440
IF (I1) 20,65,150                            SEL 450
150 CALL ADRESS (2,I2)                        SEL 460
IF (I2) 160,65,170                            SEL 470
160 I2 = -I2                                  SEL 480
170 CALL ADRESS (3,I3)                        SEL 490
IF (I3) 180,65,190                            SEL 500
180 I3 = -I3                                  SEL 510
190 CALL ADRESS (4,I4)                        SEL 520
IF (I4) 20,65,200                            SEL 530
200 IF (NRMAX) 210,210,220                    SEL 540
210 K = 9                                    SEL 550

```

```

      GO TO 30                               SEL 555
220   IF (NERROR.NE.0) GO TO 40               SEL 560
      GO TO (250,450,520,40,520), L2          SEL 565
230   IF (IARGS(5)-IARGS(4)-NRMAX+1) 120,120,240  SEL 570
240   IARGS(5)=IARGS(4)+NRMAX-1              SEL 580
      GO TO 120                             SEL 590
C     SELECT                                SEL 600
250   DO 280 I=1,NRMAX                      SEL 610
      L=IARGS(1)+I-1                         SEL 620
      K=IARGS(2)+I-1                         SEL 630
      J=IARGS(4)+I-1                         SEL 640
      M=NRMAX+I                            SEL 650
      A(I)=RC(K)                           SEL 660
      A(M)=RC(L)                           SEL 670
260   RC(J)=0.0                            SEL 680
      IF (J-I-IARGS(5)+1) 270,280,280        SEL 690
270   J=NROW+J                           SEL 700
      GO TO 260                           SEL 710
280   CONTINUE                                SEL 720
      ARG3=ABS(ARGS(3))                     SEL 730
      DO 440 I=1,NRMAX                      SEL 740
      K=NRMAX+1                           SEL 750
      L=2*NRMAX                          SEL 760
      M=3*NRMAX                          SEL 770
      N=4*NRMAX                          SEL 780
      I1=IARGS(4)+I-1                     SEL 790
      J1=IARGS(6)+I-1                     SEL 800
      DO 300 J=K,L                        SEL 810
      AT=ABS(A(I)-A(J))                  SEL 820
      IF (ARG3-AT) 300,290,290            SEL 830 .
290   M=M+1                                SEL 840
      A(M)=AT                           SEL 850
      N=N+1                                SEL 860
      A(N)=A(J)                           SEL 870
300   CONTINUE                                SEL 880
      IF (M-3*NRMAX-1) 310,330,350        SEL 890
310   IF (NARGS-5) 440,440,320            SEL 900
320   RC(J1)=0.0                           SEL 910
      GO TO 440                           SEL 920
330   RC(I1)=A(N)                          SEL 930
      IF (NARGS-5) 440,440,340            SEL 940
340   RC(J1)=1.0                           SEL 950
      GO TO 440                           SEL 960
350   M1=3*NRMAX+2                         SEL 970
360   K2=0                                 SEL 980
      DO 380 J=M1,M                        SEL 990
      IF (A(J)-A(J-1)) 370,380,380        SEL1000
370   AT=A(J)                           SEL1010
      A(J)=A(J-1)                         SEL1020
      A(J-1)=AT                           SEL1030
      N=J+NRMAX                          SEL1040
      AT=A(N)                           SEL1050
      A(N)=A(N-1)                         SEL1060
      A(N-1)=AT                           SEL1070
      K2=K2+1                           SEL1080
380   CONTINUE                                SEL1090
      IF (K2) 390,390,360                SEL1100
390   N=4*NRMAX+1                         SEL1110
400   RC(I1)=A(N)                         SEL1120

```

```

I1=I1+NROW                               SEL1130
N=N+1                                     SEL1140
IF (N-M-NRMAX) 410,410,420                SEL1150
410 IF (I1-I-IARGS(5)+1) 400,400,420      SEL1160
420 IF (NARGS-5) 440,440,430                SEL1170
430 RC(J1)=M-3*NRMAX                      SEL1180
440 CONTINUE                                SEL1190
440 GO TO 40                                SEL1200
C SEARCH                                    SEL1210
450 I1=NARGS-1                             SEL1220
DO 470 I=1,NRMAX                          SEL1230
K=IARGS(1)+I-1                           SEL1240
L=IARGS(2)+I-1                           SEL1250
M=NRMAX+I                                 SEL1260
A(I)=RC(L)                                SEL1270
A(M)=RC(K)                                SEL1280
J1=2                                       SEL1290
DO 460 N=3,I1,2                           SEL1300
L=J1*NRMAX+I                            SEL1310
M=IARGS(N)+I-1                           SEL1320
A(L)=RC(M)                                SEL1330
460 J1=J1+1                                SEL1340
DO 470 N=4,NARGS,2                        SEL1350
M=IARGS(N)+I-1                           SEL1360
470 RC(M)=0.0                                SEL1370
K=NRMAX+1                                 SEL1380
L=2*NRMAX                                SEL1390
DO 510 I=1,NRMAX                          SEL1400
AT=ABS(A(I)/1.E8)                         SEL1410
DO 500 J=K,L                               SEL1420
IF (ABS(A(I)-A(J))-AT) 480,480,500      SEL1430
480 J1=1                                   SEL1440
DO 490 N=4,NARGS,2                        SEL1450
M=IARGS(N)+I-1                           SEL1460
I1=J1*NRMAX+J                            SEL1470
RC(M)=A(I1)                                SEL1480
490 J1=J1+1                                SEL1490
GO TO 510                                SEL1500
500 CONTINUE                                SEL1510
510 CONTINUE                                SEL1520
510 GO TO 40                                SEL1530
C CENSOR OR MATCH                         SEL1540
520 DO 580 I=1,NRMAX                          SEL1550
IF (RC(I1)-RC(I2)) 540,530,560          SEL1560
530 RC(I4) = RC(I3)                         SEL1570
GO TO 570                                SEL1580
540 IF (L2-4) 530,40,570                  SEL1590
560 IF (L2.EQ.5) GO TO 570                SEL1600
RC(I4) = RC(I1)                           SEL1610
570 I1 = I1+1                                SEL1620
IF (KIND(2).EQ.0) I2=I2+1                SEL1630
IF (KIND(3).EQ.0) I3=I3+1                SEL1640
580 I4 = I4+1                                SEL1650
GO TO 40                                  SEL1660
END                                     SEL1670

```

```

SUBROUTINE SET                               SET 10
C   VERSION 5.00      SET      5/15/70      SET 20
COMMON /BLOCKA/ MODE,M,KARD(83),KARG,ARG,ARG2,NEWCD(80),KRDEND  SET 30
COMMON /BLOCRC/ NRC,RC(12600)                SET 40
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NSET 50
IARGS,VWXYZ(8),NERROR                      SET 60
DIMENSION ARGS(100)                         SET 70
EQUIVALENCE (ARGS(1),RC(12501))            SET 80
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG        SET 90
COMMON /QRS/ NDROW,IFLAG,J,NNARG           SET 100
C   THE FOLLOWING CARDS ARE NEEDED ONLY FOR TAPE OPERATIONS SET 110
COMMON /TAPE/ NAME4(2),NTPCT,IPUNCP,INUNIP,L1TP          SET 120
COMMON /BLOCKC/ KIO,INUNIT,ISCRAT,KBDOUT,KRDKNT,LLIST     SET 130
C *****
ISRFLG=1                                     SET 140
IF (NARGS.EQ.1.OR.NARGS.EQ.2) GO TO 10       SET 150
CALL ERROR (10)                             SET 160
GO TO 70                                    SET 170
10   MODE=2                                   SET 180
CALL ADRESS (NARGS,J)                      SET 190
IF (J) 20,30,40                            SET 200
20   CALL ERROR (3)                          SET 210
GO TO 70                                    SET 220
30   CALL ERROR (11)                          SET 230
GO TO 70                                    SET 240
40   NDROW=J+NROW-1                         SET 250
IF (NARGS.EQ.1) GO TO 60                   SET 260
IF (KIND(1).NE.0) GO TO 20                 SET 270
IF (IARGS(1).LE.NROW.AND.IARGS(1).GT.0) GO TO 50    SET 280
CALL ERROR (16)                           SET 290
GO TO 70                                    SET 300
50   J=J+IARGS(1)-1                         SET 310
60   IFLAG=0                                 SET 320
MODE=2                                     SET 330
GO TO 80                                    SET 340
70   IFLAG=1                                 SET 350
C   THE FOLLOWING CARDS ARE NEEDED ONLY FOR TAPE OPERATIONS SET 360
IF (L1TP.NE.48.AND.L1TP.NE.49) RETURN      SET 370
MODE=1                                     SET 380
INUNIT=INUNIP                            SET 390
C *****
80   RETURN                                  SET 400
END                                         SET 410
                                              SET 420
                                              SET 430

```

```

SUBROUTINE SETQ                                     STQ 10
C      VERSION 5.00      SETQ      5/15/70          STQ 20
COMMON /BLOCRC/ NRC,RC(12600)                      STQ 30
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NSTQ 40
IARGS,VWXYZ(8),NERROR                           STQ 50
DIMENSION ARGS(100)                             STQ 60
EQUIVALENCE (ARGS(1),RC(12501))                 STQ 70
COMMON /QRS/ NDROW,IFLAG,J,NNARG                STQ 80
COMMON /BLOCKA/ MODE,M,KARD(83),KARG,ARG,ARG2,NEWCD(80),KRDEND STQ 90
COMMON /BLOCKC/ KIO,INUNIT,ISCRAT,KBDOUT,KRDKNL,LLIST STQ 100
COMMON /TAPE/ NAME4(2),NTPCT,IPUNCP,INUNIP,L1TP   STQ 110
C      CHECK IF END OF ROW HAS BEEN EXCEEDED PREVIOUSLY IN THIS SET. STQ 120
IF (IFLAG.NE.0.OR.NARGS.EQ.0) GO TO 80           STQ 130
C      J IS WHERE NEXT DATA ITEM IS TO GO IN COLUMN      STQ 140
C      JJ IS WHERE LAST DATA ITEM OF THIS SET IS TO GO    STQ 150
C      NDROW IS ADDRESS OF LAST ELEMENT OF ROW.          STQ 160
JJ=J+NARGS-1                                      STQ 170
IF (JJ.LE.NDROW) GO TO 10                         STQ 180
CALL ERROR (201)                                  STQ 190
IFLAG=1                                         STQ 200
IF (J.GT.NDROW) GO TO 80                         STQ 210
JJ=NDROW                                         STQ 220
C      THE FOLLOWING CARDS ARE NEEDED ONLY FOR TAPE OPERATIONS STQ 230
10     IF (L1TP.NE.48) GO TO 50                     STQ 240
K=1                                              STQ 250
DO 30 I=J,JJ                                      STQ 260
IF (KIND(K).EQ.0) GO TO 20                         STQ 270
IF (ARGS(K).NE.0.) GO TO 50                         STQ 280
GO TO 30                                         STQ 290
20     IF (IARGS(K).NE.0) GO TO 50                   STQ 300
30     CONTINUE                                       STQ 310
40     INUNIT=INUNIP                                STQ 320
MODE=1                                           STQ 330
RETURN                                         STQ 340
C      THIS STATEMENT WAS 10 K=1                      STQ 350
50     K=1                                           STQ 360
C *****                                         STQ 370
DO 70 I=J,JJ                                      STQ 380
IF (KIND(K).EQ.0) GO TO 60                         STQ 390
RC(I)=ARGS(K)                                    STQ 400
GO TO 70                                         STQ 410
60     RC(I)=IARGS(K)                                STQ 420
70     K=K+1                                         STQ 430
J=JJ+1                                         STQ 440
NRMAX=MAX0(NRMAX,JJ-NDROW+NROW)                  STQ 450
C      THE FOLLOWING CARDS ARE NEEDED ONLY FOR CSET TAPE      STQ 460
IF (L1TP.NE.49) RETURN                            STQ 470
NTPCT=NTPCT-1                                    STQ 480
IF (NTPCT.EQ.0) GO TO 40                         STQ 490
C *****                                         STQ 500
80     RETURN                                         STQ 510
END                                             STQ 520

```

```

SUBROUTINE SETUP STP 10
C VERSION 5.00 SETUP 5/15/70 STP 20
COMMON /BLOCKA/ MODE,M,KARD(83),KARG,ARG,ARG2,NEWCD(80),KRDEND STP 30
COMMON /BLOCKC/ KIO,INUNIT,ISCRAT,KBDOUT,KRDKNT,LLIST STP 40
COMMON /BLOCRC/ NRC,RC(12600) STP 50
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NSTP 60
IARGS,VWXYZ(8),NERROR STP 70
DIMENSION ARGS(100) STP 80
EQUIVALENCE (ARGS(1),RC(12501)) STP 90
COMMON /HEADER/NOCARD(80),ITLE(60,6),LNCNT,IPRINT,NPAGE,IPUNCH STP 100
COMMON /PKSWT/IHCNT,IHTP STP 110
COMMON /SCRAT/ NS,NS2,A(13500) STP 120
COMMON /ICODE/ NIR,NID,NIRD,LIR,LID,LIRD STP 130
COMMON /BLOCKX/ INDEX(6,8),LEVEL STP 140
COMMON /PCONST/JPC,P(40),N(40) STP 150
COMMON /CONSTS/ PI,E,HALFPI,DEG,RAD,XALOG STP 160
C THE FOLLOWING CARD IS NEEDED ONLY FOR TAPE OPERATIONS STP 170
COMMON /ICODTP/ NITP,LITP STP 180
COMMON /TAPE/ NAME4(2),NTPCT,IPUNCP,INUNIP,L1TP STP 190
***** STP 200
P(1)=PI STP 210
P(2)=PI STP 220
P(3)=E STP 230
P(4)=E STP 240
KRDEND=80 STP 250
NERROR=0 STP 260
LEVEL=0 STP 270
MODE=1 STP 280
IPRINT=6 STP 290
IPUNCH=3 STP 300
INUNIT=5 STP 310
ISCRAT=45 STP 320
NS=13500 STP 330
KIO=0 STP 340
CALL AERR (-1) STP 350
NRC=12500 STP 360
NS2=NS/2 STP 370
C THESE VARIABLES MUST BE REDEFINED IF A NEW COMMAND IS ADDED STP 380
NIR=246 STP 390
NIRD=29 STP 400
NID=8 STP 410
LIR=300 STP 420
LID=9 STP 430
LIRD=30 STP 440
C THE FOLLOWING CARD IS NEEDED ONLY FOR TAPE OPERATIONS STP 450
NITP=9 STP 460
LITP=10 STP 470
INUNIP=INUNIT STP 480
IPUNCP=IPUNCH STP 490
***** STP 500
C ***** STP 510
C ** THESE SWITCHES MUST BE SET BEFORE COMPILING. NEEDED INFORMATION STP 520
C FOR PACKING HEADS AND FORMATS. STP 530
C ***** STP 540
C IHTP= NO. OF HEADINGS PERMITTED. STP 550
C THIS IS SET = 50 , HOWEVER IN ORDER TO SAVE SPACE ONE MAY DESIRE STP 560
C TO PERMIT FEWER HEADINGS. IF SO DIMENSION STATEMENT IN PREPAK STP 570
C FOR VARIABLE IHEAD MUST BE CHANGED. SECOND VALUE OF IHEAD INDICATES TOTAL NO. OF HEADINGS STP 580
C STP 590

```

```

C      IHTP=50
C
C      RETURN
C      END

SUBROUTINE SKSYMV (A,NROW,N,K)          SKS 10
C      VERSION 5.00      SKSYMV   5/15/70    SKS 20
C      FOR OMNITAB MATRIX   S PEAVY  1/ 3/68    SKS 30
C
C      A MATRIX TO BE TESTED FOR SKEW SYMMETRY    SKS 40
C      NROW DIMENSION OF A    SKS 60
C      N      PRESENT SIZE OF MATRIX    SKS 70
C      K      STATUS    SKS 80
C          K=2 NO SYMMETRY    SKS 90
C          K=3 EXACT SKEW SYMMETRY    SKS 100
C          K=4 RELATIVE (1.E-7) SKEW SYMMETRY    SKS 110
C
C      DIMENSION A(NROW,NROW)    SKS 120
C      K=3
C      NN=N-1
C      DO 40 J=1,NN    SKS 130
C      IF (A(J,J).EQ.0.0) GO TO 10    SKS 140
C      K=2
C      RETURN
C      I=J+1    SKS 150
C      DO 40 L=I,N    SKS 160
C      IF (A(L,J).NE.0.) GO TO 20    SKS 170
C      T=ABS(A(J,L))    SKS 180
C      GO TO 30    SKS 190
C      T=ABS(1.0+A(L,J)/A(J,L))    SKS 200
C      IF (T.EQ.0.0) GO TO 40    SKS 210
C      K=4
C      IF (T.LE.1.E-7) GO TO 40    SKS 220
C      K=2
C      RETURN
C      CONTINUE
C      IF (A(N,N).NE.0.0) K=2    SKS 230
C      RETURN
C      END

```

```

SUBROUTINE SORDER                               SOD 10
C VERSION 5.00      SORDER      5/15/70        SOD 20
COMMON /BLOCRC/ NRC,RC(12600)                  SOD 30
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NSOD 40
IARGS,VWXYZ(8),NERROR                         SOD 50
DIMENSION ARGS(100)                           SOD 60
EQUIVALENCE (ARGS(1),RC(12501))              SOD 70
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG          SOD 80
COMMON /SCRAT/ NS,NS2,A(13500)                SOD 90
C           SUBROUTINE BY CARLA MESSINA 221.04 JUNE 1967 SOD 100
C L2=8 FOR SORT, L2=9 FOR ORDER, L2=14 FOR HEIRARCHY SOD 110
C                                         SOD 120
C TYPE 1 IS      HEIRARCHY OF COL ++, STORE IN COL ++
C               HEIRARCHY GIVES THE ROW LOCATION OF THE SMALLEST NO. OF THE SOD 140
C THE FIRST COLUMN IN THE FIRST ROW OF THE SECOND COLUMN SOD 150
C THE ROW NO. OF THE SECOND LOWEST NO. OF THE FIRST COLUMN IS STOREDSOD 160
C IN THE SECOND ROW OF THE SECOND COLUMN, ..... THE ROW NO. OF THE SOD 170
C LARGEST NO. OF THE FIRST COL IS STORED IN THE NRMAX ROW OF THE 2NDSOD 180
C COLUMN. THE FIRST COLUMN IS UNCHANGED BY THIS COMMAND. SOD 190
C TYPE 2 IS      ORDER COLUMNS ++,++,++, ETC          SOD 200
C               ORDER PLACES EACH ONE OF THE GIVEN COLUMNS IN NUMERICALLY SOD 210
C INCREASING ORDER.                                     SOD 220
C TYPE 3 IS      SORT COL ++ CARRY ALONG COLUMNS ++,++, ETC SOD 230
C               SORT PLACES THE FIRST COLUMN IN NUMERICALLY INCREASING ORDER SOD 240
C WHILE PRESERVING THE ROW RELATIONSHIPS AMONG THE GIVEN COLUMNS SOD 250
C                                         SOD 260
C THESE INSTRUCTIONS CAN BE DONE FASTER IF A MACHINE LANGUAGE SOD 270
C PROGRAM IS SUBSTITUTED FOR THIS ONE.                 SOD 280
C                                         SOD 290
IF (NARGS) 10,10,40                            SOD 300
10 K=10                                         SOD 310
20 CALL ERROR (K)                           SOD 320
30 RETURN                                       SOD 330
40 CALL CHKCOL (J)                           SOD 340
IF (J) 50,60,50                                SOD 350
50 K=3                                         SOD 360
GO TO 20                                      SOD 370
60 IF (L2-9) 80,80,70                            SOD 380
70 IF (NARGS-2) 10,80,10                            SOD 390
80 IF (NERROR) 30,90,30                            SOD 400
90 IF (NRMAX-1) 100,110,120                          SOD 410
100 K=9                                         SOD 420
GO TO 20                                      SOD 430
110 IF (L2-9) 30,30,210                            SOD 440
120 K3=1                                         SOD 450
K=IARGS(1)-1                                    SOD 460
130 DO 140 I=1,NRMAX                           SOD 470
J=K+I                                         SOD 480
L=NRMAX+I                                      SOD 490
A(I)=RC(J)                                     SOD 500
140 A(L)=I                                       SOD 510
K1=NRMAX                                       SOD 520
150 K1=K1-1                                       SOD 530
K2=0                                         SOD 540
IF (K1-1) 160,160,170                            SOD 550
160 K1=2                                         SOD 560
170 DO 190 I=1,K1                                SOD 570
IF (A(I)-A(I+1)) 190,190,180                  SOD 580
180 CC=A(I)                                     SOD 590

```

```

A(I)=A(I+1)          SOD 600
A(I+1)=CC            SOD 610
L=NRMAX+I            SOD 620
CC=A(L)              SOD 630
A(L)=A(L+1)          SOD 640
A(L+1)=CC            SOD 650
K2=1                 SOD 660
190 CONTINUE          SOD 670
IF (K2) 150,200,150   SOD 680
200 IF (L2-9) 240,240,220  SOD 690
210 A(NRMAX+1)=1.0    SOD 700
220 K=IARGS(2)-1      SOD 710
DO 230 I=1,NRMAX    SOD 720
J=K+I                SOD 730
L=NRMAX+I            SOD 740
230 RC(J)=A(L)        SOD 750
GO TO 30              SOD 760
240 DO 250 I=1,NRMAX  SOD 770
J=K+I                SOD 780
250 RC(J)=A(I)        SOD 790
IF (NARGS-2) 30,260,260  SOD 800
260 IF (L2-9) 290,270,270  SOD 810
270 IF (NARGS-K3) 30,30,280  SOD 820
280 K3=K3+1            SOD 830
K=IARGS(K3)-1        SOD 840
GO TO 130             SOD 850
290 DO 310 I=2,NARGS  SOD 860
K=IARGS(I)-1          SOD 870
DO 300 J=1,NRMAX    SOD 880
L=NRMAX+J            SOD 890
J1=A(L)+FLOAT(K)     SOD 900
300 A(J)=RC(J1)        SOD 910
DO 310 J=1,NRMAX    SOD 920
J1=K+J                SOD 930
310 RC(J1)=A(J)        SOD 940
GO TO 30              SOD 950
END                  SOD 960

```

```

SUBROUTINE SORTSM (N,SUM)          SOM 10
C VERSION 5.00      SORTSM      5/15/70   SOM 20
C *
C COMMON /SCRAT/ NS,NS2,A(13500)    SOM 40
C SORT COLUMN OF PRODUCTS FOR MATRIX MULTIPLICATION    SOM 50
C AFTER SORTING START SUMMING BEGIN IN MIDDLE OF SORTED COLUMN    SOM 60
C *
C DIMENSION X(1)                  SOM 80
C DOUBLE PRECISION X,SAVE,SUM     SOM 90
C EQUIVALENCE (X,A)              SOM 100
C IF (N.NE.1) GO TO 10            SOM 110
C SUM=X(NS2)                     SOM 120
C RETURN                          SOM 130
10 K=0                            SOM 140
C IS=NS2                          SOM 150
C DO 30 I=2,N                      SOM 160
C IF (X(IS)-X(IS-1)) 20,30,30    SOM 170
20 SAVE=X(IS-1)                   SOM 180
C X(IS-1)=X(IS)                   SOM 190
C X(IS)=SAVE                      SOM 200
C K=1                            SOM 210
30 IS=IS-1                        SOM 220
C IF (K.NE.0) GO TO 10            SOM 230
C NP=N/2                          SOM 240
C IF (MOD(N,2).EQ.0) GO TO 40    SOM 250
C NPA=NS2-NP-1                   SOM 260
C NPB=NPA+2                       SOM 270
C NPC=NS2-NP                      SOM 280
C SUM=X(NPC)                     SOM 290
C GO TO 50                         SOM 300
40 SUM=0.D0                        SOM 310
C NPA=NS2-NP                      SOM 320
C NPB=NPA+1                       SOM 330
50 DO 60 I=1,NP                   SOM 340
C SUM=SUM+X(NPA)+X(NPB)           SOM 350
C NPA=NPA-1                       SOM 360
C NPB=NPB+1                       SOM 370
60 CONTINUE                         SOM 380
C RETURN                           SOM 390
C END                             SOM 400

```

```

SUBROUTINE SPACE                                SPA 10
C      VERSION 5.00     SPACE      5/15/70          SPA 20
COMMON /BLOCRC/ NRC,RC(12600)                  SPA 30
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NSPA 40
IARGS,VWXYZ(8),NERROR                         SPA 50
DIMENSION ARGS(100)                           SPA 60
EQUIVALENCE (ARGS(1),RC(12501))              SPA 70
COMMON/HEADER/NOCARD(80),ITLE(60,6),LNCNT,IPRINT,NPAGE,IPUNCH SPA 80
IF (NARGS-1) 40,32,10                         SPA 100
10    I=10                                     SPA 110
20    CALL ERROR (I)                          SPA 120
30    RETURN                                    SPA 130
32    IF (KIND(1).EQ.0) IF(IARGS(1)) 35,30,45 SPA 135
      I=20                                     SPA 140
      GO TO 20                                 SPA 150
35    I=3                                      SPA 160
      GO TO 20                                 SPA 170
40    IARGS(1)=1                               SPA 180
45    J=MIN0(60,IARGS(1))                     SPA 190
      IF (NERROR.NE.0) GO TO 30                SPA 200
      DO 50 I=1,J                            SPA 210
50    WRITE (IPRINT,60)                      SPA 220
      GO TO 30                                 SPA 230
C
60    FORMAT (1X)                           SPA 240
      END                                     SPA 250
                                         SPA 260

```

```

SUBROUTINE SPINV (A,M,KK,ISIG) SPI 10
C VERSION 5.00      SPINV      5/15/70 SPI 20
C 7058MI MATRIX INVERSION WITH MINIMUM ROUNDOFF ERROR ACCUMULATION. SPI 30
DATA ONE/1.0/,ZERO/0.0/,ER/1.E-8/ SPI 40
DIMENSION A(1) SPI 50
ISIG=0 SPI 60
N=M SPI 70
NN=KK SPI 80
N2=N+N SPI 90
DO 30 J=1,N SPI 100
NJCOL=(N+J-1)*NN SPI 110
DO 30 I=1,N SPI 120
KINJ=NJCOL+I SPI 130
IF (I-J) 10,20,10 SPI 140
10 A(KINJ)=ZERO SPI 150
GO TO 30 SPI 160
20 A(KINJ)=ONE SPI 170
30 CONTINUE SPI 180
C DETERMINE MAXIMUM ABS OF VARIABLE BEING ELIMINATED. THIS BECOMES SPI 190
L=0 SPI 200
40 L=L+1 SPI 210
LCOL=NN*L-NN SPI 220
KLL=LCOL+L SPI 230
IF (L-N) 50,100,200 SPI 240
C FIND THE LARGEST ELEMENT IN THE LTH COLUMN. SPI 250
50 J1=L SPI 260
C=ABS(A(KLL)) SPI 270
L1=L+1 SPI 280
DO 70 I=L1,N SPI 290
KIL=LCOL+I SPI 300
X=ABS(A(KIL)) SPI 310
IF (C-X) 60,70,70 SPI 320
C RECORD THE NUMBER OF THE ROW HAVING THE GREATER ELEMENT. SPI 330
60 J1=I SPI 340
C C BECOMES THE GREATER. SPI 350
C=X SPI 360
70 CONTINUE SPI 370
C INTERCHANGE ROW J1 WITH ROW L. J1 IS THE ROW WITH THE LARGEST ELEM SPI 380
C TEST TO SEE IF INTERCHANGING IS NECESSARY. SPI 390
IF (J1-L) 80,100,80 SPI 400
80 DO 90 J=L,N2 SPI 410
JCOL=NN*j-NN SPI 420
KJIJ=JCOL+J1 SPI 430
HOLD=A(KJIJ) SPI 440
KLJ=JCOL+L SPI 450
A(KJIJ)=A(KLJ) SPI 460
A(KLJ)=HOLD SPI 470
90 CONTINUE SPI 480
C IF THE LARGEST ABSOLUTE ELEMENT IN A COLUMN IS ZERO WE HAVE A SIN SPI 490
100 IF (ABS(A(KLL))-ER) 110,110,120 SPI 500
110 ISIG=4 SPI 510
GO TO 200 SPI 520
C ZERO ALL THE ELEMENTS IN THE LTH COLUMN BUT THE PIVOTAL ELEMENT. SPI 530
120 L1=1 SPI 540
L2=L-1 SPI 550
IF (L2) 130,130,150 SPI 560
130 IF (L-N) 140,170,140 SPI 570
140 L1=L+1 SPI 580
L2=N SPI 590

```

```

150 DO 160 I=L1,L2                               SPI 600
      KIL=LCOL+I                                 SPI 610
      Z=-A(KIL)/A(KLL)                           SPI 620
      DO 160 J=L,N2                               SPI 630
      JCOL=NN*N-J-NN                            SPI 640
      KIJ=JCOL+I                                SPI 650
      KLJ=JCOL+L                                SPI 660
160  A(KIJ)=A(KIJ)+Z*A(KLJ)                     SPI 670
      IF (N-L2) 40,40,130                         SPI 680
C   DIVIDE BY DIAGONAL ELEMENTS.                SPI 690
170  DO 180 I=1,N                                SPI 700
      KKK=NN*I-NN+I                             SPI 710
      ZZ=A(KKK)                                 SPI 720
      DO 180 J=1,N2                            SPI 730
      KKI=NN*N-J-NN+I                          SPI 740
180  A(KKI)=A(KKI)/ZZ                          SPI 750
C   RETURN AFTER PUTTING A INVERSE INTO B       SPI 760
      DO 190 J=1,N                            SPI 770
      JCOL=NN*N-J-NN                          SPI 780
      NJCOL=NN*N+JCOL                         SPI 790
      DO 190 I=1,N                            SPI 800
      KIJ=JCOL+I                                SPI 810
      KINJ=NJCOL+I                            SPI 820
190  A(KIJ)=A(KINJ)                           SPI 830
200  RETURN                                  SPI 840
      END                                     SPI 850

```

```

SUBROUTINE STATIS
C      VERSION 5.00      STATUS    5/15/70      STA 10
C      S PEAVY          STA 20
C      OMNITAB COMMON IS AS FOLLOWS      STA 30
C      I WITH WEIGHTS      STA 40
C          A. STATIS COL +++ WEIGHTS +++ START STORING RESULTS +++
C                  (RESULTS WILL BE STORED IN THE NEXT 4 COL)      STA 60
C          B. STATIS COL +++ WHTS +++ RESULTS +++,++,++,++,++      STA 70
C          STA 80
C      II WITHOUT WHTS      STA 90
C          A. SAME AS I. A. EXCEPT WHTS COL OMITTED      STA 100
C          B. SAME AS I. B. EXCEPT WHTS COL OMITTED      STA 110
COMMON /BLOCRC/ NRC,RC(12600)      STA 120
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NSTA 130
IARGS,VWXYZ(8),NERROR      STA 140
DIMENSION ARGS(100)      STA 150
EQUIVALENCE (ARGS(1),RC(12501))      STA 160
COMMON /SCRAT/ NS,NS2,A(13500)      STA 170
COMMON/HEADER/NOCARD(80),ITLE(60,6),LNCNT,IPRINT,NPAGE,IPUNCH      STA 180
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG      STA 200
DIMENSION SA(3125,3), ISA(3125)      STA 210
DIMENSION IB(10)      STA 220
EQUIVALENCE (A(101),ISA), (A(3226),SA)      STA 230
DIMENSION BCON(4), BKCON(4), AKCON(4), AT5(6), CK1(6), DK2(6), XK1STA 240
1(7), YK2(7)      STA 250
DATA BCON(1),BCON(2),BCON(3),BCON(4)/3.6948,-1.6561,.406,2.7764/,BSTA 260
1KCON(1),BKCON(2),BKCON(3),BKCON(4)/7.45894,-.89082,.61522,2.56706/STA 270
2,AKCON(1),AKCON(2),AKCON(3),AKCON(4)/-.51732,-.61863,-.04122,.5589STA 280
37/,AT5(1),AT5(2),AT5(3),AT5(4),AT5(5),AT5(6)/1.9599640,2.3722712,2STA 290
4.8224986,2.5558497,1.5895341,.7328982/,CK1(1),CK1(2),CK1(3),CK1(4)STA 300
5,CK1(5),CK1(6)/-.70285,-.02006,-.01687,-.01447,-.01263,.67839/,DK2STA 310
6(1),DK2(2),DK2(3),DK2(4),DK2(5),DK2(6)/-1.49016,.13384,.09764,.074STA 320
776,.05931,1.68641/,XK1(1),XK1(2),XK1(3),XK1(4),XK1(5),XK1(6),XK1(7STA 330
8)/-40.343875,14.1365,-2.743342,.84143957,.001066,-6.3701507E-6,1.7STA 340
949484E-8/,YK2(1),YK2(2),YK2(3),YK2(4),YK2(5),YK2(6),YK2(7)/50.2982STA 350
$33,-11.395210,6.0537922,1.1542370,-9.8051279E-4,5.5609437E-6,1.458STA 360
$4433E-8/,CONK/1.959964/      STA 370
DATA ZERO/0.0/,ONE/1.0/,TWO/2.0/      STA 380
IF (L2.EQ.1.OR.NARGS.NE.1) GO TO 5      STA 385
CALL ERROR (236)      STA 387
RETURN      STA 390
5      DO 10 I=1,60      STA 395
10      A(I)=0.0      STA 400
      NXCOL=IARGS(1)      STA 410
      NXWT=IARGS(2)      STA 420
      ISTORE=1      STA 430
      NAR=NARGS      STA 440
      IWT=1      STA 450
      IF (NARGS.EQ.1) GO TO 30      STA 460
      IF (NARGS.EQ.3.AND.IARGS(NARGS).LT.0) GO TO 20      STA 470
      GO TO 40      STA 480
20      NARGS=NARGS-1      STA 490
      IWT=2      STA 500
30      ISTORE=2      STA 510
      GO TO 50      STA 520
40      IF (NARGS.NE.2.AND.NARGS.NE.3.AND.NARGS.NE.5.AND.NARGS.NE.6) CALL STA 530
1ERROR (10)      STA 540
50      J=NARGS      STA 550
      CALL CKIND (J)      STA 560
      IF (J.NE.0) CALL ERROR (3)      STA 570

```

```

CALL CHKCOL (J) STA 580
IF (J.NE.0) CALL ERROR (11) STA 590
IF (NRMAX*4.LE.NS) GO TO 60 STA 600
CALL ERROR (214) STA 610
RETURN STA 620
60 IF (NERROR.NE.0) RETURN STA 630
IXN=NRMAX STA 640
A(1)=NRMAX STA 650
K=IARGS(1) STA 660
M=1 STA 670
IF (NAR.EQ.3.OR.NAR.EQ.6) GO TO 80 STA 680
NZW=NRMAX STA 690
SUM=0.0 STA 700
S2=0. STA 710
WT=0.0 STA 720
ASUMWT=0. STA 730
DO 70 I=1,IXN STA 740
SA(I,2)=RC(K) STA 750
SA(I,3)=1.0 STA 760
ISA(I)=M STA 770
SA(I,1)=RC(K) STA 780
K=K+1 STA 790
M=M+1 STA 800
WT=WT+1. STA 810
SUM=SUM+SA(I,2) STA 820
ASUMWT=ASUMWT+ABS(SA(I,2)) STA 830
70 S2=S2+SA(I,2)**2 STA 840
SUMWT=SUM STA 850
GO TO 110 STA 860
80 SUM=0. STA 870
WT=0. STA 880
SUMWT=0.0 STA 890
MA=IARGS(2) STA 900
S2=0.0 STA 910
IWT=2 STA 920
NEGWT=0 STA 930
ASUMWT=0. STA 940
DO 100 I=1,IXN STA 950
IF (RC(MA).EQ.0.) GO TO 90 STA 960
IF (RC(MA).LT.0.0) NEGWT=NEGWT+1 STA 970
SA(M,2)=RC(K) STA 980
SA(M,3)=RC(MA) STA 990
ISA(M)=M STA1000
SA(M,1)=RC(K) STA1010
S2=S2+SA(M,2)**2*RC(MA) STA1020
SUM=SUM+RC(K) STA1030
WT=WT+RC(MA) STA1040
SUMWT=SA(M,2)*RC(MA)+SUMWT STA1050
ASUMWT=ASUMWT+ABS(SA(M,2))*RC(MA) STA1060
M=M+1 STA1070
90 K=K+1 STA1080
100 MA=MA+1 STA1090
NZW=M-1 STA1100
IF (NEGWT.GT.0) CALL ERROR (223) STA1110
IF (NZW.GT.0) GO TO 110 STA1120
CALL ERROR (224) STA1130
RETURN STA1140
110 A(2)=NZW STA1150
A(3)=SUM/A(2) STA1160

```

```

A(4)=SUMWT / WT           STA1170
A(24)=(2.0*A(2)-1.)/3.0   STA1180
A(25)=FSQRT((16.*A(2)-29.)/90.) STA1190
A(39)=S2                  STA1200
A(42)=ASUMWT              STA1210
A(43)=ASUMWT / WT          STA1220
IXN=NZW                   STA1230
IXNM1=IXN-1                STA1240
120 IST=0                  STA1250
DO 130 I=2,IXN             STA1260
IF (SA(I-1,1).LE.SA(I,1)) GO TO 130 STA1270
K=ISA(I-1)                 STA1280
ISA(I-1)=ISA(I)             STA1290
ISA(I)=K                   STA1300
T=SA(I-1,1)                 STA1310
SA(I-1,1)=SA(I,1)            STA1320
SA(I,1)=T                   STA1330
IST=1                      STA1340
130 CONTINUE                STA1350
IF (IST.NE.0) GO TO 120     STA1360
NALPHA=.25*A(2)              STA1370
IXA=NALPHA+1                 STA1380
IXNA=IXN-NALPHA              STA1390
TSUM=0.                      STA1400
TWSUM=0                      STA1410
TWT=0                        STA1420
DO 140 I=IXA,IXNA             STA1430
M=ISA(I)                     STA1440
TWSUM=TWSUM+SA(I,1)*SA(M,3)    STA1450
TWT=TWT+SA(M,3)               STA1460
140 TSUM=TSUM+SA(I,1)           STA1470
A(7)=TSUM/(A(2)-2.*FLOAT(NALPHA)) STA1480
A(8)=TWSUM/TWT                STA1490
N2=(NZW+1)/2                  STA1500
A(5)=SA(N2,1)                 STA1510
IF (MOD(NZW,2).EQ.0) A(5)=(A(5)+SA(N2+1,1))/TWO STA1520
A(6)=(SA(1,1)+SA(IXN,1))/TWO STA1530
A(11)=SA(IXN,1)-SA(1,1)        STA1540
A(34)=SA(1,1)                 STA1550
A(35)=SA(IXN,1)                STA1560
DELX=A(11)/10.                 STA1570
XB=SA(1,1)                     STA1580
XT=XB+DELX                     STA1590
L=1                           STA1600
DO 170 I=1,10                  STA1610
IC=0                           STA1620
150 IF (SA(L,1).GE.XT) GO TO 160 STA1630
IC=IC+1                         STA1640
L=L+1                           STA1650
IF (L.NE.IXN) GO TO 150         STA1660
160 A(I+50)=IC                  STA1670
170 XT=XT+DELX                  STA1680
IF (L.GT.IXN) GO TO 190         STA1690
DO 180 I=L,IXN                  STA1700
IF (SA(I,1).GE.XT-DELX) A(60)=A(60)+1. STA1710
180 CONTINUE                      STA1720
190 DO 200 I=1,IXNM1              STA1730
200 SA(I,3)=SA(I+1,1)-SA(I,1)    STA1740
LA=1                           STA1750

```

```

DO 210 I=1,IXN                      STA1760
K=ISA(I)                            STA1770
SA(K,1)=LA                          STA1780
210 LA=LA+1                           STA1790
K=0                                 STA1800
RNS=0.                               STA1810
RNSS=ONE                            STA1820
LR=0                                STA1830
DO 250 I=1,IXNM1                     STA1840
IF (SA(I,3).NE.0.0.AND.K.EQ.0) GO TO 240 STA1850
IF (SA(I,3).NE.0.0) GO TO 220        STA1860
RNS=RNS+RNSS                         STA1870
K=K+1                               STA1880
GO TO 250                            STA1890
220 K=K+1                            STA1900
RNS=RNS+RNSS                         STA1910
RNS=RNS/FLOAT(K)                     STA1920
DO 230 L=1,K                          STA1930
LR=LR+1                             STA1940
LRR=ISA(LR)                          STA1950
230 SA(LRR,1)=RNS                     STA1960
LR=LR-1                            STA1970
RNS=0.                               STA1980
K=0                                 STA1990
240 LR=LR+1                           STA2000
250 RNSS=RNSS+ONE                     STA2010
ICI=0                               STA2020
IPLUS=0                             STA2030
IMINUS=0                            STA2040
IDRUNS=0                            STA2050
IC=0                                STA2060
ADEV=0.0                            STA2070
DEV3=0.0                            STA2080
DEV2=0.0                            STA2090
DEV=0.0                             STA2100
DEVI=0.0                            STA2110
DEVWT=0.                            STA2120
DEV4=0.0                            STA2130
AK=1.                               STA2140
KWT=IARGS(2)                         STA2150
NRXX=KWT+NRMAX-1                    STA2160
TA=1.0                             STA2170
DO 320 I=1,IXN                       STA2180
T=SA(I,2)-A(4)                      STA2190
SA(I,3)=T                           STA2200
DEV=T+DEV                           STA2210
ADEV=ADEV+ABS(T)                     STA2220
DEV2=DEV2+T**2                        STA2230
DEV3=DEV3+T**3                        STA2240
DEV4=DEV4+T**4                        STA2250
DEVI=AK*T+DEVI                       STA2260
AK=AK+1.0                           STA2270
IF (IWT.EQ.1) GO TO 280             STA2280
260 IF (RC(KWT).NE.0.) GO TO 270             STA2290
IF (KWT.GE.NRXX) GO TO 290           STA2300
KWT=KWT+1                           STA2310
GO TO 260                            STA2320
270 TA=RC(KWT)                         STA2330
280 DEVWT=DEVWT+TA*T**2              STA2340

```

```

290 IF (T.LT.0.0) GO TO 300 STA2350
IPLUS=IPLUS+1 STA2360
ICI=+1 STA2370
GO TO 310 STA2380
300 IMINUS=IMINUS+1 STA2390
ICI=-1 STA2400
310 IF (IC.EQ.ICI) GO TO 320 STA2410
IC=ICI STA2420
IDRUNS=IDRUNS+1 STA2430
320 KWT=KWT+1 STA2440
A(13)=DEVWT/(A(2)-1.) STA2450
A(9)=FSQRT(A(13)) STA2460
A(10)=A(9)/FSQRT(WT) STA2470
A(14)=100.*A(9)/A(4) STA2480
A(28)=IPLUS STA2490
A(29)=IMINUS STA2500
A(31)=1.+(2.*A(28)*A(29)/A(2)) STA2510
A(32)=FSQRT((2.*A(28)*A(29)*(2.*A(28)*A(29)-A(28)-A(29)))/((A(28)+STA2520
1A(29))**2*(A(2)-1.))) STA2530
A(36)=(DEV3/A(2))**2/((A(2)-1.)/A(2)*A(13))**3 STA2540
A(37)=(DEV4/A(2))/((A(2)-1.)/A(2)*A(13))**2 STA2550
A(38)=SUMWT STA2560
A(40)=DEVWT STA2570
A(30)=IDRUNS STA2580
A(33)=(A(30)-A(31))/A(32) STA2590
A(19)=12.*DEVI/(A(2)*(A(2)**2-1.)) STA2600
A(20)=FSQRT((1./(A(2)-2.))*(12.*DEV2/(A(2)*(A(2)**2-1.))-A(19)**2))STA2610
1) STA2620
A(21)=A(19)/A(20) STA2630
CALL PROB (ONE,A(2)-ONE,A(21)*A(21),A(22)) STA2640
DIF=0 STA2650
IRUN=1 STA2660
DO 325 I=2,IXN STA2665
TA=SA(I,2)-SA(I-1,2) STA2670
IF(TA) 326,325,326 STA2675
325 CONTINUE STA2680
326 DO 330 I=2,IXN STA2685
T=SA(I,2)-SA(I-1,2) STA2690
DIF=DIF+T**2 STA2700
IF (TA*T.GE.0.0) GO TO 330 STA2710
TA=T STA2720
IRUN=IRUN+1 STA2730
330 CONTINUE STA2740
A(23)=IRUN STA2750
A(26)=DIF/(A(2)-1.) STA2760
A(27)=A(26)/A(13) STA2770
A(41)=A(4)*FSQRT(WT)/A(9) STA2780
A(12)=ADEV/A(2) STA2790
NU=NZW-1 STA2800
VNU=NU STA2810
T=ZERO STA2820
TK1=ZERO STA2830
TK2=ZERO STA2840
IF (NU.GE.5) GO TO 350 STA2850
DO 340 I=1,4 STA2860
V=I/NU STA2870
T=T+BCON(I)*V STA2880
TK2=BKCON(I)*V+TK2 STA2890
340 TK1=TK1+AKCON(I)*V STA2900

```

```

      GO TO 400                               STA2910
350   T=(((AT5(6)/VNU+AT5(5))/VNU+AT5(4))/VNU+AT5(3))/VNU+AT5(2)) /VNU+ASTA2920
      1T5(1)                                 STA2930
      IF (NU.GT.10) GO TO 370                 STA2940
      DO 360 I=1,6                           STA2950
      V=(I+4)/NU                            STA2960
      TK1=TK1+CK1(I)*V                      STA2970
360   TK2=TK2+DK2(I)*V                      STA2980
      GO TO 400                             STA2990
370   IF (NU.GT.100) GO TO 390                STA3000
      DO 380 I=1,7                           STA3010
      V=VNU** (I-4)                          STA3020
      TK1=TK1+XK1(I)*V                      STA3030
380   TK2=TK2+YK2(I)*V                      STA3040
      GO TO 400                             STA3050
390   V2=FSQRT(TWO*VNU)                     STA3060
      V2M1=FSQRT(TWO*VNU-ONE)                STA3070
      TK1=V2 / (CONK+V2M1)                   STA3080
      TK2=V2 / (-CONK+V2M1)                  STA3090
400   A(15)=A(4)-T*A(10)                   STA3100
      A(16)=A(4)+T*A(10)                   STA3110
      A(17)=TK1*A(9)                        STA3120
      A(18)=TK2*A(9)                        STA3130
C     START PRINT OUT                      STA3140
      IF (L2.EQ.2) GO TO 560                 STA3150
      CALL PAGE (4)                         STA3160
      IF (IWT.EQ.2) GO TO 410                 STA3170
      WRITE (IPRINT,640) NXCOL,NZW           STA3180
      GO TO 440                             STA3190
410   IF (NZW.NE.NRMAX) GO TO 420          STA3200
      WRITE (IPRINT,650) NXCOL,NXWT,NZW       STA3210
      GO TO 430                             STA3220
420   WRITE (IPRINT,660) NXCOL,NXWT,NZW,NRMAX STA3230
430   WRITE (IPRINT,670)                     STA3240
440   DO 450 I=1,10                         STA3250
450   IB(I)=A(I+50)                        STA3260
      WRITE (IPRINT,680) (IB(I),I=1,10)        STA3270
      WRITE (IPRINT,690)                     STA3280
      WRITE (IPRINT,700) (A(I+2),A(I+8),I=1,6) STA3290
      WRITE (IPRINT,710) (A(I),I=15,18)        STA3300
      WRITE (IPRINT,720) (A(I),A(I+15),I=19,22),(A(I),I=38,41) STA3310
      IB(1)=A(23)                           STA3320
      IB(2)=A(28)                           STA3330
      IB(3)=A(29)                           STA3340
      IB(4)=A(30)                           STA3350
      WRITE (IPRINT,730) IB(1),A(42),A(24),A(43),(A(I),I=25,27),(IB(I),I=28,30) STA3360
      I=2,4),(A(I),I=31,33)                 STA3370
      WRITE (IPRINT,740)                     STA3380
      KB=ISA(1)                            STA3390
      T=SA(KB,2)                           STA3400
      LINEP=40                            STA3410
      LINE=0                                STA3420
      LW=IARGS(2)                           STA3430
      DO 520 I=1,IXNM1                      STA3440
      IF (LINEP.LT.40) GO TO 470             STA3450
      LINEP=0                                STA3460
      CALL PAGE (4)                         STA3470
      WRITE (IPRINT,750)                     STA3480
      IF (IWT.EQ.1) GO TO 460                 STA3490

```

```

      WRITE (IPRINT,760)
      GO TO 470
460      WRITE (IPRINT,770)
470      K=ISA(I+1)
         TA=SA(K,2)-T
         GO TO (500,480), IWT
480      IF (RC(LW).NE.0.0) GO TO 490
         LW=LW+1
         GO TO 480
490      WRITE (IPRINT,790) I,SA(I,2),SA(I,1),SA(I,3),RC(LW),ISA(I),T,TA
         LW=LW+1
         GO TO 510
500      WRITE (IPRINT,780) I,SA(I,2),SA(I,1),SA(I,3),ISA(I),T,TA
510      T=SA(K,2)
         LINE=LINE+1
         IF (LINE.NE.10) GO TO 520
         LINE=0
         LINEP=LINEP+10
         WRITE (6,800)
520      CONTINUE
         IF (IWT.EQ.1) GO TO 550
530      IF (RC(LW).NE.0.0) GO TO 540
         LW=LW+1
         GO TO 530
540      WRITE (IPRINT,790) NZW,SA(NZW,2),SA(NZW,1),SA(NZW,3),RC(LW),ISA(NZ
         1W),T
         GO TO 560
550      WRITE (IPRINT,780) NZW,SA(NZW,2),SA(NZW,1),SA(NZW,3),ISA(NZW),T
560      IF (ISTORE.EQ.2) RETURN
         IF (NARGS.EQ.2.OR.NARGS.EQ.3) GO TO 570
         L=IARGS(NARGS-3)
         M=IARGS(NARGS-2)
         K=IARGS(NARGS-1)
         J=IARGS(NARGS)
         GO TO 580
570      L=IARGS(NARGS)
         M=L+NROW
         K=M+NROW
         J=K+NROW
580      DO 590 I=1,NZW
         MB=ISA(I)
         RC(K)=SA(MB,2)
         RC(M)=SA(I,1)
         RC(J)=SA(I,3)
         M=M+1
         K=K+1
590      J=J+1
         IF (NZW.EQ.NRMAX) GO TO 610
         NZW1=NZW+1
         DO 600 I=NZW1,NRMAX
         RC(M)=0.
         RC(K)=0.
         RC(J)=0.
         M=M+1
         K=K+1
600      J=J+1
610      NTOP=60
         IF (NROW.LT.NTOP) NTOP=NROW
         DO 620 I=1,NTOP

```

```

        RC(L)=A(I)                      STA4090
620      L=L+1                         STA4100
        IF (NRMAX.LT.60) RETURN          STA4110
        DO 630 I=61,NRMAX              STA4120
        RC(L)=0.                         STA4130
630      L=L+1                         STA4140
        RETURN                          STA4150
C
640      FORMAT (1H0,4X,28HSTATISTICAL ANALYSIS OF COL ,I4,33X,4HN = ,I4) STA4170
650      FORMAT (1H0,4X,28HSTATISTICAL ANALYSIS OF COL ,I4,8X,15HWEIGHTS INSTA4180
1 COL ,I4,6X,4HN = ,I4)             STA4190
660      FORMAT (1H0,4X,28HSTATISTICAL ANALYSIS OF COL ,I4,8X,15HWEIGHTS INSTA4200
1 COL ,I4,6X,4HN = ,I4,33H(No of Non-Zero Wts) COL LENGTH =,I4) STA4210
670      FORMAT (1H0,24X,64HALL COMPUTATIONS ARE BASED ON OBSERVATIONS WITHSTA4220
1 NON-ZERO WEIGHTS)                STA4230
680      FORMAT (1H0/15X,28HFREQUENCY DISTRIBUTION (1-6),7X,10I6)           STA4240
690      FORMAT (1H0/5X,26HMEASURES OF LOCATION (2-2),34X,28HMEASURES OF DISTA4250
1SPERSION (2-6))                  STA4260
700      FORMAT(1H0,
1 9X,26HUNWEIGHTED MEAN          =, 1PE15.7,20X,                         STA4270
2 26HSTANDARD DEVIATION          =, E15.7    /                         STA4275
3 10X,26HWEIGHTED MEAN          =, E15.7,20X,                         STA4280
4 26HS.D. OF MEAN               =, E15.7    /                         STA4290
5 10X,26HMEDIAN                 =, E15.7,20X,                         STA4295
6 26HRANGE                      =, E15.7    /                         STA4300
7 10X,26HMID-RANGE              =, E15.7,20X,                         STA4305
8 26HMEAN DEVIATION             =, E15.7    /                         STA4310
9 10X,26H25 PCT UNWTD TRIMMED MEAN=, E15.7,20X,                         STA4315
A 26HVARIANC                   =, E15.7    /                         STA4320
B 10X,26H25 PCT WTD TRIMMED MEAN =, E15.7,20X,                         STA4325
C 26HCOEFFICIENT OF VARIATION =, E15.7    )                         STA4330
710      FORMAT (1H0//20X,50HA TWO-SIDED 95 PCT CONFIDENCE INTERVAL FOR MEASTA4350
1N IS1PE11.4,3H TO,E11.4,6H (2-2)/20X,50HA TWO-SIDED 95 PCT CONFIDE STA4360
2NCE INTERVAL FOR S.D. IS,E11.4,3H TO,E11.4,6H (2-7))                 STA4370
720      FORMAT (1H0//5X,30HLINEAR TREND STATISTICS (5-1) ,30X,16HOTHER STASTA4380
1TISTICS//10X,5HSLOPE,20X,1H=,1PE15.7,20X,7HMINIMUM,18X,1H=,E15.7/1STA4390
20X,13HS.D. OF SLOPE,12X,1H=,E15.7,20X,7HMAXIMUM,18X,1H=,E15.7/10X,STA4400
326HSLOPE/S.D. OF SLOPE = T =,E15.7,20X,8HBETA ONE,17X,1H=,E15.7/1STA4410
40X,35HPROB EXCEEDING ABS VALUE OF OBS T =,OPF6.3,20X,8HBETA TWO,17STA4420
5X,1H=,1PE15.7/71X,17HWTD SUM OF VALUES,8X,1H=,E15.7/71X,18HWTD SUMSTA4430
6 OF SQUARES,7X,1H=,E15.7/5X,24HTESTS FOR NON-RANDOMNESS,42X,26HWTDSTA4440
7 SUM OF DEVS SQUARED =,E15.7/71X,11HSTUDENT,S T,14X,1H=,E15.7) STA4450
730      FORMAT (10X,26HNO OF RUNS UP AND DOWN =,I5,30X,26HWTD SUM ABSOLUSTA4460
1TE VALUES =,1PE15.7/10X,26HEXPECTED NO OF RUNS =,OPF7.1,28X,STA4470
226HWTD AVE ABSOLUTE VALUES =,1PE15.7/10X,26HS.D. OF NO OF RUNS STA4480
3 =,OPF8.2/10X,26HMEAN SQ SUCCESSIVE DIFF =,1PE16.7/10X,26HMEANSTA4490
4 SQ SUCC DIFF/VAR =,OPF9.3//10X,24HDEVIATIONS FROM WTD MEAN//1STA4500
55X,21HNO OF + SIGNS =,I5/15X,21HNO OF - SIGNS =I5/15X,STA4510
610HNO OF RUNS,10X,1H=,I5/15X,21HEXPECTED NO OF RUNS =,F7.1/15X,12HSTA4520
7S.D. OF RUNS,8X,1H=,F8.2/15X,21HDIFF./S.D. OF RUNS =F9.3) STA4530
740      FORMAT (////68H NOTE - ITEMS IN PARENTHESES REFER TO PAGE NUMBER STA4540
1IN NBS HANDBOOK 91)                      STA4550
750      FORMAT (//27X,12HOBSERVATIONS,47X,20HORDERED OBSERVATIONS) STA4560
760      FORMAT (1H0,8X,1HI,9X,4HX(I),9X,4HRANK,7X,9HX(I)-MEAN,7X,4HW(I),16STA4570
1X,3HNO.,8X,4HX(J),10X,11HX(J+1)-X(J)) STA4580
770      FORMAT (1H0,8X,1HI,9X,4HX(I),9X,4HRANK,7X,9HX(I)-MEAN,27X,3HNO.,8XSTA4590
1,4HX(J),10X,11HX(J+1)-X(J)) STA4600
780      FORMAT (I10,1PE17.7,OPF9.1,1PE17.7,22X,I6,1P2E17.7) STA4610
790      FORMAT (I10,1PE17.7,OPF9.1,1PE17.7,1PE12.3,10X,I6,1P2E17.7) STA4620
800      FORMAT (1H )                           STA4630
        END                                STA4640

```

```

SUBROUTINE STM (NSTMT) STM 10
C VERSION 5.00 STM 20
COMMON /BLOCKA/ MODE,M,KARD(83),KARG,ARG,ARG2,NEWCD(80),KRDEND STM 30
C STM 40
C THIS SUBROUTINE ASSEMBLES AND CHECKS A STATEMENT NUMBER. STM 50
C STM 60
C CALLED BY.. .MAIN. STM 70
MISC=10*KARD(M) STM 80
10 M=M+1 STM 90
K=KARD(M) STM 100
IF (K.GE.10) GO TO 30 STM 110
MISC=10*(MISC+K) STM 120
IF (MISC.LT.10000) GO TO 10 STM 130
C STM 140
C ILLEGAL STATEMENT NUMBER EXIT STM 150
C STM 160
20 KARG=1 STM 170
RETURN STM 180
C STM 190
C NON-NUMERIC FOUND, IS IT A . STM 200
C STM 210
30 IF (K.EQ.37) GO TO 50 STM 220
C STM 230
C IS IT A / STM 240
C STM 250
40 IF (K.EQ.36) GO TO 70 STM 260
C STM 270
C IS IT A SPACE STM 280
C STM 290
IF (K-44) 20,60,20 STM 300
C STM 310
C . FOUND, MUST BE FOLLOWED BY ONE AND ONLY ONE NUMERAL STM 320
C STM 330
50 M=M+1 STM 340
K=KARD(M) STM 350
IF (K.GE.10) GO TO 20 STM 360
MISC=MISC+K STM 370
60 M=M+1 STM 380
K=KARD(M) STM 390
GO TO 40 STM 400
70 M=M+1 STM 410
K=KARD(M) STM 420
C STM 430
C / FOUND, MUST BE FOLLOWED BY BLANKS THEN/OR A LETTER STM 440
C STM 450
IF (K.EQ.44) GO TO 70 STM 460
IF (K.GE.36.OR.K.LT.10) GO TO 20 STM 470
C STM 480
C LEGAL STATEMENT NUMBER FOUND STM 490
C STM 500
NSTMT=MISC STM 510
KARG=0 STM 520
RETURN STM 530
END STM 540

```

```

SUBROUTINE STORE (J) STO 10
C VERSION 5.00 STORE 5/15/70 STO 20
COMMON /BLOCKA/ MODE,M,KARD(83),KARG,ARG,ARG2,NEWCD(80),KRDEND STO 30
COMMON /BLOCKB/ NSTMT,NSTMTH,NCOM,LCOM,IOVFL,COM(2000) STO 40
COMMON /BLOCRC/ NRC,RC(12600) STO 50
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NSTO 60
1ARGS,VWXYZ(8),NERROR STO 70
DIMENSION ARGS(100) STO 80
EQUIVALENCE (ARGS(1),RC(12501)) STO 90
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG STO 100
C STORAGE LAYOUT.. STATEMENT NUMBER STO 110
C NUMBER OF WORDS IN ENTRY STO 120
C NARGS+64*(L1+64*L2) STO 130
C ALL ITEMS ARE STORED IN ( ENTRY 1 ) STO 140
C FLOATING POINT TO ALLOW ( 2 ) STO 150
C CONVERSION TO DOUBLE- ..... STO 160
C PRECISION. ( LAST WORD ) STO 170
C STO 180
IF (IOVFL.NE.0) RETURN STO 190
IZE=J+2 STO 200
IF (NSTMT.GT.NSTMTH) GO TO 90 STO 210
C STATEMENT IS AN INSERTION OR A REPLACEMENT STO 220
C STO 230
C STO 240
L=LOCATE(NSTMT) STO 250
IF (L.GT.0) GO TO 30 STO 260
C STO 270
L=-L STO 280
IDIF=IZE STO 290
10 LL=NCOM STO 300
C STATEMENT IS AN INSERTION, OPEN GAP STO 310
II=LL+IDIF STO 320
IF (II.GE.LCOM) GO TO 100 STO 330
DO 20 I=L,NCOM STO 340
COM(II)=COM(LL) STO 350
II=II-1 STO 360
20 LL=LL-1 STO 370
GO TO 60 STO 380
C STO 390
C STATEMENT IS REPLACEMENT STO 400
C STO 410
30 IDIF=IZE-IFIX(COM(L+1)) STO 420
IF (IDIF) 40,60,10 STO 430
C STO 440
C NEW STATEMENT SMALLER THAN OLD, CLOSE UP GAP. STO 450
C STO 460
40 I=L-IDIF STO 470
II=L STO 480
DO 50 IA=I,NCOM STO 490
COM(II)=COM(IA) STO 500
50 II=II+1 STO 510
C STO 520
C INSERT STATEMENT STO 530
C STO 540
60 COM(L)=NSTMT STO 550
COM(L+1)=IZE STO 560
COM(L+2)=NARGS+64*(L1+64*L2) STO 570
NCOM=NCOM+IDIF STO 580
IF (IZE.EQ.3) GO TO 80 STO 590

```

	DO 70 I=4,IZE	STO 600
	COM(L+3)=ARGTAB(I-3)	STO 610
70	L=L+1	STO 620
80	CONTINUE	STO 630
	RETURN	STO 640
C		STO 650
C	PUT STATEMENT ON END	STO 660
C		STO 670
90	L=NCOM	STO 680
	IDIF=IZE	STO 690
	NSTMTH=NSTMTH	STO 700
	NSTMTH=NSTMTH	STO 710
	IF (NCOM+IDIF.LT.LCOM) GO TO 60	STO 720
C		STO 730
C	COM STORAGE OVERFLOW	STO 740
C		STO 750
100	IOVFL=1	STO 760
	CALL ERROR (12)	STO 770
	RETURN	STO 780
	END	STO 790

C	SUBROUTINE STORMT (C,N,NP,K,A)	STT 10
C	VERSION 5.00 STORMT 5/15/70	STT 20
C	*	STT 40
C	SUBROUTINE STORES MATRIX C(NP,K ) FROM SCRATCH AREA A	STT 50
C	*	STT 60
	DIMENSION A(1), C(N,1)	STT 70
	IS=1	STT 80
	DO 10 J=1,K	STT 90
	DO 10 I=1,NP	STT 100
	C(I,J)=A(IS)	STT 110
10	IS=IS+1	STT 120
	RETURN	STT 130
	END	STT 140

```

SUBROUTINE STRUVE (Z,A,B,C)                               STR   10
C VERSION 5.00      STRUVE      5/15/70                  STR   20
C DIMENSION C(1)                                         STR   30
C DOUBLE PRECISION Z,A,B,C,X,P,Q,R,S,DBEJ               STR   40
C X=DABS(Z)                                              STR   50
C IF (X.GT..0D0) GO TO 10                                STR   60
C A=.0D0                                                 STR   70
C B=.0D0                                                 STR   80
C GO TO 40                                               STR   90
10  IF (X.GT.70.D0) GO TO 30                                STR 100
    CALL BEJN (0,C,X)                                       STR 110
    P=.0D0                                                 STR 120
    Q=.0D0                                                 STR 130
    DO 20 N=1,49                                         STR 140
    J=2*N                                                 STR 150
    K=J+1                                                 STR 160
    R=J-1                                                 STR 170
    S=4*N**2-1                                           STR 180
    P=P+C(J)/R                                         STR 190
20  Q=Q+C(K)/S                                         STR 200
    A=P/.78539816339D0                                    STR 210
    B=(2.D0*Q+1.D0-C(1))/1.5707963268D0                STR 220
    GO TO 40                                              STR 230
30  S=1.D0/X**2                                         STR 240
    P=1.D0-S*(1.D0-9.D0*S*(1.D0-25.D0*S*(1.D0-49.D0*S))) STR 250
    A=DBEJ(X,0,5)+P/(X*1.5707963268D0)                 STR 260
    Q=1.D0+S*(1.D0-3.D0*S*(1.D0-15.D0*S*(1.D0-35.D0*S))) STR 270
    B=DBEJ(X,1,5)+Q/(1.5707963268D0)                   STR 280
40  RETURN                                              STR 290
    END                                                 STR 300

```

```

SUBROUTINE SYMV (A,NROW,N,K)                           SYM   10
C VERSION 5.00      SYMV      5/15/70                  SYM   20
C FOR OMNITAB WRITTEN BY S PEAVY 11/29/67             SYM   30
C A-FIRST ELEMENT OF MATRIX A                         SYM   40
C NROW -NO. OF ROWS IN A AS DEFINED IN A DIMENSION ST SYM   50
C N -PRESENT SIZE OF A                            SYM   60
C K -STATUS FOR SYMMETRY                          SYM   70
C K=0 EXACT SYMMETRY A A(I,J)/A(J,I)=1           SYM   80
C K=1 SYMM TO A RELATIVE RROR ABS(1-A(I,J)/A(J,I))= OR LESS 1.E-7 SYM   90
C K=2 NO SYMMETRY                                 SYM 100
C
DIMENSION A(NROW,NROW)                                SYM 110
K=0                                                 SYM 120
NN=N-1                                              SYM 130
DO 40 J=1,NN                                         SYM 140
I=J+1                                              SYM 150
DO 40 L=I,N                                         SYM 160
IF (A(J,L)) 20,10,20                                SYM 170
10  T=ABS(A(L,J))                                     SYM 180
    GO TO 30                                         SYM 190
20  T=ABS(1.0-A(L,J)/A(J,L))                        SYM 200
30  IF (T.EQ.0.) GO TO 40                            SYM 210
    K=1                                              SYM 220
    IF (T.LE.1.E-7) GO TO 40                         SYM 230
    K=2                                              SYM 240
    RETURN                                         SYM 250
40  CONTINUE                                         SYM 260
    RETURN                                         SYM 270
    END                                              SYM 280

```

```

SUBROUTINE TAPOP                                     TAP 10
C   VERSION 5.00      TAPOP      5/15/70          TAP 20
C   RV SUBROUTINE USED WITH TAPE COMMANDS          TAP 30
COMMON /TAPE/ NAME4(2),NTPCT,IPUNCP,INUNIP,L1TP    TAP 40
COMMON /CODE/ IALPH(6),NALPH(5),ID(9,3),IR(300,4),IRD(30,6) TAP 50
COMMON/ABCDEF/L(48)                                TAP 55
COMMON /BLOCKA/ MODE,M,KARD(83),KARG,ARG,ARG2,NEWCD(80),KRDEND TAP 60
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG               TAP 70
COMMON /CODETP/ ITP(10,4)                            TAP 80
NAME4(1)=0                                         TAP 90
NAME4(2)=0                                         TAP 100
NTPCT=0                                           TAP 110
10 K=KARD(M)                                       TAP 120
C   THE TAPE ID MAY BE A NUMBER OR A LETTER          TAP 130
IF (K.LT.36) IF(K-10) 20,30,30                      TAP 135
M=M+1                                             TAP 140
GO TO 10                                         TAP 150
20 ITAPE=KARD(M)+6                                TAP 160
MP=M-1                                           TAP 170
GO TO 50                                         TAP 180
30 MP=M-1                                         TAP 190
CALL NNAME (NAME4(1))                           TAP 200
ITAPE=0                                         TAP 210
DO 40 I=1,6                                      TAP 220
IF (NAME4(1).NE.IALPH(I)) GO TO 40              TAP 230
ITAPE=I+6                                         TAP 240
GO TO 50                                         TAP 250
40 CONTINUE                                       TAP 260
50 IF(ITAPE.GT.9) GO TO 60                      TAP 270
KARD(MP)=L(45)                                    TAP 280
KARD(MP+1)=ITAPE                                 TAP 290
GO TO 80                                         TAP 300
60 KARD(MP)=1                                     TAP 310
KARD(MP+1)=ITAPE-10                            TAP 320
C   CREAD READ WRITE                            TAP 340
80 IF (NAME(1).NE.ITP(2,1).AND.NAME(1).NE.ITP(1,1).AND.NAME(1).NE.ITPTAP 350
1(3,1)) GO TO 110                               TAP 360
NAME4(1)=0                                         TAP 370
NAME4(2)=0                                         TAP 380
M=M+1                                             TAP 390
90 K=KARD(M)                                       TAP 400
IF (K.LT.36) IF (K-10) 110,100,100             TAP 410
M=M+1                                             TAP 420
GO TO 90                                         TAP 430
100 CALL NNAME (NAME4(1))                         TAP 440
110 M=MP                                           TAP 450
RETURN                                         TAP 460
END                                            TAP 470

```

```

SUBROUTINE TAPOP2                                     TP2  10
C VERSION 5.00      TAPOP2      5/15/70             TP2  20
C THIS SUBROUTINE IS NEEDED ONLY FOR TAPE OPERATIONS   TP2  30
C L1= 45  L2=1,7                                      TP2  40
C     READ TAPE A-F A-F (FORMAT) INTO COLUMNS ++,++,++,++, ETC.  TP2  50
C     READ UNTIL A RECORD OF A ZEROS ARE ENCOUNTERED       TP2  60
C     IF NO FORMAT IS GIVEN, CARDS ARE READ AS IN READ COMMAND  TP2  70
C L1= 46  L2=1,7                                      TP2  80
C     CREAD TAPE A-F A-F (FORMAT) , , CARDS INTO COLUMNS ++,++,++,+, ETP2  90
C     CREAD TAPE A-F , ,CARDS INTO COLUMNS ++,++,++,ETC.        TP2 100
C     READ USING A COUNTER                                TP2 110
C L1= 47  L2=1,7                                      TP2 120
C     WRITE TAPE A-F A-F(FORMAT) FROM COLUMNS ++,++,++,++,ETC.  TP2 130
C A RECORD OF ZEROS IS WRITTEN AFTER NRMAX VALUES      TP2 140
C L1= 48  L2=1                                      TP2 150
C     SET TAPE A-F INTO COLUMNS ++
C     SET TAPE A-F INTO ROW , , OF COLUMN ++
C READ UNTIL A RECORD OF ZEROS IS ENCOUNTERED          TP2 180
C L1= 49  L2=1                                      TP2 190
C     CSET TAPE A-F , , VALUES INTO COLUMN ++
C     CSET TAPE A-F , , VALUES INTO ROW , , OF COLUMN ++
C     READ USING A COUNTER                                TP2 220
C L1= 50  L2=1                                      TP2 230
C     ENDFILE TAPE A-F                                 TP2 240
C L1=50  L2=2                                      TP2 250
C     REWIND TAPE A-F                                 TP2 260
C L1= 50  L2=3                                      TP2 270
C     SKIP TAPE A-F FORWARD , , RECORDS               TP2 280
C L1= 50  L2=4                                      TP2 290
C     BACKSPACE TAPE A-F , , RECORDS                 TP2 300
COMMON /BLOCKC/ KIO,INUNIT,ISCRAT,KBDOUT,KRDKNT,LLIST    TP2 310
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NTP2 320
1ARGS,VWXYZ(8),NERROR                                    TP2 330
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG                  TP2 340
COMMON/HEADER/NOCARD(80),ITLE(60,6),LNCNT,IPRINT,NPAGE,IPUNCH  TP2 350
COMMON /SCRAT/ NS,NS2,A(13500)                         TP2 370
COMMON /TAPE/ NAME4(2),NTPCT,IPUNCP,INUNIP,L1TP        TP2 380
C CHECK FOR CORRECT NUMBER OF ARGUMENTS                TP2 390
IF(L1.LT.50) IF(NARGS-2) 5,30,30                      TP2 395
GO TO (10,10,20,20), L2                                TP2 400
5 CALL ERROR (10)                                       TP2 410
GO TO 30                                              TP2 420
10 IF (NARGS.NE.1) CALL ERROR (10)                     TP2 430
GO TO 40                                              TP2 440
20 IF (NARGS.NE.2) CALL ERROR (10)                     TP2 450
C ALL ARGUMENTS SHOULD BE INTEGERS                   TP2 460
30 J=NARGS                                         TP2 470
CALL CKIND (J)                                         TP2 480
IF (J.NE.0) CALL ERROR (3)                           TP2 490
C IS TAPE NUMBER CORRECT                            TP2 500
40 IF (IARGS(1).LT.7.OR.IARGS(1).GT.12) CALL ERROR (28)  TP2 510
IF (NERROR.NE.0) RETURN                               TP2 520
L1P=L1-44                                         TP2 530
GO TO (50,70,90,60,80,170), L1P                      TP2 540
50 IF (L2.EQ.1) GO TO 60                            TP2 550
INUNIT=IARGS(1)                                       TP2 560
IARGS(1)=0                                           TP2 570
GO TO 130                                            TP2 580
60 IA=2                                             TP2 590

```

```

GO TO 100                                     TP2 600
70  IF (L2.NE.1) GO TO 60                      TP2 610
80  NTPCT=IARGS(2)                            TP2 620
     IA=3                                      TP2 630
     GO TO 100                                 TP2 640
90  IA=2                                       TP2 650
     IPUNCH=IARGS(1)                           TP2 660
     GO TO 110                                 TP2 670
100 INUNIT=IARGS(1)                           TP2 680
110 I=1                                         TP2 690
     DO 120 II=IA,NARGS                      TP2 700
     IARGS(I)=IARGS(II)                         TP2 710
120  I=I+1                                     TP2 720
     NARGS=NARGS-IA+1                          TP2 730
130  L1TP=L1                                    TP2 740
     GO TO (140,140,150,160,160), L1P        TP2 750
140  CALL READX                                TP2 760
     RETURN                                     TP2 770
150  CALL PUNCH                                TP2 780
     RETURN                                     TP2 790
160  CALL SET                                   TP2 800
     RETURN                                     TP2 810
170  ITPP=IARGS(1)                            TP2 820
     GO TO (180,190,200,220), L2              TP2 830
180  ENDFILE ITPP                             TP2 840
     RETURN                                     TP2 850
190  REWIND ITPP                              TP2 860
     RETURN                                     TP2 870
200  IREC=IARGS(2)                            TP2 880
     DO 210 I=1,IREC                          TP2 890
     READ (ITPP,240) A(1)                      TP2 900
210  CONTINUE                                  TP2 910
     RETURN                                     TP2 920
220  IREC=IARGS(2)                            TP2 930
     DO 230 I=1,IREC                          TP2 940
     BACKSPACE ITPP                           TP2 950
230  CONTINUE                                  TP2 960
     RETURN                                     TP2 970
C
240  FORMAT (80A1)                            TP2 990
     END                                         TP21000

```

C SUBROUTINE THERMO THE 10  
 C VERSION 5.00 THERMO 5/15/70 THE 20  
 C IT NOW CONTAINS THE COMMANDS CTOF, FTOC, ATOMIC, MOLWT, EINSTEIN, THE 30  
 C PFTRANSLATIONAL, PFATOMIC, AND PARTFUNCTION THE 40  
 C THERMODYNAMIC PACKAGE WRITTEN BY R. MCCLENON, NSRDS-NBS, NOV. 69 THE 50  
 C THE 60  
 C THE VALUES FOR L2 ARE -- THE 70  
 C 1 -CTOF (CENTIGRADE TO FAHRENHEIT) THE 80  
 C 2 - FTOC (FAHRENHEIT TO CENTIGRADE) THE 90  
 C 3 - ATOMIC MASS TABLE THE 100  
 C 4 - MOLWT (MOLECULAR WEIGHT) THE 110  
 C 5 - EINSTEIN FUNCTION THE 120  
 C 6 - PFTRANS (PARTITION FUNCTION TRANSLATIONAL) THE 130  
 C 7 - PFATOM (P.F. ATOMIC) THE 140  
 C 8 - PARTFUNCTION THE 150  
 C 9 - BOLDISTRIBUTION (BOLZMAN DISTRIBUTION) THE 160  
 C THE 170  
 C WRITTEN BY R. MCCLENON, NSRDS-NBS, DEC. 1969 THE 180  
 C THE 190  
 C COMMAND FORMATS ARE AS FOLLOWS -- THE 200  
 C CTOF OF \$\$ STORE IN COL ++ THE 210  
 C FTOC OF \$\$ STORE IN COL ++ THE 220  
 C ATOMIC MASSES STORE IN COL ++ THE 230  
 C MOLWT Z=,, AMOUNT=,, Z=,, AMOUNT=,, ... STORE SUM IN COL ++ THE 240  
 C EINSTEIN TEMP IN \$\$ VIB FREQ IN WAVE NO IN \$\$ START STORING IN ++ THE 250  
 C OR EINSTEIN TEMP IN \$\$ FREQ IN \$\$ GAS CONST R=,, START IN ++ THE 260  
 C PFTRANS TEMP IN \$\$ MOL WT M IN \$\$ START STORING IN \$\$ THE 270  
 C PFATOM TEMP IN \$\$ MOL WT M IN \$\$ WAVE NO IN ++ DEGEN G IN ++ THE 280  
 C START STORING IN COL ++ THE 290  
 C PARTFUNC TEMP IN \$\$ WAVE NO IN ++ G IN ++ START STORING IN ++ THE 300  
 C VIBDIST TEMP IN \$\$ WAVE NO IN ++ G IN ++ START STORING IN ++ THE 310  
 C THE 320  
 C SEE HANDBOOK 101 FOR DETAILS ON STORAGE BY ALL COMMANDS EXCEPT THE 330  
 C VIBDIST (WHICH IS NEW) THE 340  
 C VIBDIST STORES THE PERCENTAGE OF MOLECULES IN EACH OF THE THE 350  
 C VIBRATIONAL ENERGY LEVELS. IF THERE ARE N ENERGY LEVELS VIBDIST THE 360  
 C WILL USE N COLUMNS FOR STORAGE THE 370  
 C THE 380  
 C COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NTHE 390  
 1IARGS,VWXYZ(8),NERROR THE 400  
 EQUIVALENCE (ARGS(1),RC(12501)) THE 410  
 COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG THE 420  
 DIMENSION ARGS(100) THE 430  
 COMMON /BLOCRC/ NRC,RC(12600) THE 440  
 DOUBLE PRECISION X,EXX,EXDIF,FDEXP,FDLOG,Q0,Q1,Q2,G,QQ THE 450  
 COMMON /SCRAT/ NS,NS2,A(13500) THE 460  
 C DIMENSION QQ(NS2) THE 470  
 DIMENSION QQ(6750) THE 480  
 EQUIVALENCE (A(1),QQ(1)) THE 490  
 DIMENSION ATWT(103) THE 500  
 DATA ATWT(1),ATWT(2),ATWT(3),ATWT(4),ATWT(5),ATWT(6)/1.00800,4.002THE 510  
 160,6.94100,9.01218,10.81000,12.01100/ THE 520  
 DATA ATWT(7),ATWT(8),ATWT(9),ATWT(10),ATWT(11),ATWT(12)/14.00670,1THE 530  
 15.99940,18.99840,20.17900,22.98980,24.30500/ THE 540  
 DATA ATWT(13),ATWT(14),ATWT(15),ATWT(16),ATWT(17),ATWT(18)/26.9815THE 550  
 10,28.08600,30.97380,32.06000,35.45300,39.48000/ THE 560  
 DATA ATWT(19),ATWT(20),ATWT(21),ATWT(22),ATWT(23),ATWT(24)/39.1020THE 570  
 10,40.08000,44.95590,47.90000,50.94140,51.99600/ THE 580  
 DATA ATWT(25),ATWT(26),ATWT(27),ATWT(28),ATWT(29),ATWT(30)/54.9380THE 590

10,55.84700,58.93320,58.71000,63.54600,65.37000/ THE 600  
 DATA ATWT(31),ATWT(32),ATWT(33),ATWT(34),ATWT(35),ATWT(36)/69.7200THE 610  
 10,72.59000,74.92160,78.96000,79.90400,83.80000/ THE 620  
 DATA ATWT(37),ATWT(38),ATWT(39),ATWT(40),ATWT(41),ATWT(42)/85.4678THE 630  
 10,87.62000,88.90590,91.22000,92.90640,95.94000/ THE 640  
 DATA ATWT(43),ATWT(44),ATWT(45),ATWT(46),ATWT(47),ATWT(48)/98.9062THE 650  
 10,101.07000,102.90550,106.40000,107.86800,112.40000/ THE 660  
 DATA ATWT(49),ATWT(50),ATWT(51),ATWT(52),ATWT(53),ATWT(54)/114.820THE 670  
 100,118.69000,121.75000,127.60000,126.90450,131.30000/ THE 680  
 DATA ATWT(55),ATWT(56),ATWT(57),ATWT(58),ATWT(59),ATWT(60)/132.905THE 690  
 150,137.34000,138.90550,140.12000,140.90770,144.24000/ THE 700  
 DATA ATWT(61),ATWT(62),ATWT(63),ATWT(64),ATWT(65),ATWT(66)/147.000THE 710  
 100,150.40000,151.96000,157.20000,158.92540,162.50000/ THE 720  
 DATA ATWT(67),ATWT(68),ATWT(69),ATWT(70),ATWT(71),ATWT(72)/164.930THE 730  
 130,167.26000,168.93420,173.04000,174.97000,178.49000/ THE 740  
 DATA ATWT(73),ATWT(74),ATWT(75),ATWT(76),ATWT(77),ATWT(78)/180.947THE 750  
 190,183.85000,186.20000,190.20000,192.22000,195.09000/ THE 760  
 DATA ATWT(79),ATWT(80),ATWT(81),ATWT(82),ATWT(83),ATWT(84)/196.966THE 770  
 150,200.59000,204.37000,207.20000,208.98060,210.00000/ THE 780  
 DATA ATWT(85),ATWT(86),ATWT(87),ATWT(88),ATWT(89),ATWT(90)/210.000THE 790  
 100,222.00000,223.00000,226.02540,227.02000,232.03810/ THE 800  
 DATA ATWT(91),ATWT(92),ATWT(93),ATWT(94),ATWT(95),ATWT(96)/231.035THE 810  
 190,238.02900,237.04820,239.00000,243.00000,247.00000/ THE 820  
 DATA ATWT(97),ATWT(98),ATWT(99),ATWT(100),ATWT(101),ATWT(102)/247.THE 830  
 100000,249.00000,254.00000,253.00000,255.00000,257.00000/ THE 840  
 DATA ATWT(103)/255.0/ THE 850  
 GO TO (10,250,260,380,490,590,640,710,760), L2 THE 860  
 C THIS IS CTOF THE 870  
 10 M=1 THE 880  
 20 IF (NARGS-2) 1210,30,1210 THE 890  
 30 CALL ADRESS (2,I2) THE 900  
 40 IF (I2) 1220,1230,40 THE 910  
 40 CALL ADRESS (1,I1) THE 920  
 40 IF (I1) 50,1230,50 THE 930  
 50 IF (NRMAX) 1240,1240,60 THE 940  
 60 IF (NERROR) 1180,70,1180 THE 950  
 70 IF (I1) 160,1230,80 THE 960  
 80 IE=0 THE 970  
 DO 150 J=1,NRMAX THE 980  
 I1=I1+J-1 THE 990  
 I2=I2+J-1 THE1000  
 IF (M-1) 1180,100,90 THE1010  
 90 RC(I1)= (RC(I1)-32.0)/1.8 THE1020  
 IF (RC(I1)+273.15) 110,150,150 THE1030  
 100 IF (RC(I1)+273.15) 110,140,140 THE1040  
 110 IF (IE) 130,120,130 THE1050  
 120 CALL ERROR (230) THE1060  
 IE=1 THE1070  
 130 IF (M-1) 1180,140,150 THE1080  
 140 RC(I1)=(1.8\*RC(I1))+32.0 THE1090  
 150 CONTINUE THE1100  
 GO TO 1180 THE1110  
 160 IF (M-1) 1180,180,170 THE1120  
 170 T=(ARGS(1)-32.)/1.8 THE1130  
 IF (T+273.15) 190,230,230 THE1140  
 180 IF (ARGS(1)+273.15) 190,220,220 THE1150  
 190 IF (IE) 210,200,210 THE1160  
 200 CALL ERROR (230) THE1170  
 IE=1 THE1180

```

210 IF (M-1) 1180,220,230 THE1190
220 T=(1.8*ARGS(1))+32.0 THE1200
230 DO 240 J=1,NRMAX THE1210
   II2=I2+J-1 THE1220
240 RC(II2)=T THE1230
   GO TO 1180 THE1240
C THIS IS FTOC THE1250
250 M=2 THE1260
   GO TO 20 THE1270
C THIS IS ATOMIC WEIGHT THE1280
260 IF (NARGS-1) 1210,280,270 THE1290
270 CALL ERROR (221) THE1300
280 CALL ADDRESS (1,I1) THE1310
   IF (I1) 1220,1230,290 THE1320
290 IF (NROW-92) 300,310,310 THE1330
300 CALL ERROR (226) THE1340
310 IF (NERROR) 1180,320,1180 THE1350
320 IF (NROW-103) 330,340,340 THE1360
330 L=NROW THE1370
   GO TO 350 THE1380
340 L=103 THE1390
350 DO 360 J=1,L THE1400
   II1=I1+J-1 THE1410
360 RC(II1)=ATWT(J) THE1420
   IF (NRMAX-L) 370,1180,1180 THE1430
370 NRMAX=L THE1440
   GO TO 1180 THE1450
C THIS IS MOLWT THE1460
380 I=NARGS THE1470
   CALL CKIND (I) THE1480
   IF (I-1) 390,1220,1220 THE1490
390 N=NARGS/2 THE1500
   IF (NARGS-2*N) 400,1210,400 THE1510
400 IF (NRMAX) 1240,1240,410 THE1520
410 CALL ADDRESS (NARGS,I) THE1530
   IF (I) 1220,1230,420 THE1540
420 WT=0.0 THE1550
   IF (N-1) 1210,430,430 THE1560
430 IF (NERROR) 1180,440,1180 THE1570
440 DO 470 J=2,NARGS,2 THE1580
   K=IARGS(J-1) THE1590
   IF (K-103) 450,450,1250 THE1600
450 IF (K) 1250,1250,460 THE1610
460 WT=WT+ATWT(K)*FLOAT(IARGS(J)) THE1620
470 CONTINUE THE1630
   DO 480 J=1,NRMAX THE1640
   II=I+J-1 THE1650
480 RC(II)=WT THE1660
   GO TO 1180 THE1670
C SPACE RESERVED FOR EINSTEIN, PFTRANS, PFATOMIC, AND PARTFUNCTION THE1680
C THIS IS EINSTEIN THE1690
490 IF (NARGS-5) 500,1210,1210 THE1700
500 IF (NARGS-3) 1210,510,530 THE1710
510 CALL ADDRESS (3,I) THE1720
   IF (I) 1220,1230,520 THE1730
520 R=1.0 THE1740
   II=IARGS(3) THE1750
   GO TO 560 THE1760
530 CALL ADDRESS (4,I) THE1770

```

	IF (I) 1220,1230,540	THE1780
540	IF (KIND(3)-1) 1220,550,1220	THE1790
550	R=ARGS(3)	THE1800
	II=IARGS(4)	THE1810
	IF (R) 1250,1250,560	THE1820
560	CALL ADRESS (2,IFQ)	THE1830
	IF (IFQ) 570,1230,580	THE1840
570	F=ARGS(2)	THE1850
	IF (F) 1250,1250,580	THE1860
580	IF (NCOL-II-7) 1230,800,800	THE1870
590	IF (NARGS-3) 1210,600,1210	THE1880
600	CALL ADRESS (3,I)	THE1890
	IF (I) 1220,1230,610	THE1900
610	II=IARGS(3)	THE1910
	CALL ADRESS (2,IWT)	THE1920
	IF (IWT) 620,1230,630	THE1930
620	WT=ARGS(2)	THE1940
	IF (WT) 1250,1250,630	THE1950
630	IF (NCOL-II-6) 1230,800,800	THE1960
640	IF (NARGS-5) 1210,650,1210	THE1970
650	CALL ADRESS (5,I)	THE1980
	IF (I) 1220,1230,660	THE1990
660	II=IARGS(5)	THE2000
	CALL ADRESS (2,IWT)	THE2010
	IF (IWT) 670,1230,680	THE2020
670	WT=ARGS(2)	THE2030
	IF (WT) 1250,1250,680	THE2040
680	CALL ADRESS (3,IFQ)	THE2050
	IF (IFQ) 1220,1230,690	THE2060
690	CALL ADRESS (4,IG)	THE2070
	IF (IG) 1220,1230,700	THE2080
700	IF (NCOL-II-6) 1230,800,800	THE2090
710	IF (NARGS-4) 1210,720,1210	THE2100
720	CALL ADRESS (4,I)	THE2110
	IF (I) 1220,1230,730	THE2120
730	II=IARGS(4)	THE2130
	CALL ADRESS (2,IFQ)	THE2140
	IF (IFQ) 1220,1230,740	THE2150
740	CALL ADRESS (3,IG)	THE2160
	IF (IG) 1220,1230,750	THE2170
750	IWT=0	THE2180
	IF (NCOL-II-3) 1230,800,800	THE2190
760	IF (NARGS-4) 1210,770,1210	THE2200
770	CALL ADRESS (4,I)	THE2210
	IF (I) 1220,1230,780	THE2220
780	IWT=0	THE2230
	II=IARGS(4)	THE2240
	CALL ADRESS (2,IFQ)	THE2250
	IF (IFQ) 1220,1230,790	THE2260
790	CALL ADRESS (3,IG)	THE2270
	IF (IG) 1220,1230,800	THE2280
800	CALL ADRESS (1,ITP)	THE2290
	IF (ITP) 810,1230,820	THE2300
810	T=ARGS(1)	THE2310
	IF (T) 1250,1250,820	THE2320
820	IE=0	THE2330
	IF (NRMAX) 1240,1240,830	THE2340
830	IF (NERROR) 1180,840,1180	THE2350
840	IF (L2-8) 890,850,850	THE2360

```

850 DO 880 J=NROW,1,-1 THE2370
    IIG=IG+J-1 THE2380
    IF (RC(IIG)) 1260,880,860 THE2390
860 KK=J THE2400
    IF (KK-NS2) 890,890,870 THE2410
870 IF (L2-8) 890,890,1190 THE2420
880 CONTINUE THE2430
    GO TO 1270 THE2440
890 DO 1170 J=1,NRMAX THE2450
    IF (ITP-1) 910,900,900 THE2460
900 IIT=ITP+J-1 THE2470
    T=RC(IIT) THE2480
    IF (T) 1150,910,910 THE2490
910 IF (IWT) 940,920,930 THE2500
920 WT=1.0 THE2510
    GO TO 940 THE2520
930 IIW=IWT+J-1 THE2530
    WT=RC(IIW) THE2540
    IF (WT) 1150,940,940 THE2550
940 IF (L2-8) 1000,950,950 THE2560
950 Q0=0. THE2570
    Q1=0. THE2580
    Q2=0. THE2590
    DO 990 JJ=1,KK THE2600
    IIF=IFQ+JJ-1 THE2610
    IIG=IG+JJ-1 THE2620
    E=RC(IIF) THE2630
    G=RC(IIG) THE2640
    IF (G) 1260,960,960 THE2650
960 IF (E) 1130,970,970 THE2660
970 X=1.43879D0*DBLE(E)/DBLE(T) THE2670
    EXX=FDEXP(-X) THE2680
    Q0=Q0+G*EXX THE2690
    Q1=Q1+G*X*EXX THE2700
    Q2=Q2+G*X*X*EXX THE2710
    IF (L2-8) 990,990,980 THE2720
980 QQ(JJ)=G*EXX THE2730
990 CONTINUE THE2740
    GO TO 1010 THE2750
1000 Q0=1.0 THE2760
    Q1=0. THE2770
    Q2=0. THE2780
    IF (L2-6) 1020,1010,1010 THE2790
1010 FE=2.5*FLOG(T)+1.5*FLOG(WT)-3.66495+SNGL(FDLOG(Q0)) THE2800
    HE=2.5D0+Q1/Q0 THE2810
    S=FE+HE THE2820
    CP=2.5D0+Q2/Q0-(Q1/Q0)*(Q1/Q0) THE2830
    HBYT=HE*T THE2840
    GO TO 1050 THE2850
1020 IF (IFQ-1) 1040,1030,1030 THE2860
1030 IIF=IFQ+J-1 THE2870
    E=RC(IIF) THE2880
    IF (E) 1130,1040,1040 THE2890
1040 X=1.43879D0*DBLE(E)/DBLE(T) THE2900
    EXX=FDEXP(-X) THE2910
    EXDIF=1.0D0-EXX THE2920
    FE=-FDLOG(EXDIF)*R THE2930
    HE=(X*EXX/EXDIF)*R THE2940
    CP=R*X*X*EXX/(EXDIF*EXDIF) THE2950

```

	S=FE+HE	THE2960
	HBYT=HE*T	THE2970
1050	K=I+J-1	THE2980
	IF (L2-8) 1060,1090,1100	THE2990
1060	IF (L2-6) 1070,1080,1080	THE3000
1070	RC(K)=E	THE3010
	K=K+NROW	THE3020
1080	RC(K)=T	THE3030
	K=K+NROW	THE3040
	RC(K)=FE	THE3050
	K=K+NROW	THE3060
	RC(K)=HE	THE3070
	K=K+NROW	THE3080
	RC(K)=S	THE3090
	K=K+NROW	THE3100
	RC(K)=CP	THE3110
	K=K+NROW	THE3120
	RC(K)=HBYT	THE3130
	GO TO 1170	THE3140
1090	RC(K)=Q0	THE3150
	K=K+NROW	THE3160
	RC(K)=Q1	THE3170
	K=K+NROW	THE3180
	RC(K)=Q2	THE3190
	GO TO 1170	THE3200
1100	IF (NCOL-II-KK) 1200,1110,1110	THE3210
1110	DO 1120 JJ=1,KK	THE3220
	RC(K)=QQ(JJ)/Q0	THE3230
1120	K=K+NROW	THE3240
	GO TO 1170	THE3250
1130	Q0=1.0	THE3260
	Q1=0.	THE3270
	Q2=0.	THE3280
	IF (IE) 1010,1140,1010	THE3290
1140	CALL ERROR (229)	THE3300
	IE=1	THE3310
	GO TO 1010	THE3320
1150	FE=0.	THE3330
	HE=0.	THE3340
	CP=0.	THE3350
	S=0.	THE3360
	HBYT=0.	THE3370
	IF (IE) 1050,1160,1050	THE3380
1160	CALL ERROR (229)	THE3390
	IE=1	THE3400
	GO TO 1050	THE3410
1170	CONTINUE	THE3420
1180	RETURN	THE3430
1190	CALL ERROR (23)	THE3440
	GO TO 1180	THE3450
1200	CALL ERROR (17)	THE3460
	GO TO 1180	THE3470
1210	CALL ERROR (10)	THE3480
	GO TO 1180	THE3490
1220	CALL ERROR (20)	THE3500
	GO TO 1180	THE3510
1230	CALL ERROR (11)	THE3520
	GO TO 1180	THE3530
1240	CALL ERROR (9)	THE3540

	GO TO 1180	THE3550
1250	CALL ERROR (3)	THE3560
	GO TO 1180	THE3570
1260	CALL ERROR (25)	THE3580
	GO TO 1180	THE3590
1270	CALL ERROR (224)	THE3600
	GO TO 1180	THE3610
	END	THE3620

	SUBROUTINE TPCTPT (V,T)	TPC 10
C	VERSION 5.00        TPCTPT      5/15/70	TPC 20
	IF (V.LE.0.) GO TO 30	TPC 30
	IF (V-AINT(V)) 30,10,30	TPC 40
10	IF (V.GT.4.) GO TO 20	TPC 50
	T=3.6948*AINT(1./V)-1.6561*AINT(2./V)+.406*AINT(3./V)+2.7764*AINT(TPC 60	TPC 60
	14./V)	TPC 70
	RETURN	TPC 80
20	T=1.959964+2.3722712/V+2.8224986/V**2+2.5558497/V**3+1.5895341/V**TPC 90	TPC 90
	14+.73289821/V**5	TPC 100
C	25      FORMAT (1X/10X,,* INFORMATIVE DIAGNOSTIC * V IMPROPER,)	TPC 110
C	30      WRITE (IPRINT,25)	TPC 120
30	RETURN	TPC 130
	END	TPC 140

```

SUBROUTINE TRANSF                                     TRA 10
C VERSION 5.00      TRANSF   5/15/70                TRA 20
C SUBROUTINE TRANSF R.V. 5/2/68                      TRA 30
C *
C SUBROUTINE TO PROVIDE TRANSFORMATIONS B=UAU(T) AND C=U(I)AU    TRA 50
C L2=1      TRANSFORMATION B=UAU(T)                  TRA 60
C           GENERAL FORMS OF TRANSFORM                 TRA 70
C           M(XAXT)      A(,) K,K   U(,) N,K          STORE IN C(,) TRA 80
C L2=2      BACK TRANSFORMATION C=U(T)ALL            TRA 100
C           GENERAL FORMS OF BACKTRANS                 TRA 110
C           M(XTAX)      A(,) N,N   U(,) N,K          STORE IN C(,) TRA 120
C *
C COMMON /SCRAT/ NS,NS2,A(13500)                     TRA 150
C COMMON /BLOCRC/ NRC,RC(12600)                      TRA 160
C COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NTRA 170
IARGS,VWXYZ(8),NERROR                                TRA 180
DIMENSION ARGS(100)                                 TRA 190
EQUIVALENCE (ARGS(1),RC(12501))                   TRA 200
COMMON /BLOCKE/ NAME(4),L1,L2,ISRFLG               TRA 210
DIMENSION X(3000)                                   TRA 220
DOUBLE PRECISION X,SUM                            TRA 230
EQUIVALENCE (X,A)                                 TRA 240
C *
C CHECK TO SEE IF WE HAVE CORRECT NUMBER OF ARGUMENTS TRA 250
C *
IF(NARGS.NE.10) CALL ERROR (10)                    TRA 280
C *
C CHECK TO SEE IF ALL ARGUMENTS ARE INTEGERS        TRA 300
C *
J=NARGS                                         TRA 320
CALL CKIND (J)                                    TRA 330
IF (J.NE.0) CALL ERROR (3)                         TRA 340
C *
C CHECK TO SEE IF DIMENSIONS ARE CORRECT          TRA 360
C *
GO TO (30,40),L2                                    TRA 380
30 IF (IARGS(3).NE.IARGS(4).OR.IARGS(3).NE.IARGS(8)) CALL ERROR (3) TRA 440
GO TO 50                                         TRA 450
40 IF (IARGS(3).NE.IARGS(4).OR.IARGS(3).NE.IARGS(7)) CALL ERROR (3) TRA 460
C *
C CHECK TO SEE IF DIMENSIONS ARE OUT OF RANGE     TRA 480
C COMPUTE ADDRESSES                               TRA 490
C *
50 IF (NARGS.EQ.10) GO TO 60                         TRA 510
IARGS(12)=IARGS(L2+5)                             TRA 520
IARGS(11)=IARGS(L2+5)                             TRA 530
GO TO 70                                         TRA 540
60 IARGS(12)=IARGS(L2+6)                           TRA 550
IARGS(11)=IARGS(L2+6)                           TRA 560
GO TO 80                                         TRA 570
70 IARGS(10)=IARGS(NARGS)                          TRA 580
IARGS(9)=IARGS(NARGS-1)                          TRA 590
IARGS(8)=IARGS(NARGS-2)                          TRA 600
IARGS(7)=IARGS(NARGS-3)                          TRA 610
IARGS(6)=IARGS(5)                                TRA 620
IARGS(5)=IARGS(4)                                TRA 630
IARGS(4)=IARGS(3)                                TRA 640
80 J=3                                         TRA 650
CALL MTXCHK (J)                                TRA 660

```

```

        IF (J-1) 110,90,100                      TRA 670
90      CALL ERROR (3)                         TRA 680
       RETURN                                     TRA 690
100     CALL ERROR (17)                        TRA 700
       RETURN                                     TRA 710
C      *
C      CHECK FOR PREVIOUS ERRORS               TRA 720
C      *
110     IF (NERROR.NE.0) RETURN                 TRA 730
IROWA=IARGS(3)                           TRA 740
ISP=1                                    TRA 750
IROWU=IARGS(11)                          TRA 760
GO TO (120,130), L2                     TRA 770
120     IADD1=1                                TRA 780
IADD2=NROW                               TRA 790
GO TO 140                                 TRA 800
130     IADD1=NROW                            TRA 810
IADD2=1                                  TRA 820
140     DO 180 J=1,IROWU                      TRA 830
DO 170 I=1,IROWU                          TRA 840
IUP=IARGS(5)+(I-1)*IADD1                TRA 850
IA=IARGS(1)                             TRA 860
IUT=IARGS(5)+(J-1)*IADD1                TRA 870
ISX=NS2                                  TRA 880
DO 160 L=1,IROWA                         TRA 890
IU=IUP                                  TRA 900
DO 150 K=1,IROWA                         TRA 910
X(ISX)=RC(IU)*RC(IA)*RC(IUT)           TRA 920
ISX=ISX-1                               TRA 930
IU=IU+IADD2                            TRA 940
IA=IA+1                                  TRA 950
150     CONTINUE                               TRA 960
IA=IA+NROW-IROWA                         TRA 970
IUT=IUT+IADD2                           TRA 980
160     CONTINUE                               TRA 990
CALL SORTSM (IROWA*IROWA,SUM)           TRA1000
A(ISP)=SUM                               TRA1010
ISP=ISP+1                                TRA1020
170     CONTINUE                               TRA1030
180     CONTINUE                               TRA1040
C      *
C      STORE RESULTS IN WORKSHEET          TRA1050
C      *
190     IS=1                                   TRA1060
IC=IARGS(9)                            TRA1070
DO 200 J=1,IROWU                         TRA1080
DO 190 I=1,IROWU                         TRA1090
RC(IC)=A(IS)                            TRA1100
IS=IS+1                                  TRA1110
IC=IC+1                                  TRA1120
190     CONTINUE                               TRA1130
IC=IC+NROW-IROWU                         TRA1140
200     CONTINUE                               TRA1150
RETURN                                    TRA1160
END                                     TRA1170
                                         TRA1180
                                         TRA1190
                                         TRA1200
                                         TRA1210

```



```

IF (NARGS.EQ.5) M12=IARGS(5)          TWO 600
M13=NARGS                           TWO 610
M14=1                               TWO 620
M15=M9-1                            TWO 630
M16=M6+M8                           TWO 640
M17=M4/2                            TWO 650
M20=M9                             TWO 660
IF (M1.LT.2.OR.M2.LT.2) GO TO 90    TWO 670
M21=2*(M1+M2)+1                   TWO 680
C   LINES 30 TO 90 GIVE ERROR CHECKS  TWO 690
IF (M13-4) 10,20,30                TWO 700
10  CALL ERROR (10)                 TWO 710
RETURN
20  IF (IARGS(3)-IARGS(4)) 80,90,90  TWO 730
30  IF (M13-5) 40,40,10             TWO 740
40  IF (IARGS(5)-NCOL) 60,60,50    TWO 750
50  CALL ERROR (11)                 TWO 760
RETURN
60  IF (IARGS(3)-IARGS(5)) 70,90,70  TWO 780
70  IF (IARGS(5)-IARGS(4)) 80,90,90  TWO 790
80  DO 110 I=1,M13                TWO 800
IF (KIND(I)) 90,100,90            TWO 810
90  CALL ERROR (3)                 TWO 820
RETURN
100 IF (IARGS(I)) 50,50,110        TWO 840
110 CONTINUE
IF (NRMAX-IARGS(1)*IARGS(2)) 120,130,120  TWO 860
120 CALL ERROR (16)                 TWO 870
RETURN
130 IF (M5+M11+2.LT.NCOL.AND.M21+6.LT.NROW) GO TO 140  TWO 890
CALL ERROR (17)
RETURN
140 NARGS=M5
IF (M5+7.GT.NCOL) GO TO 10          TWO 930
DO 150 I=1,M5                      TWO 940
KIND(I)=0                           TWO 950
150 IARGS(I)=M11+I-1                TWO 960
CALL CHKCOLUMN (J)                  TWO 970
IF (J.NE.0) GO TO 90                TWO 980
IF (NERROR.NE.0) RETURN            TWO 990
C   LINES 100 TO 150 CONSTRUCT DESIGN MATRIX  TWO1000
K=IARGS(1)-1
DO 160 I=1,M4                      TWO1010
M19=K+I                           TWO1020
TWO1030
160 RC(M19)=1.0                     TWO1040
DO 180 I=2,M1                      TWO1050
K1=IARGS(I)                        TWO1060
DO 170 K=1,M6                      TWO1070
DO 170 J=1,M2                      TWO1080
K2=K1+M2*(K-1)+J-1                TWO1090
RC(K2)=0.
170 IF (K.EQ.I-1) RC(K2)=1.0        TWO1110
DO 180 J=1,M2                      TWO1120
K2=K1+M2*(M1-1)+J-1                TWO1130
180 RC(K2)=-1.0                     TWO1140
DO 200 I=M7,M5                    TWO1150
DO 200 K=1,M1                      TWO1160
DO 190 J=1,M8                      TWO1170
K2=IARGS(I)+M2*(K-1)+J-1          TWO1180

```

	RC(K2)=0.0	TW01190
190	IF (J.EQ.I-M1) RC(K2)=1.0	TW01200
	K2=IARGS(I)+M2*K-1	TW01210
200	RC(K2)=-1.0	TW01220
C	LINES 300 TO 325 CALL NEW ORTHO	TW01230
	NARGS=M1+M2+6	TW01240
	IARGS(1)=M3	TW01250
	IARGS(2)=M11	TW01260
	DO 210 I=4,NARGS	TW01270
210	IARGS(I)=M11+I-4	TW01280
	CALL ADRESS (1,K)	TW01290
	IF (M13.EQ.4) GO TO 220	TW01300
	IARGS(2)=M12	TW01310
	CALL ADRESS (2,J)	TW01320
	J=J-1	TW01330
220	CONS1=RC(K)	TW01340
	CONS2=RC(K)	TW01350
	DO 240 I=1,M4	TW01360
	IF (M13.EQ.4) GO TO 230	TW01370
	J=J+1	TW01380
	IF (RC(J)) 240,240,230	TW01390
230	IF (RC(K).LT.CONS1) CONS1=RC(K)	TW01400
	IF (RC(K).GT.CONS2) CONS2=RC(K)	TW01410
240	K=K+1	TW01420
	K=K-M4	TW01430
	B=(CONS1+CONS2)/2.0	TW01440
	DO 250 I=1,M4	TW01450
	M19=K-1+I	TW01460
250	RC(M19)=RC(M19)-B	TW01470
	DO 260 I=1,NARGS	TW01480
260	KIND(I)=0	TW01490
	L2=4	TW01500
	IARGS(3)=M1+M2-1	TW01510
	GO TO 690	TW01520
270	IF (NERROR.NE.0) RETURN	TW01530
C	LINES 330 TO 360 COMPUTE COEFFICIENTS AND THEIR STD. DEVIATIONS	TW01540
	CALL ADRESS (NARGS-3,K)	TW01550
	AST0(M1+11)=0.	TW01560
	DO 280 I=1,M6	TW01570
	M19=I+K	TW01580
	AST0(I+11)=RC(M19)	TW01590
	AST0(M1+11)=AST0(M1+11)-RC(M19)	TW01600
	K3=12+M1+M2+I	TW01610
	K4=K+M1+M8+I	TW01620
280	AST0(K3)=RC(K4)	TW01630
	K3=M1+M1+M2+11	TW01640
	AST0(K3+1)=AST0(K3)	TW01650
	K4=K+M6+M2	TW01660
	AST0(M5+13)=RC(K4)	TW01670
	AST0(M5+12)=0.	TW01680
	DO 290 I=1,M8	TW01690
	K4=11+M1+I	TW01700
	K5=K+M6+I	TW01710
	AST0(K4)=RC(K5)	TW01720
	K4=K5+M8+M1	TW01730
	K6=K3+I+1	TW01740
	AST0(K6)=RC(K4)	TW01750
290	AST0(M5+12)=AST0(M5+12)-RC(K5)	TW01760
	AST0(M21+11)=AST0(M21+10)	TW01770

C	K5=K+M21+7	TW01780
C	DO 285 I=1,6	TW01790
C	K5=K5-1	TW01800
C	285 RC(K5)=RC(K5-4)	TW01810
C	LINES 400 TO 430 COMPUTE TERMS FOR ANOVA	TW01820
C	IF (LL.EQ.7) GO TO 350	TW01830
C	IF (M13.EQ.5) GO TO 430	TW01840
300	CALL ADRESS (NARGS,K)	TW01850
	A(1)=0.	TW01860
	A(2)=0.	TW01870
	DO 310 I=1,M6	TW01880
	M19=K+I	TW01890
310	A(1)=A(1)+RC(M19)	TW01900
	DO 320 I=1,M8	TW01910
	K4=M6+I+K	TW01920
320	A(2)=A(2)+RC(K4)	TW01930
	K4=K+M5	TW01940
	A(3)=RC(K4)	TW01950
	A(4)=RC(K4+1)-RC(K)	TW01960
	A(5)=A(1)/FLOAT(M6)	TW01970
	A(6)=A(2)/FLOAT(M8)	TW01980
	A(7)=A(3)/FLOAT(M9)	TW01990
	A(8)=A(5)/A(7)	TW02000
	A(9)=A(6)/A(7)	TW02010
	CALL PROB (FLOAT(M6),FLOAT(M9),A(8),A(10))	TW02020
	CALL PROB (FLOAT(M8),FLOAT(M9),A(9),A(11))	TW02030
C	CALL RFORMAT (A(1),4,8,NW1,NDEC1,20,A(1),A(1),0,0)	TW02040
	CALL RFORMAT (A(5),3,8,NW2,NDEC2,20,A(1),A(1),0,0)	TW02050
	CALL PAGE (4)	TW02060
	WRITE (IPRINT,1680) M1,M2	TW02070
	IF (M13.EQ.5) GO TO 510	TW02080
	CALL RFORMAT (A(1),1,8,NW1,NDEC1,0,A(1),A(101),25-NW1,0)	TW02090
C	CALL RFORMAT (A(1),1,8,NW2,NDEC2,0,A(5),A(126),25-NW2,0)	TW02100
	LINES 500 TO 550 PRINT ANOVA WHEN ALL WEIGHTS EQUAL ONE.	TW02110
	WRITE (IPRINT,1690) M6,(A(I),I=101,150),A(8),A(10)	TW02120
	CALL RFORMAT (A(1),1,8,NW1,NDEC1,0,A(2),A(101),25-NW1,0)	TW02130
	CALL RFORMAT (A(1),1,8,NW2,NDEC2,0,A(6),A(126),25-NW2,0)	TW02140
	WRITE (IPRINT,1700) M8,(A(I),I=101,150),A(9),A(11)	TW02150
	CALL RFORMAT (A(1),1,8,NW1,NDEC1,0,A(3),A(201),25-NW1,0)	TW02160
	CALL RFORMAT (A(1),1,8,NW2,NDEC2,0,A(7),A(226),25-NW2,0)	TW02170
	WRITE (IPRINT,1710) M9,(A(I),I=201,250)	TW02180
	CALL RFORMAT (A(1),1,8,NW1,NDEC1,0,A(4),A(101),25-NW1,0)	TW02190
C	WRITE (IPRINT,1720) M10,(A(I),I=101,125)	TW02200
	LINES 600 TO 645 COMPUTE AND PRINT FOR TUKEY'S TEST	TW02210
	CALL ADRESS (1,J)	TW02220
	A(12)=0.	TW02230
	DO 330 I1=1,M1	TW02240
	DO 330 I2=1,M2	TW02250
	J1=J+M2*(I1-1)+I2-1	TW02260
	M19=M1+I1+I2	TW02270
330	A(12)=A(12)+AST0(I1+I1)*AST0(M19)*RC(J1)	TW02280
	A(12)=(A(12)*A(12))/((A(1)/FLOAT(M1))*(A(2)/FLOAT(M2)))	TW02290
	A(13)=A(3)-A(12)	TW02300
	A(6)=A(13)/FLOAT(M9-1)	TW02310
	A(16)=A(12)/A(6)	TW02320
	CALL PROB (1.,FLOAT(M9)-1.,A(16),A(17))	TW02330
	A(5)=A(12)	TW02340
	A(14)=A(3)	TW02350
	CALL RFORMAT (A(12),3,8,NW1,NDEC1,20,A(1),A(1),0,0)	TW02360

```

CALL RFORMAT (A(5),3,8,NW2,NDEC2,20,A(1),A(1),0,0)           TW02370
CALL RFORMAT (A(1),1,8,NW1,NDEC1,0,A(12),A(101),25-NW1,0)   TW02380
CALL RFORMAT (A(1),1,8,NW2,NDEC2,0,A(5),A(126),25-NW2,0)   TW02390
WRITE (IPRINT,1740) M14,(A(I),I=101,150),A(16),A(17)        TW02400
CALL RFORMAT (A(1),1,8,NW1,NDEC1,0,A(13),A(101),25-NW1,0)   TW02410
CALL RFORMAT (A(1),1,8,NW2,NDEC2,0,A(6),A(126),25-NW2,0)   TW02420
WRITE (IPRINT,1750) M15,(A(I),I=101,150)                     TW02430
CALL RFORMAT (A(1),1,8,NW1,NDEC1,0,A(14),A(101),25-NW1,0)   TW02440
CALL RFORMAT (A(1),1,8,NW2,NDEC2,0,A(7),A(126),25-NW2,0)   TW02450
WRITE (IPRINT,1710) M9,(A(I),I=101,150)                      TW02460
WRITE (IPRINT,1760)                                         TW02470
DO 340 I=1,4                                              TW02480
340 ASTO(I)=A(I)                                         TW02490
C LINES 650 TO 690 PRINT COEFFICIENTS AND THEIR STD. DEVIATIONS TW02500
350 CALL ADRESS (1,K)                                     TW02510
DO 360 I=1,M4                                         TW02520
K5=K-1+I                                         TW02530
360 RC(K5)=RC(K5)+B                                     TW02540
J=K                                              TW02550
CALL ADRESS (NARGS-3,K)                                TW02560
K5=K+M21+7                                         TW02570
DO 370 I=1,6                                         TW02580
K5=K5-1                                         TW02590
370 RC(K5)=RC(K5-4)                                    TW02600
DO 380 I=1,M21                                     TW02610
K5=K+I                                         TW02620
380 RC(K5)=ASTO(I+11)                                TW02630
RC(K)=RC(K)+B                                     TW02640
K5=K+M21+1                                         TW02650
A(1000)=RC(K5)                                     TW02660
RC(K5)=RC(K5+3)                                    TW02670
CALL ADRESS (2,J1)                                 TW02680
CALL ADRESS (NARGS,J2)                                TW02690
A(1002)=0.0                                         TW02700
SUM=0.0D0                                         TW02710
DO 390 I=1,M4                                     TW02720
A(1002)=A(1002)+RC(J1)                            TW02730
SUM=SUM+RC(J1)*RC(J)**2                           TW02740
J=J+1                                         TW02750
390 J1=J1+1                                         TW02760
J1=J2+M1+M2                                         TW02770
RC(J1)=SUM                                         TW02780
RC(J1+1)=RC(J1+1)+B*FSQRT(A(1002))             TW02790
RC(J2)=RC(J1+1)**2                                TW02800
IF (LL.EQ.6) GO TO 400                           TW02810
RC(K5)=A(1000)                                     TW02820
RETURN                                         TW02830
400 K5=K+M5+2                                         TW02840
CALL RFORMAT (RC(K),M5+2,8,NW3,NDEC3,20,A(1),A(1),0,0)   TW02850
CALL RFORMAT (RC(K5),M5+3,8,NW4,NDEC4,20,A(1),A(1),0,0)   TW02860
CALL RFORMAT (A(1),1,8,NW3,NDEC3,0,RC(K),A(101),25-NW3,0) TW02870
CALL RFORMAT (A(1),1,8,NW4,NDEC4,0,RC(K5),A(126),25-NW4,0) TW02880
WRITE (IPRINT,1770) (A(J),J=101,150)                 TW02890
K5=K+M21+1                                         TW02900
RC(K5)=A(1000)                                     TW02910
DO 410 I=1,M1                                     TW02920
K5=K+I                                         TW02930
K6=K5+M5+2                                         TW02940
CALL RFORMAT (A(1),1,8,NW3,NDEC3,0,RC(K5),A(101),25-NW3,0) TW02950

```

	CALL RFORMAT (A(1),1,8,NW4,NDEC4,0,RC(K6),A(126),25-NW4,0)	TW02960
410	WRITE (IPRINT,1780) I,(A(J),J=101,150)	TW02970
	WRITE (IPRINT,1760)	TW02980
	DO 420 I=1,M2	TW02990
	K5=K+M1+I	TW03000
	K6=K5+M5+2	TW03010
	CALL RFORMAT (A(1),1,8,NW3,NDEC3,0,RC(K5),A(101),25-NW3,0)	TW03020
	CALL RFORMAT (A(1),1,8,NW4,NDEC4,0,RC(K6),A(126),25-NW4,0)	TW03030
420	WRITE (IPRINT,1790) I,(A(J),J=101,150)	TW03040
	K6=K+M21+4	TW03050
	CALL RFORMAT (A(1),1,8,NW4,NDEC4,0,RC(K6),A(101),50-NW4,0)	TW03060
	WRITE (IPRINT,1760)	TW03070
	WRITE (IPRINT,1800) (A(J),J=101,150)	TW03080
	ASTO(1)=ASTO(1)+ASTO(2)+ASTO(3)	TW03090
	ASTO(2)=ABS(ASTO(4)-ASTO(1))/ASTO(4)	TW03100
	IF (ASTO(2).GT.5.E-7) CALL ERROR (228)	TW03110
	GO TO 630	TW03120
C	LINES 700 TO 780 GIVE ANOVA WHEN WEIGHTS ARE SPECIFIED	TW03130
430	CALL ADRESS (2,J)	TW03140
C	CHECK ON WEIGHTS	TW03150
	K5=J-1	TW03160
	DO 470 I1=1,M1	TW03170
	M31=0	TW03180
	DO 460 I2=1,M2	TW03190
	K5=K5+1	TW03200
	IF (RC(K5)) 440,450,460	TW03210
440	CALL ERROR (223)	TW03220
	RETURN	TW03230
450	M31=M31+1	TW03240
	M9=M9-1	TW03250
460	CONTINUE	TW03260
	IF (M31.NE.M2) GO TO 470	TW03270
	CALL ERROR (224)	TW03280
	RETURN	TW03290
470	CONTINUE	TW03300
	K5=J-1	TW03310
	DO 500 I1=1,M2	TW03320
	M31=0	TW03330
	DO 490 I2=1,M1	TW03340
	K5=K5+1	TW03350
	IF (RC(K5)) 480,480,490	TW03360
480	M31=M31+1	TW03370
490	CONTINUE	TW03380
	IF (M3.NE.M1) GO TO 500	TW03390
	CALL ERROR (224)	TW03400
	RETURN	TW03410
500	CONTINUE	TW03420
	M10=M6+M8+M9	TW03430
	GO TO 300	TW03440
510	J=5	TW03450
	DO 520 I=1,M8	TW03460
	IARGS(J)=M11+M6+I	TW03470
520	J=J+1	TW03480
	DO 530 I=1,M6	TW03490
	IARGS(J)=M11+I	TW03500
530	J=J+1	TW03510
	DO 540 I=1,11	TW03520
540	ASTO(I)=A(I)	TW03530
	CALL ADRESS (1,J)	TW03540

	CALL ADRESS (2,K)	TW03550
	DO 550 I=1,M17	TW03560
	K5=J-1+I	TW03570
	K6=J+M4-I	TW03580
	A(20)=RC(K5)	TW03590
	RC(K5)=RC(K6)	TW03600
	RC(K6)=A(20)	TW03610
	K5=K-1+I	TW03620
	K6=K+M4-I	TW03630
	A(20)=RC(K5)	TW03640
	RC(K5)=RC(K6)	TW03650
550	RC(K6)=A(20)	TW03660
	GO TO 690	TW03670
560	CALL ADRESS (1,J)	TW03680
	CALL ADRESS (2,K)	TW03690
	CALL ADRESS (NARGS-2,K1)	TW03700
	CALL ADRESS (NARGS-1,K2)	TW03710
	DO 570 I=1,M17	TW03720
	K5=J-1+I	TW03730
	K6=J+M4-I	TW03740
	A(20)=RC(K5)	TW03750
	RC(K5)=RC(K6)	TW03760
	RC(K6)=A(20)	TW03770
	K5=K2-1+I	TW03780
	K6=K2+M4-I	TW03790
	A(20)=RC(K5)	TW03800
	RC(K5)=RC(K6)	TW03810
	RC(K6)=A(20)	TW03820
	K5=K1-1+I	TW03830
	K6=K1+M4-I	TW03840
	A(20)=RC(K5)	TW03850
	RC(K5)=RC(K6)	TW03860
	RC(K6)=A(20)	TW03870
	K5=K-1+I	TW03880
	K6=K+M4-I	TW03890
	A(20)=RC(K5)	TW03900
	RC(K5)=RC(K6)	TW03910
570	RC(K6)=A(20)	TW03920
	A(18)=0.	TW03930
	CALL ADRESS (NARGS,K)	TW03940
	DO 580 I=1,M8	TW03950
	K5=I+K	TW03960
580	A(18)=A(18)+RC(K5)	TW03970
	A(17)=0.	TW03980
	DO 590 I=1,M6	TW03990
	K5=M2+I+K-1	TW04000
590	A(17)=A(17)+RC(K5)	TW04010
	J1=M21-2	TW04020
	K5=201	TW04030
	K6=K+1	TW04040
	DO 600 I=1,J1	TW04050
	A(K5)=RC(K6)	TW04060
	K5=K5+1	TW04070
600	K6=K6+1	TW04080
	K5=K+1	TW04090
	K6=201+M8	TW04100
	J1=K+M16+4	TW04110
	J2=204+M16+M8	TW04120
	DO 610 I=1,M6	TW04130

	RC(K5)=A(K6)	TW04140
	RC(J1)=A(J2)	TW04150
	K5=K5+1	TW04160
	K6=K6+1	TW04170
	J1=J1+1	TW04180
610	J2=J2+1	TW04190
	J1=K+M16+M8+3	TW04200
	J2=204+M16	TW04210
	DO 620 I=1,M8	TW04220
	RC(K5)=A(I+200)	TW04230
	RC(J1)=A(J2)	TW04240
	K5=K5+1	TW04250
	J1=J1+1	TW04260
620	J2=J2+1	TW04270
	A(19)=A(17)/FLOAT(M6)	TW04280
	A(20)=A(18)/FLOAT(M8)	TW04290
	A(21)=A(19)/ASTO(7)	TW04300
	CALL PROB (FLOAT(M6),FLOAT(M9),A(21),A(22))	TW04310
	CALL RFORMAT (A(1),1,8,NW1,NDEC1,0,A(17),A(101),25-NW1,0)	TW04320
	CALL RFORMAT (A(1),1,8,NW2,NDEC2,0,A(19),A(126),25-NW2,0)	TW04330
	WRITE (IPRINT,1690) M6,(A(I),I=101,150),A(21),A(22)	TW04340
	CALL RFORMAT (A(1),1,8,NW1,NDEC1,0,A(18),A(101),25-NW1,0)	TW04350
	CALL RFORMAT (A(1),1,8,NW2,NDEC2,0,A(20),A(126),25-NW2,0)	TW04360
	WRITE (IPRINT,1700) M8,(A(I),I=101,150)	TW04370
	CALL RFORMAT (A(1),1,8,NW1,NDEC1,0,ASTO(3),A(101),25-NW1,0)	TW04380
	CALL RFORMAT (A(1),1,8,NW2,NDEC2,0,ASTO(7),A(126),25-NW2,0)	TW04390
	WRITE (IPRINT,1710) M9,(A(I),I=101,150)	TW04400
	CALL RFORMAT (A(1),1,8,NW1,NDEC1,0,ASTO(4),A(101),25-NW1,0)	TW04410
	WRITE (IPRINT,1720) M10,(A(I),I=101,125)	TW04420
	CALL RFORMAT (A(1),1,8,NW1,NDEC1,0,ASTO(1),A(101),25-NW1,0)	TW04430
	CALL RFORMAT (A(1),1,8,NW2,NDEC2,0,ASTO(5),A(126),25-NW2,0)	TW04440
	WRITE (IPRINT,1690) M6,(A(I),I=101,150)	TW04450
	CALL RFORMAT (A(1),1,8,NW1,NDEC1,0,ASTO(2),A(101),25-NW1,0)	TW04460
	CALL RFORMAT (A(1),1,8,NW2,NDEC2,0,ASTO(6),A(126),25-NW2,0)	TW04470
	WRITE (IPRINT,1700) M8,(A(I),I=101,150),ASTO(9),ASTO(11)	TW04480
	CALL RFORMAT (A(1),1,8,NW1,NDEC1,0,ASTO(3),A(101),25-NW1,0)	TW04490
	CALL RFORMAT (A(1),1,8,NW2,NDEC2,0,ASTO(7),A(126),25-NW2,0)	TW04500
	WRITE (IPRINT,1710) M9,(A(I),I=101,150)	TW04510
	CALL RFORMAT (A(1),1,8,NW1,NDEC1,0,ASTO(4),A(101),25-NW1,0)	TW04520
	WRITE (IPRINT,1720) M10,(A(I),I=101,125)	TW04530
	M31=M5+M9	TW04540
	M32=M4-M31	TW04550
	WRITE (IPRINT,1730) M31,M32,M12	TW04560
	CALL ADRESS (NARGS-3,K)	TW04570
	K5=K+M6+M2+M2	TW04580
	K6=K+M1+M2	TW04590
	K3=M1+M1+M2+12	TW04600
	ASTO(K3)=RC(K5)	TW04610
	ASTO(M21+11)=RC(K6)	TW04620
	GO TO 350	TW04630
630	CALL PAGE (4)	TW04640
	WRITE (IPRINT,1810) M1,M2	TW04650
	M31=MIN0(15,M2)	TW04660
	DO 640 I=1,M31	TW04670
640	KIND(I)=I	TW04680
	WRITE (IPRINT,1820) (KIND(I),I=1,M31)	TW04690
	WRITE (IPRINT,1830)	TW04700
	CALL ADRESS (NARGS-2,J)	TW04710
	CALL ADRESS (NARGS-1,K3)	TW04720

```

IF (M13.EQ.5) KIND(2)=0          TW04730
IF (M13.EQ.5) CALL ADRESS (2,M32) TW04740
DO 670 I=1,M4                   TW04750
IF (M13.EQ.5) GO TO 650         TW04760
A(I)=RC(J)/FSQRT(RC(K6)**2-RC(K3)**2)
GO TO 660                         TW04780
650 IF (RC(M32).GT.0.0) A(I)=RC(J)/FSQRT((RC(K6)**2)/RC(M32)-RC(K3)**2) TW04790
1)                                TW04800
IF (RC(M32).LE.0.0) A(I)=0.0      TW04810
M32=M32+1                         TW04820
660 J=J+1                          TW04830
670 K3=K3+1                        TW04840
DO 680 I=1,M1                     TW04850
WRITE (IPRINT,1760)                TW04860
M31=M2*(I-1)                      TW04870
KA=M31+M2                         TW04880
KB=M31+1                          TW04890
680 WRITE (IPRINT,1840) I,(A(K),K=KB,KA) TW04900
RETURN                             TW04910
C THIS IS NEW ORTHO - SFIT PORTION TW04920
690 NMUI=1                          TW04930
CALL ADRESS (1,IIRGS(1))          TW04940
CALL ADRESS (2,IIRGS(2))          TW04950
NST=1                              TW04960
NEND=NARGS                         TW04970
DO 700 I=4,NEND                   TW04980
CALL ADRESS (I,IIRGS(I))          TW04990
700 CONTINUE                         TW05000
M=IARGS(3)                         TW05010
N=NRMAX                            TW05020
FN=N                               TW05030
SU=0.0                             TW05040
L22=IIRGS(2)                       TW05050
L22A=L22                           TW05060
DO 730 I=1,N                        TW05070
IF (RC(L22A)) 710,730,720        TW05080
710 CALL ERROR (25)                TW05090
RETURN                             TW05100
720 SU=SU+1.0                       TW05110
730 L22A=L22A+1                    TW05120
FM=M                               TW05130
IF (SU-FM) 740,750,760             TW05140
740 CALL ERROR (24)                TW05150
RETURN                             TW05160
750 DENOM=1.0                       TW05170
GO TO 770                           TW05180
760 DENOM=FSQRT(SU-FM)              TW05190
770 NPM=N+M                          TW05200
M31=M-1                           TW05210
M32=M+1                           TW05220
N1=N-1                            TW05230
N2=N+1                            TW05240
MD1=(M*(M32))/2                   TW05250
ND1=M32*NPM                        TW05260
ND2=M*NPM                          TW05270
MD3=ND2+N                          TW05280
ND3=ND1                            TW05290
ND4=ND3+NPM                        TW05300
ND5=ND4+NPM                        TW05310

```

ND6=ND5+M32	TW05320
ND66=MD1+M	TW05330
ND7=ND6+ND66	TW05340
ND8=ND7+ND66	TW05350
ND9=ND8+M32	TW05360
ND10=ND9+M	TW05370
ND11=ND10+M	TW05380
ND12=ND11+M	TW05390
ND13=ND12+MD1	TW05400
ND14=ND13+M32	TW05410
ND16=ND14+M	TW05420
ND17=ND16+M	TW05430
ND18=ND17+M	TW05440
ND19=ND18+NPM	TW05450
ND20=ND19+N	TW05460
IF (ND20.GT.NS) CALL ERROR (23)	TW05470
IF (NERROR.NE.0) RETURN	TW05480
NRBAR=1	TW05490
I=4	TW05500
MXARGS=M+4	TW05510
L44=MXARGS-1	TW05520
J=1	TW05530
DO 790 I1=I,L44	TW05540
K1=J	TW05550
L33=IIRGS(I1)	TW05560
K2=K1	TW05570
DO 780 I2=1,N	TW05580
A(K2)=RC(L33)	TW05590
K2=K2+1	TW05600
780 L33=L33+1	TW05610
790 J=J+NPM	TW05620
K1=N2	TW05630
DO 810 K=1,M	TW05640
K2=K1	TW05650
DO 800 I=1,M	TW05660
A(K2)=0.	TW05670
800 K2=K2+1	TW05680
K2=K1+K-1	TW05690
K1=K1+NPM	TW05700
810 A(K2)=1.0	TW05710
NBEI=1	TW05720
NRHI=1	TW05730
I18=1+ND13	TW05740
NGAI=2	TW05750
NSII=2	TW05760
NDEI=1	TW05770
NNUI=1	TW05780
LZ1=1	TW05790
LZ2=1	TW05800
K=1	TW05810
820 NTHI=1	TW05820
830 NALI=1	TW05830
NOMI=1	TW05840
NJ=ND3+N+1	TW05850
DO 840 J=1,M	TW05860
A(NJ)=0.	TW05870
840 NJ=NJ+1	TW05880
KD1=(K-1)*NPM	TW05890
I1=ND3+1	TW05900

	I2=KD1+1	TW05910
	L22A=L22	TW05920
	DO 860 I=1,N	TW05930
	A(I1)=A(I2)*RC(L22A)	TW05940
	L22A=L22A+1	TW05950
	I1=I1+1	TW05960
860	I2=I2+1	TW05970
	GO TO (870,900), NOMI	TW05980
870	IA1=1	TW05990
	IA2=ND5+1	TW06000
	DO 890 I=1,K	TW06010
	I2=IA1	TW06020
	SUM=0.0	TW06030
	J2=ND3+1	TW06040
	DO 880 J=1,NPM	TW06050
	SUM=SUM+A(J2)*A(I2)	TW06060
	I2=I2+1	TW06070
880	J2=J2+1	TW06080
	A(IA2)=SUM	TW06090
	IA1=IA1+NPM	TW06100
890	IA2=IA2+1	TW06110
	GO TO 930	TW06120
900	DK2=0.	TW06130
	I1=(K-1)*NPM+1	TW06140
	IND3=ND3+1	TW06150
	DO 910 I=1,NPM	TW06160
	DK2=DK2+A(IND3)*A(I1)	TW06170
	I1=I1+1	TW06180
910	IND3=IND3+1	TW06190
	DK=FDSQRT(DK2)	TW06200
	A(I18)=DK	TW06210
	I18=I18+1	TW06220
	K1=(K-1)*NPM+1	TW06230
	DO 920 I=1,NPM	TW06240
	A(K1)=A(K1)/DK	TW06250
920	K1=K1+1	TW06260
	NOMI=1	TW06270
	GO TO 850	TW06280
930	GO TO (940,1000), NDEI	TW06290
940	LZ1=-LZ1	TW06300
	IF (LZ1) 990,950,950	TW06310
950	K1=K-1	TW06320
	IRUTH=ND5+1	TW06330
	DO 960 I=1,K1	TW06340
	A(IRUTH)=-A(IRUTH)	TW06350
960	IRUTH=IRUTH+1	TW06360
	IRUTH=K+ND5	TW06370
	A(IRUTH)=1.0	TW06380
	J2=ND4+1	TW06390
	DO 980 I=1,NPM	TW06400
	SUM=0.0	TW06410
	J1=I	TW06420
	J3=ND5+1	TW06430
	DO 970 J=1,K	TW06440
	SUM=SUM+A(J1)*A(J3)	TW06450
	J1=J1+NPM	TW06460
970	J3=J3+1	TW06470
	A(J2)=SUM	TW06480
980	J2=J2+1	TW06490

	GO TO 1090	TW06500
990	ISAL=I18+M32	TW06510
	IRUTH=ND5+K	TW06520
	A(ISAL)=FSQRT(A(IRUTH))	TW06530
	GO TO 950	TW06540
1000	LZ2=-LZ2	TW06550
	IF (LZ2) 1010,950,950	TW06560
1010	DO 1020 I=1,M	TW06570
	IND5=ND5+I	TW06580
	IND9=ND9+I	TW06590
	IND8=ND8+I	TW06600
	A(IND8)=A(IND5)	TW06610
1020	A(IND9)=A(IND5)*A(IND5)	TW06620
	A(IND8+1)=A(IND5+1)	TW06630
	A(ND10+1)=A(IND8+1)-A(ND9+1)	TW06640
	IND10=ND10+1	TW06650
	IND9=ND9+1	TW06660
	DO 1030 J=1,M	TW06670
	IND10=IND10+1	TW06680
	IND9=IND9+1	TW06690
1030	A(IND10)=A(IND10-1)-A(IND9)	TW06700
	FI=1.0	TW06710
	IND10=ND10	TW06720
	IND11=ND11	TW06730
	DO 1080 I=1,M	TW06740
	IND10=IND10+1	TW06750
	IND11=IND11+1	TW06760
	IF (FN-FI) 1070,1070,1040	TW06770
1040	IF (A(IND10)) 1050,1060,1060	TW06780
1050	A(IND11)=-FSQRT(ABS(A(IND10))/(FN-FI))	TW06790
	GO TO 1080	TW06800
1060	A(IND11)=FSQRT(A(IND10)/(FN-FI))	TW06810
	GO TO 1080	TW06820
1070	A(IND10)=-1.0	TW06830
1080	FI=FI+1.0	TW06840
	GO TO 950	TW06850
1090	GO TO (1100,1120,1250), NTHI	TW06860
1100	K1=(K-1)*NPM+1	TW06870
	IND4=ND4+1	TW06880
	DO 1110 I=1,NPM	TW06890
	A(K1)=A(IND4)	TW06900
	K1=K1+1	TW06910
1110	IND4=IND4+1	TW06920
	GO TO 1210	TW06930
1120	IND18=ND18+1	TW06940
	IND4=ND4+1	TW06950
	DO 1130 I=1,N	TW06960
	A(IND18)=A(IND4)	TW06970
	IND18=IND18+1	TW06980
1130	IND4=IND4+1	TW06990
	NI=N+1	TW07000
	DO 1140 I=1,M	TW07010
	KK1=ND18+NI	TW07020
	IND4=ND4+NI	TW07030
	A(KK1)=-A(IND4)	TW07040
1140	NI=NI+1	TW07050
	IND4=ND4	TW07060
	IND19=ND19	TW07070
	DO 1150 I=1,N	TW07080

	IND4=IND4+1	TW07090
	IND19=IND19+1	TW07100
1150	A(IND19)=A(IND4)	TW07110
	IF (NARGS-MXARGS) 1200,1180,1160	TW07120
1160	L66=IIRGS(MXARGS+1)-1	TW07130
	L66A=L66	TW07140
	IND4=ND4	TW07150
	DO 1170 I=1,N	TW07160
	L66A=L66A+1	TW07170
	IND4=IND4+1	TW07180
1170	RC(L66A)=A(IND4)	TW07190
1180	L55=IIRGS(MXARGS)	TW07200
	L55A=L55-1	TW07210
	NI=N+ND4	TW07220
	DO 1190 I=1,M	TW07230
	NI=NI+1	TW07240
	L55A=L55A+1	TW07250
1190	RC(L55A)=-A(NI)	TW07260
1200	NTHI=3	TW07270
	GO TO 1100	TW07280
1210	GO TO (1220,1230), NALI	TW07290
1220	NOMI=2	TW07300
	NALI=2	TW07310
	GO TO 850	TW07320
1230	IF (K-M) 1240,1280,1280	TW07330
1240	K=K+1	TW07340
	GO TO 820	TW07350
1250	GO TO (1260,1270), NNUI	TW07360
1260	NNUI=2	TW07370
	GO TO 1370	TW07380
1270	SS=DK/DENOM	TW07390
	SSQ=SS*SS	TW07400
	GO TO 1370	TW07410
1280	GO TO (1290,1250), NBEI	TW07420
1290	K1=1	TW07430
	DO 1310 I=1,M	TW07440
	I1=I*N+(I-1)*M	TW07450
	DO 1300 J=1,I	TW07460
	I2=J+I1	TW07470
	K2=K1+ND12	TW07480
	A(K2)=A(I2)	TW07490
1300	K1=K1+1	TW07500
1310	CONTINUE	TW07510
	NDEI=2	TW07520
	NBEI=2	TW07530
	NTHI=2	TW07540
	K=K+1	TW07550
	GO TO (1250,1320), NGAI	TW07560
1320	CONTINUE	TW07570
	DO 1340 IL=1,M	TW07580
	LOC=IX1(IL,0,ND6)+1	TW07590
	DO 1340 J=1,IL	TW07600
	SUM=0.	TW07610
	DO 1330 KK=IL,M	TW07620
	LOC1=IX1(KK,IL,ND12)	TW07630
	LOC2=IX1(KK,J,ND12)	TW07640
1330	SUM=SUM+A(LOC1)*A(LOC2)	TW07650
	A(LOC)=SUM	TW07660
1340	LOC=LOC+1	TW07670

J1=3+ND6	TW07680
J=ND16+2	TW07690
A(ND16+1)=FSQRT(A(ND6+1))	TW07700
DO 1350 I=2,M	TW07710
IF (M.EQ.1) GO TO 1360	TW07720
A(J)=FSQRT(A(J1))	TW07730
J=J+1	TW07740
1350 J1=J1+I+1	TW07750
1360 NGA1=1	TW07760
GO TO 1250	TW07770
1370 GO TO (1380,1420), NRHI	TW07780
1380 IF (NRBAR) 1390,1460,1390	TW07790
1390 NRBAR=NRBAR-1	TW07800
NTHI=2	TW07810
NRHI=2	TW07820
L11=IIRGS(1)-1	TW07830
L11A=L11+1	TW07840
I1=ND2+1	TW07850
DO 1400 I=1,N	TW07860
A(I1)=RC(L11A)	TW07870
L11A=L11A+1	TW07880
1400 I1=I1+1	TW07890
I1=MD3+1	TW07900
DO 1410 I=1,M	TW07910
A(I1)=0.	TW07920
1410 I1=I1+1	TW07930
GO TO 830	TW07940
1420 GO TO (1380,1430), NSII	TW07950
1430 IND7=ND7+1	TW07960
IND6=ND6+1	TW07970
DO 1440 I=1,MD1	TW07980
A(IND7)=SSQ*A(IND6)	TW07990
IND7=IND7+1	TW08000
1440 IND6=IND6+1	TW08010
IND16=ND16+1	TW08020
IND17=ND17+1	TW08030
DO 1450 I=1,M	TW08040
A(IND17)=SS*A(IND16)	TW08050
IND16=IND16+1	TW08060
1450 IND17=IND17+1	TW08070
GO TO 1380	TW08080
1460 L55A=L55+M	TW08090
ISF=1	TW08100
L55B=L55A+M-1	TW08110
IF (L55B.LE.NROW+L55-1) GO TO 1470	TW08120
IF (M.GE.NROW) GO TO 1580	TW08130
ISF=2	TW08140
L55B=L55+NROW-1	TW08150
1470 IND17=ND17+1	TW08160
DO 1480 I=L55A,L55B	TW08170
RC(I)=A(IND17)	TW08180
1480 IND17=IND17+1	TW08190
GO TO (1490,1580), ISF	TW08200
1490 IMS=NROW-2*M	TW08210
IF (IMS.GT.6) IMS=6	TW08220
GO TO (1570,1560,1550,1540,1530,1500), IMS	TW08230
1500 L11A=IIRGS(1)	TW08240
L22A=L22	TW08250
YBAR=0.0	TW08260

```

WSUM=0.0                                     TW08270
DO 1510 I=1,N                                TW08280
WSUM=WSUM+RC(L22A)                           TW08290
YBAR=YBAR+RC(L22A)*RC(L11A)                  TW08300
L22A=L22A+1                                    TW08310
1510  L11A=L11A+1                            TW08320
YBAR=YBAR/WSUM                               TW08330
L11A=IIRGS(1)                                TW08340
L22A=L22                                     TW08350
YYBAR=0.                                      TW08360
DO 1520 I=1,N                                TW08370
YYBAR=RC(L22A)*(RC(L11A)-YBAR)**2+YYBAR    TW08380
L22A=L22A+1                                  TW08390
1520  L11A=L11A+1                            TW08400
R2=1.-SSQ*(SU-FM)/YYBAR                      TW08410
IF (R2.LT.0.) R2=0.0                          TW08420
IF (R2.GT.1.0) R2=1.0                         TW08430
RC(L55B+6)=R2                                 TW08440
1530  RC(L55B+5)=SSQ                         TW08450
1540  RC(L55B+4)=SS                         TW08460
1550  RC(L55B+3)=SU-FM                      TW08470
1560  RC(L55B+2)=FM                         TW08480
1570  RC(L55B+1)=SU                         TW08490
1580  IND2=ND2+1                            TW08500
IND3=ND3+1                                    TW08510
IND4=ND19+1                                   TW08520
IPIC=1                                       TW08530
IY=IIRGS(1)                                 TW08540
YSUM=0.0                                     TW08550
L22A=L22                                     TW08560
DO 1600 I=1,N                                TW08570
IP=IPIC                                     TW08580
SP=0.0                                       TW08590
DO 1590 II=1,M                               TW08600
SP=SP+A(IP)**2                             TW08610
1590  IP=IP+NPM                            TW08620
A(IND2)=FSQRT(SP)*SS                        TW08630
IPIC=IPIC+1                                 TW08640
IND2=IND2+1                                 TW08650
A(IND3)=RC(IY)-A(IND4)                      TW08660
IND3=IND3+1                                 TW08670
IND4=IND4+1                                 TW08680
YSUM=YSUM+RC(IY)**2*RC(L22A)                TW08690
L22A=L22A+1                                TW08700
1600  IY=IY+1                                TW08710
LST=IIRGS(MXARGS+3)                         TW08720
IND9=ND9                                     TW08730
DO 1610 I=1,M                                TW08740
IND9=IND9+1                                 TW08750
RC(LST)=A(IND9)                            TW08760
1610  LST=LST+1                            TW08770
IF (NROW-(M+1)) 1650,1630,1620            TW08780
1620  RC(LST+1)=A(ND9)                      TW08790
1630  RC(LST)=(SU-FM)*SSQ                  TW08800
LST=LST+2                                    TW08810
IF (M+2.GE.NROW) GO TO 1650                 TW08820
LSTA=LST+M-1                                TW08830
IF (2*M+2.GT.NROW) LSTA=IIRGS(MXARGS+3)+NROW-1 TW08840
IND8=ND8                                     TW08850

```

```

DO 1640 I=LST,LSTA                                TW08860
IND8=IND8+1                                         TW08870
1640 RC(I)=A(IND8)                                 TW08880
1650 LSTOR=IIRGS(MXARGS+2)                         TW08890
IPIC=1                                              TW08900
IND2=ND2+1                                           TW08910
DO 1660 I=1,N                                       TW08920
RC(LSTOR)=A(IND2)                                 TW08930
IND2=IND2+1                                         TW08940
1660 LSTOR=LSTOR+1                               TW08950
IF (IARGS(5)-IARGS(4)-1) 1670,270,560           TW08960
1670 RETURN                                         TW08970
C                                                 TW08980
C                                                 TW08990
1680 FORMAT (//,31X,34H ANALYSIS OF VARIANCE FOR TWO-WAY ,I2,3H X ,I2,6TW09000
1H TABLE,/,4X,7H SOURCE,13X,5H D.F.,10X,14HSUM OF SQUARES,13X,12HMTW09010
2EAN SQUARES,10X,17HF RATIO   F PROB./)          TW09020
1690 FORMAT (4X,20H BETWEEN ROWS      ,I4,50A1,6X,0PF11.3,F9.3)    TW09030
1700 FORMAT (4X,20H BETWEEN COLS     ,I4,50A1,6X,0PF11.3,F9.3)    TW09040
1710 FORMAT (4X,20H RESIDUALS       ,I4,50A1)                      TW09050
1720 FORMAT (4X,20H TOTAL          ,I4,25A1//)                     TW09060
1730 FORMAT (9X,40HA WEIGHTED LEAST SQUARES ANALYSIS USING ,I4,22H NON-TW09070
1ZERO WEIGHTS AND ,I4,24H ZERO WEIGHTS IN COLUMN ,I4/)          TW09080
1740 FORMAT (39X,31HTUKEY'S TEST FOR NON-ADDITIONALITY//,4X,20H NON-ADDITIONALITY ,I4,50A1,6X,0PF11.3,F9.3) TW09090
1750 FORMAT (4X,20H BALANCE        ,I4,50A1)                      TW09100
1760 FORMAT (1X)                                         TW09120
1770 FORMAT (//5X,11HCoefficient,14X,8HEstimate,17X,9HStd. Dev.//5X,10HTW09130
1GRAND MEAN,50A1//)                                TW09140
1780 FORMAT (5X,7HROW      ,I3,50A1)                      TW09150
1790 FORMAT (5X,7HCOLUMN ,I3,50A1)                      TW09160
1800 FORMAT (5X,10HRESIDUAL ,50A1)                      TW09170
1810 FORMAT (//10X,I2,3H X ,I2,86H TABLE OF RESIDUALS, STANDARDIZED BY DIVIDING EACH RESIDUAL BY ITS STANDARD DEVIATION.) TW09180
1820 FORMAT (/,8H COLUMN,15(3X,I4,1X))                TW09200
1830 FORMAT (8H      ROW )                            TW09210
1840 FORMAT (2X,I4,2X,15(2X,F6.2))                  TW09220
END                                               TW09230

```

```

SUBROUTINE VARCON (NAME)           VAR   10
C VERSION 5.00      VARCON      5/15/70    VAR   20
COMMON /BLOCKA/ MODE,M,KARD(83),KARG,ARG,ARG2,NEWCD(80),KRDEND  VAR   30
DIMENSION NAME(2), N(14)          VAR   40
DATA N(1),N(2),N(3),N(4),N(5),N(6),N(7),N(8),N(9),N(10),N(11),N(12) VAR   50
1),N(13),N(14)/10705,2604,16038,16767,17496,18225,18954,1377,15001,VAR   60
25*0/                           VAR   70
C                                     VAR   80
C     LOOKUP NAME IN VARIABLE-NAME TABLE    VAR   90
C                                     VAR  100
C     NAMES IN TABLE                      VAR  110
C                                     VAR  120
C     NRMAX,COLTOP,V,W,X,Y,Z            VAR  130
C                                     VAR  140
DO 10 I=1,7                         VAR  150
IF (NAME(1).EQ.N(I).AND.NAME(2).EQ.N(I+7)) GO TO 20    VAR  160
10 CONTINUE                          VAR  170
I=0                                 VAR  180
20 ARG=I                            VAR  190
RETURN                             VAR  200
END                                VAR  210

SUBROUTINE VECTOR (A,J)             VEC   10
C VERSION 5.00      VECTOR      5/15/70    VEC   20
COMMON /BLOCRC/ NRC,RC(12600)       VEC   30
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NVEC  40
1ARGS,VWXYZ(8),NERROR              VEC   50
DIMENSION ARGS(100)                 VEC   60
EQUIVALENCE (ARGS(1),RC(12501))    VEC   70
C                                     VEC   80
C     VECTORIZE A IN TO COLUMN STARTING AT J    VEC   90
C                                     VEC  100
IF (NRMAX.EQ.0) GO TO 20           VEC  110
K=J+NRMAX-1                        VEC  120
DO 10 I=J,K                         VEC  130
10 RC(I)=A                          VEC  140
20 RETURN                           VEC  150
END                                VEC  160

```

C	SUBROUTINE XECUTE	XEC 10
C	VERSION 5.00 XECUTE 5/15/70	XEC 20
C	COMMON/BLOCKE/ NAME(4),L1,L2,ISRFLG	XEC 30
C	COMMON/BLOCKX/ INDEX(6,8),LEVEL	XEC 40
C *****		XEC 50
C	L1=1-10 FOR COMMANDS CONSISTING OF ONE OR TWO VARIABLES	XEC 60
C	EXAMPLES RESET	XEC 70
C	RESET X	XEC 80
C	PRINT A	XEC 90
C	L1=11-50 FOR COMMANDS CONSISTING OF ONE VARIABLE	XEC 100
C	EXAMPLES ADD	XEC 110
C	MPROP	XEC 120
C	L1=51-63 FOR COMMANDS CONSISTING OF TWO VARIABLES	XEC 130
C	EXAMPLES CLOSE UP	XEC 140
C	M(X'X)	XEC 150
90	GO TO (100,200,300,400,500,600,700,800,900,1000,	XEC 160
	11100,1200,1300,1400,1500,1600,1700,1800,1900,2000,	XEC 170
	22100,2200,2300,2400,2500,2600,2700,2800,2900,3000,	XEC 180
	33100,3200,3300,3400,3500,3600,3700,3800,3900,4000,	XEC 190
	44100,4200,4300,4400,4500,4600,4700,4800,4900,5000,	XEC 200
	55100,5200,5300,5400,5500,5600,5700,5800,5900,6000	XEC 210
	66100,6200,6300),L1	XEC 220
C	RESET	XEC 230
100	CALL RESET	XEC 240
	GO TO 9999	XEC 250
C	PRINT PRINT A-F	XEC 260
200	CALL PRINTX	XEC 270
	GO TO 9999	XEC 280
C	PUNCH	XEC 290
300	CALL PUNCH	XEC 300
	GO TO 9999	XEC 310
C	APRINT APRINT A-F	XEC 320
400	CALL APRINT	XEC 330
	GO TO 9999	XEC 340
C	READ READ A-F	XEC 350
500	CALL READX	XEC 360
	GO TO 9999	XEC 370
C	ABRIDGE	XEC 380
600	CALL ABRIDG	XEC 390
	GO TO 9999	XEC 400
C	MPRINT MPRINT A-F	XEC 410
700	CALL APRINT	XEC 420
C	NPRINT NPRINT A-F	XEC 430
	GO TO 9999	XEC 440
800	CALL PRINTX	XEC 450
	GO TO 9999	XEC 460
C	L1=9 AVAILABLE	XEC 470
900	RETURN	XEC 480
C	L1=10 AVAILABLE	XEC 490
1000	RETURN	XEC 500
C	ADD,SUB,MULT,DIV,RAISE,SUBTRACT,DIVIDE,MULTIPLY	XEC 510
1100	CALL ARITH	XEC 520
	GO TO 9999	XEC 530
C	SIN,ASIN,SIND,ASIND,SINH,ASINH	XEC 540
C	COS,ACOS,COSD,ACOSD,COSH,ACOSH	XEC 550
C	TAN,ATAN,TAND,ATAND,TANH,ATANH,NEGEXP	XEC 560
C	COT,ACOT,COTD,ACOTD,COTH,ACOTH	XEC 570
C	ABS,ABSOLUTE,EXP,EXPONENT,LOG,LOGE,LOGTEN,ANTILOG,SQRT,RAISE	XEC 580
C	INTEGER,FRACTION,SQUARE	XEC 590

1200	CALL FUNCT	XEC 600
	GO TO 9999	XEC 610
C	L1=13	XEC 620
1300	GO TO (1301,1302,1303,1304,1305,1306,1307,1308,1309,1310, 11311,1312,1313,1314),L2	XEC 630
C	GENERATE	XEC 640
1301	CALL GENER	XEC 650
	GO TO 9999	XEC 660
C	SET	XEC 670
1302	CALL SET	XEC 680
	GO TO 9999	XEC 690
C	FIXED	XEC 700
1303	CALL FIXFL0	XEC 710
	GO TO 9999	XEC 720
C	FLOATING	XEC 730
1304	GO TO 1303	XEC 740
C	PLOT	XEC 750
1305	CALL PLOT	XEC 760
	GO TO 9999	XEC 770
C	PAGE PLOT	XEC 780
1306	CALL PLOT	XEC 790
	GO TO 9999	XEC 800
C	L1=13 L2=7 AVAILABLE	XEC 810
1307	RETURN	XEC 820
C	NEW PAGE	XEC 830
1308	CALL PAGE(4)	XEC 840
	GO TO 9999	XEC 850
C	SPACE	XEC 860
1309	CALL SPACE	XEC 870
	GO TO 9999	XEC 880
C	CGS	XEC 890
1310	CALL PHYCON(0)	XEC 900
	GO TO 9999	XEC 910
C	SI	XEC 920
1311	CALL PHYCON(-1)	XEC 930
	GO TO 9999	XEC 940
C	FLEXIBLE	XEC 950
1312	GO TO 1303	XEC 960
C	PRINT NOTE	XEC 970
1313	CALL NOTEPR(3)	XEC 980
	GO TO 9999	XEC 990
C	ROUND	XEC1000
1314	CALL FNEIC	XEC1010
	GO TO 9999	XEC1020
C	L1=14	XEC1030
1400	GO TO(1401,1402,1403,1404,1405,1406,1407,1408,1409,1410,1411,1412,1413,1414,1415),L2	XEC1040
C	BEGIN	XEC1050
1401	CALL BEGIN	XEC1060
	GO TO 9999	XEC1070
C	SCAN	XEC1080
1402	GO TO 1401	XEC1090
C	REPEAT EXECUTE PERFORM	XEC1100
1403	J=1	XEC1110
	CALL REPIINC(J)	XEC1120
	IF(J)9999,9999,90	XEC1130
C	L1=14 L2= 4 AVAILABLE	XEC1140
1404	RETURN	XEC1150
C	L1=14 L2=5 AVAILABLE	XEC1160
		XEC1170
		XEC1180

1405	RETURN	XEC1190
C	INCREMENT	XEC1200
1406	J=3	XEC1210
	CALL REPINC(J)	XEC1220
	GO TO 9999	XEC1230
C	L1=14 L2=7 AVAILABLE	XEC1240
1407	RETURN	XEC1250
C	RESTORE	XEC1260
1408	J=3	XEC1270
	CALL REPINC(J)	XEC1280
	GO TO 9999	XEC1290
C	IFLT	XEC1300
1409	CALL IFS	XEC1310
	GO TO 9999	XEC1320
C	IFEQ	XEC1330
1410	GO TO 1409	XEC1340
C	IFGT	XEC1350
1411	GO TO 1409	XEC1360
C	IFGE	XEC1370
1412	GO TO 1409	XEC1380
C	IFNE	XEC1390
1413	GO TO 1409	XEC1400
C	IFLE	XEC1410
1414	GO TO 1409	XEC1420
C	COMPARE	XEC1430
1415	GO TO 1409	XEC1440
C	MDEFINE MZERO MERASE MIDENT MDIAGONAL	XEC1450
C	ADEFINE AZERO AERASE	XEC1460
1500	CALL MOP	XEC1470
	GO TO 9999	XEC1480
C	MINVERT INVERT SOLVE	XEC1490
1600	CALL INVERT	XEC1500
	GO TO 9999	XEC1510
C	L1=17 NO L2 VALUES ARE NEEDED IN SUBROUTINES	XEC1520
1700	GO TO (1701,1702,1703,1704,1705),L2	XEC1530
C	MMULT MMULTIPLY	XEC1540
1701	CALL MMULT	XEC1550
	GO TO 9999	XEC1560
C	MRAISE	XEC1570
1702	CALL MRAISE	XEC1580
	GO TO 9999	XEC1590
C	MKRONECKER	XEC1600
1703	CALL MKRON	XEC1610
	GO TO 9999	XEC1620
C	MTRAIN	XEC1630
1704	CALL MTRIAN	XEC1640
	GO TO 9999	XEC1650
C	MEIGEN	XEC1660
1705	CALL MEIGEN	XEC1670
	GO TO 9999	XEC1680
1800	IF (L2.GT.8) GO TO 1809	XEC1690
C	MADD MSUB MTRANS SCALAR AMULT MSUBTRACT	XEC1700
C	AADD ASUB ATRANS ARAISE ADIVIDE ASUBTRACT AMULTIPLY	XEC1710
	CALL MATRIX	XEC1720
	GO TO 9999	XEC1730
C	ACOALES AAVERA	XEC1740
1809	CALL COALES	XEC1750
	GO TO 9999	XEC1760
C	NORMLAGUERE LAGUERE HERMITE LEGENDRE TCHEBYSHEV UCHEBYSHEV	XEC1770

1900	CALL ALLSUB	XEC1780
	GO TO 9999	XEC1790
C	PARSUM PARPRODUCT RMS AVERAGE SUM	XEC1800
2000	CALL MSCROW	XEC1810
	GO TO 9999	XEC1820
C	L1=21	XEC1830
2100	GO TO (2101,2102,2103,2104,2105,2106,2107,2108,2109,2110,2111, 12112,2113,2114,2115,2116,2117,2118,2119),L2	XEC1840
C	ROWSUM ROW SUM	XEC1850
2101	CALL PROROW	XEC1860
	GO TO 9999	XEC1870
C	PRODUCT	XEC1880
2102	GO TO 2101	XEC1900
C	DEFINE	XEC1910
2103	CALL DEFINE	XEC1920
	GO TO 9999	XEC1930
C	L1=21 L2=4 AVAILABLE	XEC1940
2104	RETURN	XEC1950
C	MAX MAXIMUM	XEC1960
2105	CALL EXTREM	XEC1970
	GO TO 9999	XEC1980
C	MIN MINIMUM	XEC1990
2106	GO TO 2105	XEC2000
C	L1=21 L2=7 AVAILABLE	XEC2010
2107	RETURN	XEC2020
C	SQRT	XEC2030
2108	CALL SORDER	XEC2040
	GOTO 9999	XEC2050
C	ORDER	XEC2060
2109	GO TO 2108	XEC2070
C	ERASE	XEC2080
2110	CALL ERASE	XEC2090
	GO TO 9999	XEC2100
C	EXCHANGE	XEC2110
2111	CALL EXCHNG	XEC2120
	GO TO 9999	XEC2130
C	FLIP	XEC2140
2112	CALL FLIP	XEC2150
	GO TO 9999	XEC2160
C	CHANGE	XEC2170
2113	CALL CHANGE	XEC2180
	GO TO 9999	XEC2190
C	HIERARCHY	XEC2200
2114	GO TO 2108	XEC2210
C	LIST	XEC2220
C	WRITE SUBROUTINE FOR LIST	XEC2230
2115	CALL LIST(0)	XEC2240
	GO TO 9999	XEC2250
C	NO LIST	XEC2260
C	WRITE SUBROUTINE	XEC2270
2116	CALL LIST(1)	XEC2280
	GO TO 9999	XEC2290
C	NULL	XEC2300
2117	GO TO 9999	XEC2310
C	ERROR	XEC2320
2118	CALL FNEC	XEC2330
	GO TO 9999	XEC2340
C	CERF	XEC2350
2119	GO TO 2118	XEC2360

C	POLYFIT FIT MORTHO	XEC2370
C	SPOLYFIT SFIT	XEC2380
2200	CALL ORTHO	XEC2390
	GO TO 9999	XEC2400
C	L1=23	XEC2410
2300	GO TO (2301,2302,2303,2304,2305,2306,2307,2308,2309,2310,2311, 12312),L2	XEC2420
C	CLOSE UP	XEC2430
2301	CALL MISC2	XEC2440
	GO TO 9999	XEC2450
C	COUNT	XEC2460
2302	GO TO 2301	XEC2470
C	SHORTEN	XEC2480
2303	GO TO 2301	XEC2490
C	EXPAND	XEC2500
2304	GO TO 2301	XEC2510
C	DUPLICATE	XEC2520
2305	GO TO 2301	XEC2530
C	MOVE AMOVE MMOVE	XEC2540
2306	CALL MOVE	XEC2550
	GO TO 9999	XEC2560
C	L1=23 L2=7 AVAILABLE	XEC2570
2307	RETURN	XEC2580
C	L1=23 L2=8 AVAILABLE	XEC2590
2308	RETURN	XEC2600
C	L1=23 L2=9 AVAILABLE	XEC2610
2309	RETURN	XEC2620
C	PROMOTE	XEC2630
2310	CALL PDMOTE	XEC2640
	GO TO 9999	XEC2650
C	DEMOTE	XEC2660
2311	GO TO 2310	XEC2670
C	DIMENSION DIM	XEC2680
2312	CALL DIMENS	XEC2690
	GO TO 9999	XEC2700
C	L1=24	XEC2710
2400	GO TO (2401,2402,2403,2404,2405,2406,2407,2408,2409,2410, 1 2411,2412,2413,2414,2415),L2	XEC2720
C	STATIS	XEC2730
2401	CALL STATIS	XEC2740
	GO TO 9999	XEC2750
C	SSTATIS	XEC2760
2402	GO TO 2401	XEC2770
C	RANKS	XEC2780
2403	CALL RANKS	XEC2790
	GO TO 9999	XEC2800
C	GAUSS QUADRATURE	XEC2810
2404	CALL GQUAD	XEC2820
	GO TO 9999	XEC2830
C	F PROBABILITY	XEC2840
2405	CALL FPROB	XEC2850
	GO TO 9999	XEC2860
C	TWOWAY	XEC2870
2406	CALL TWOWAY(L2)	XEC2880
	GO TO 9999	XEC2890
C	STTWOWAY	XEC2900
2407	CALL TWOWAY(L2)	XEC2910
	GO TO 9999	XEC2920
C	HISTOG	XEC2930
		XEC2940
		XEC2950

2408	CALL HISTGM	XEC2960
	GO TO 9999	XEC2970
C	NHISTO	XEC2980
2409	CALL HISTGM	XEC2990
	GO TO 9999	XEC3000
C	FREQUE	XEC3010
2410	CALL FRDIST	XEC3020
	GO TO 9999	XEC3030
C	CORREL	XEC3040
2411	CALL CORREL	XEC3050
	GO TO 9999	XEC3060
C	SCORRE	XEC3070
2412	CALL CORREL	XEC3080
	GO TO 9999	XEC3090
C	ONEWAY	XEC3100
2413	CALL ONEWAY	XEC3110
	GO TO 9999	XEC3120
C	SONEWAY	XEC3130
2414	GO TO 2413	XEC3140
C	UNIFORM RANDOM	XEC3150
2415	CALL FNKC	XEC3160
	GO TO 9999	XEC3170
C	SELECT SEARCH CENSOR MATCH	XEC3180
2500	IF (L2.EQ.4) GO TO 2504	XEC3190
	CALL SELECT	XEC3200
	GO TO 9999	XEC3210
2504	CALL INTERP	XEC3220
	GO TO 9999	XEC3230
C	MVECDIAG MVECMMAT MMATVEC MVECDIAGONAL	XEC3240
2600	CALL EXPCON	XEC3250
	GO TO 9999	XEC3260
C	MPROPERTIES APROPERTIES SMPROP SAPROP	XEC3270
2700	CALL MPROP	XEC3280
	GO TO 9999	XEC3290
C	ITERATE ISETUP ISOLATE	XEC3300
2800	CALL ITERAT	XEC3310
	GO TO 9999	XEC3320
C	SEPARATE INSERT MAXMIN EXTREMA	XEC3330
2900	CALL CMSEPA	XEC3340
	GO TO 9999	XEC3350
C	BESSEL SUBROUTINES	XEC3360
3000	CALL BESSEL	XEC3370
	GO TO 9999	XEC3380
C	THERMO	XEC3390
3100	CALL THERMO	XEC3400
	GO TO 9999	XEC3410
C	COMPLEX ARITHMETIC	XEC3420
3200	CALL COMPLX	XEC3430
	GO TO 9999	XEC3440
C	L1=33	XEC3450
C	L1=34	XEC3460
C	L1=35	XEC3470
C	L1=36	XEC3480
C	L1=37	XEC3490
C	L1=38	XEC3500
C	L1=39	XEC3510
C	L1=40	XEC3520
C	L1=41	XEC3530
C	L1=42	XEC3540

C	L1=43	XEC3550
C	L1=44	XEC3560
3300	RETURN	XEC3570
3400	RETURN	XEC3580
3500	RETURN	XEC3590
3600	RETURN	XEC3600
3700	RETURN	XEC3610
3800	RETURN	XEC3620
3900	RETURN	XEC3630
4000	RETURN	XEC3640
4100	RETURN	XEC3650
4200	RETURN	XEC3660
4300	RETURN	XEC3670
4400	RETURN	XEC3680
C*****	THE FOLLOWING CARDS ARE NEEDED ONLY FOR TAPE OPERATIONS	XEC3690
C	THE STATEMENTS 4500 -5000 WERE RETURN STATEMENTS	XEC3700
C	READ TAPE	XEC3710
4500	CALL TAPOP2	XEC3720
	GO TO 9999	XEC3730
C	CREAD TAPE	XEC3740
4600	GO TO 4500	XEC3750
C	WRITE TAPE	XEC3760
4700	GO TO 4500	XEC3770
C	SET TAPE	XEC3780
4800	GO TO 4500	XEC3790
C	CSET TAPE	XEC3800
4900	GO TO 4500	XEC3810
C	ENDFILE TAPE,REWIND TAPE SKIP TAPE, BACKSPACE TAPE	XEC3820
5000	GO TO 4500	XEC3830
C*****	*****	XEC3840
C	M(XX') M(X'X) M(XAX') M(X'AX)	XEC3850
5100	CALL MXTX	XEC3860
	GO TO 9999	XEC3870
C	M(AD),M(DA)	XEC3880
5200	CALL MDAMAD	XEC3890
	GO TO 9999	XEC3900
C	M(V'A) M(AV)	XEC3910
5300	CALL ARYVEC	XEC3920
	GO TO 9999	XEC3930
C	L1=54	XEC3940
5400	GO TO (5401,5402,5403,5404,5405,5406,5407),L2	XEC3950
C	L2=1	XEC3960
5401	RETURN	XEC3970
C	DUMMY A	XEC3990
5402	CALL DUMMYA	XEC4000
	GO TO 9999	XEC4010
C	DUMMY B	XEC4020
5403	CALL DUMMYB	XEC4030
	GO TO 9999	XEC4040
C	DUMMY C	XEC4050
5404	CALL DUMMYC	XEC4060
	GO TO 9999	XEC4070
C	DUMMY D	XEC4080
5405	CALL DUMMYD	XEC4090
	GO TO 9999	XEC4100
C	DUMMY E	XEC4110
5406	CALL DUMMYE	XEC4120
	GO TO 9999	XEC4130
C	DUMMY F	XEC4140

5407	CALL DUMMYF	XEC4150
	GO TO 9999	XEC4160
C	L1=55	XEC4170
C	L1=56	XEC4180
C	L1=57	XEC4190
C	L1=58	XEC4200
C	L1=59	XEC4210
C	L1=60	XEC4220
C	L1=61	XEC4230
C	L1=62	XEC4240
C	L1=63	XEC4250
5500	RETURN	XEC4260
5600	RETURN	XEC4270
5700	RETURN	XEC4280
5800	RETURN	XEC4290
5900	RETURN	XEC4300
6000	RETURN	XEC4310
6100	RETURN	XEC4320
6200	RETURN	XEC4330
6300	RETURN	XEC4340
9999	CALL AERR(0)	XEC4350
	IF (LEVEL.LE.0) RETURN	XEC4360
	J=2	XEC4370
	CALL REPINC(J)	XEC4380
	IF (LEVEL.LE.0) RETURN	XEC4390
	GO TO 90	XEC4400
	END	XEC4410

	SUBROUTINE XFORMAT	XFO	10
C	VERSION 5.00 XFORMAT 5/15/70	XFO	20
C	COMMON /BLOCKA/ MODE,M,KARD(83),KARG,ARG,ARG2,NEWCD(80),KRDEND	XFO	30
C	LOOK FOR LETTER A-F FOLLOWED BY NON-ALPHANUMERIC CHARACTER	XFO	40
C	A \$ = 46 STOPS THE SCAN	XFO	60
10	M=M+1	XFO	70
	IF (KARD(M).LT.10.OR.KARD(M).GT.15) IF(KARD(M)-46) 10,20,10	XFO	75
	IF (KARD(M+1).LE.35) GO TO 20	XFO	80
C	CALL PREPAK TO STORE FORMAT	XFO	90
C	IF IND=0 FORMAT IS O.K. AND STORED	XFO	100
C	IF IND=1 NUMBER OF ( DOES NOT EQUAL THE NUMBER OF )	XFO	110
C	CALL PREPAK (1,IND,IR,IR,IR)	XFO	120
	IF (IND.EQ.0) RETURN	XFO	130
20	CALL ERROR (205)	XFO	140
	RETURN	XFO	150
	END	XFO	160
		XFO	170

	SUBROUTINE XHEAD	XHE	10
C	VERSION 5.00 XHEAD 5/15/70	XHE	20
C	COMMON /BLOCKA/ MODE,M,KARD(83),KARG,ARG,ARG2,NEWCD(80),KRDEND	XHE	30
C	COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NXHE	XHE	40
	IARGS,VWXYZ(8),NERROR	XHE	50
C	COMMON /ABCDEF/ L(48)	XHE	60
	GO TO 20	XHE	70
10	M=M+1	XHE	80
20	IF (KARD(M).GE.10) IF (KARD(M)-46) 10,30,10	XHE	85
	CALL AARGS	XHE	90
	I=ARG	XHE	100
	IF (KARG.EQ.0.AND.I.GT.0.AND.I.LE.NCOL) GO TO 60	XHE	110
30	CALL ERROR (204)	XHE	120
	RETURN	XHE	130
50	M=M+1	XHE	140
60	IF (KARD(M).NE.36) IF (KARD(M)-46) 50,30,50	XHE	145
C		XHE	150
C	SLASH FOUND. PICK UP NEXT 12 CHARACTERS IN FORMAT A1 AND PACK	XHE	160
C		XHE	170
80	CALL PREPAK (2,IND,I,IR,IR)	XHE	230
	RETURN	XHE	240
	END	XHE	250

```

SUBROUTINE XOMNIT (LG) XOM 10
C VERSION 5.00 XOMNIT 5/15/70 XOM 20
C ADD A SWITCH TO DETERMINE WHETHER PRINTX OR RPRINT WILL BE USED XOM 21
C INITIALIZE SWITCH TO ZERO FLEXIBLE FORMAT(RPRINT) WILL BE USED XOM 22
C SWITCH WILL BE SET TO 1 IF FIXED OR FLOATING IS ENCOUNTERED XOM 23
C IF PRINT COMMAND HAS DEC. ARGS FLEXIBLE FORMAT WILL BE USED XOM 24
C SWITCH WILL NOT BE CHANGED XOM 25
C FLEXIBLE COMMAND WILL CHANGE SWITCH TO 0 XOM 26
C IF PRINT WITH ALL INTEGER ARGS AND SWITCH=0 USE FLEXIBLE FORMAT XOM 27
C IF PRINT WITH ALL INTEGER ARGS AND SWITCH=1, USE SPECIFIED FORMAT XOM 28
C (FIXED OR FLOATING) XOM 29
COMMON /BLOCKA/ MODE,M,KARD(83),KARG,ARG,ARG2,NEWCD(80),KRDEND XOM 30
COMMON /BLOCKB/ NSTMT,NSTMTH,NCOM,LCOM,IOVFL,COM(2000) XOM 40
COMMON /BLOCKC/ KIO,INUNIT,ISCRAT,KBDOUT,KRDKNL,LLIST XOM 50
COMMON /BLOCRC/ NRC,RC(12600) XOM 60
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NXOM 70
1ARGS,VWXYZ(8),NERROR XOM 80
DIMENSION ARGS(100) XOM 90
EQUIVALENCE (ARGS(1),RC(12501)) XOM 100
COMMON /HEADER/ NOCARD(80),ITLE(60,6),LNCNT,IPRINT,NPAGE,IPUNCH XOM 110
COMMON/PKSWT/IHCNT,IHTP XOM 115
COMMON/FMAT/IFMTX(6),IOSWT,IFMTS(6),LHEAD(96) XOM 120
COMMON/BLOCKX/INDEX(6,8),LEVEL XOM 125
COMMON /SPRV/ NERCON,NERR,ISWERR XOM 130
COMMON/ABCDEF/L(48) XOM 140
COMMON/PCONST/JPC,P(40),N(40) XOM 150
C XOM 170
C IF LG IS NEG, FIRST CARD WAS NOT 'OMNITAB' CARD. IF LG= 0, FIRST XOM 180
C CARD = ,OMNITAB,, ELSE SUBSEQUENT 'OMNITAB' CARD FOUND. XOM 190
C XOM 200
IF (LG) 40,20,10 XOM 210
C XOM 220
C GO THROUGH 'STOP' SEQUENCE AND RETURN XOM 230
10 CALL XSTOP XOM 240
20 DO 30 I=1,80 XOM 250
30 NOCARD(I)=NEWCD(I) XOM 260
C XOM 270
C INITIALIZE SYSTEM XOM 280
C XOM 290
40 DO 50 I=1,60 XOM 300
DO 50 J=1,6 XOM 310
50 ITLE(I,J)=L(45) XOM 320
DO 60 I=1,6 XOM 330
60 IFMTX(I)=IFMTS(I) XOM 340
IOSWT=0 XOM 350
IHCNT=0 XOM 360
CALL PREPAK(3,IND,IND,IND,IND) XOM 370
MODE=1 XOM 400
NRMAX=0 XOM 410
NROW=201 XOM 420
NCOL=62 XOM 430
KRDEND=80 XOM 440
LLIST=3 XOM 450
NERROR=0 XOM 460
NSTMT=0 XOM 470
NSTMTH=0 XOM 480
NCOM=1 XOM 490
LCOM=2000 XOM 500
LEVEL=0 XOM 505

```

	I0VFL=0	XOM 510
	NPAGE=0	XOM 520
	NRCC=NRC+100	XOM 530
	DO 90 I=1, NRCC	XOM 540
90	RC(I)=0.	XOM 550
	NERR=0	XOM 560
	NERCON=100	XOM 570
	ISWERR=0	XOM 580
	DO 100 I=1, 8	XOM 584
100	VWXYZ(I)=0.0	XOM 586
	JPC=-1	XOM 588
	CALL NOTEPR (0)	XOM 600
	RETURN	XOM 605
	END	XOM 610

```

SUBROUTINE XPND (T,K,Y,KND) XPN 10
C VERSION 5.00 XPND 5/15/70 XPN 20
COMMON /BLOCRC/ NRC,RC(12600) XPN 30
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NXPN 40
IARGS,VWXYZ(8),NERROR XPN 50
DIMENSION ARGS(100) XPN 60
EQUIVALENCE (ARGS(1),RC(12501)) XPN 70
DIMENSION T(2) XPN 80
C XPN 90
C THIS SUBROUTINE TAKES A ''STATEMENT'' REFERENCE AS STORED XPN 100
C AND EXPANDS IT INTO THE PROPER ARGUMENT WITH CHECKING. XPN 110
C XPN 120
C K IS RETURNED 0 IF ARG IN STATEMENT IS ONE WORD LONG XPN 130
C K IS RETURNED 1 IF ARG IN STATEMENT IS TWO WORDS LONG. XPN 140
C K IS RETURNED -( ERROR NUMBER ) IF ERROR OCCURS. XPN 150
C XPN 160
IT=-T(1) XPN 170
IF (IT.LT.16) GO TO 40 XPN 180
C XPN 190
C ''ROW, COL'' ENTRY XPN 200
C XPN 210
IT=IT-8208 XPN 220
IF (IT.GT.0.AND.IT.LE.NROW) GO TO 10 XPN 230
K=-16 XPN 240
GO TO 20 XPN 250
10 IARGS(100)=ABS(T(2))-8192. XPN 260
KIND(100)=0 XPN 270
CALL ADRESS (100,J) XPN 280
IF (J.NE.0) GO TO 30 XPN 290
K=-11 XPN 300
20 RETURN XPN 310
30 J=J+IT XPN 320
KND=0 XPN 330
IF (T(2).LT.0.) KND=1 XPN 340
Y=RC(J-1) XPN 350
K=1 XPN 360
GO TO 20 XPN 370
C XPN 380
C NRMAX, V, W, X, Y, Z, REFERENCE. XPN 390
C XPN 400
40 IU=IT/2 XPN 410
KND=IT-2*IU XPN 420
K=0 XPN 430
IF (IU.LE.1) GO TO 50 XPN 440
Y=VWXYZ(IU-2) XPN 450
GO TO 20 XPN 460
50 Y=NRMAX XPN 470
GO TO 20 XPN 480
END XPN 490

```

```

SUBROUTINE XSTOP                               XST 10
C      VERSION 5.00    XSTOP      5/15/70          XST 20
COMMON /BLOCKC/ KIO,INUNIT,ISCRAT,KBDOUT,KRDKNT,LLIST   XST 30
COMMON /BLOCRC/ NRC,RC(12600)                  XST 40
COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NROW,NCOL,NXST 50
1ARGS,VWXYZ(8),NERROR                      XST 60
DIMENSION ARGS(100)                         XST 70
EQUIVALENCE (ARGS(1),RC(12501))            XST 80
COMMON/HEADER/NOCARD(80),ITLE(60,6),LNCNT,IPRINT,NPAGE,IPUNCH XST 90
COMMON /SCRAT/ NS,NS2,A(13500)              XST 110
DIMENSION ITEMP(84)                         XST 120
EQUIVALENCE (ITEMP(1),A(1))                XST 130
DATA IZ,IP,NO,KOMMA/1HZ,1H+,1H0,1H,/        XST 140
                                         XST 150
C      THIS ROUTINE REWINDS THE SCRATCH UNIT AND PRINTS IT.  XST 160
C
REWIND ISCRAT                                XST 170
LLIST=0                                         XST 180
IF (NERROR.EQ.0) LLIST=3                     XST 190
10     CALL PAGE (0)                         XST 200
       WRITE (IPRINT,90)                      XST 210
       DO 40 J=1,50                           XST 220
       READ (ISCRAT,100) ITEMP               XST 230
       IF (ITEMP(1).EQ.IZ) GO TO 50          XST 240
       IF (ITEMP(1).EQ.IP) GO TO 30          XST 250
       IF (ITEMP(1).EQ.KOMMA) GO TO 20         XST 260
       WRITE (IPRINT,110) ITEMP               XST 270
       GO TO 40                             XST 280
20     LLIST=3                           XST 290
       IF (ITEMP(2).EQ.NO.AND.NERROR.EQ.0) LLIST=0 XST 300
       GO TO 40                           XST 310
30     WRITE (IPRINT,120) (ITEMP(I),I=2,84)  XST 320
40     CONTINUE                           XST 330
       GO TO 10                           XST 340
50     REWIND ISCRAT                     XST 350
       IF (NERROR-1) 80,60,70             XST 360
60     WRITE (IPRINT,130)                 XST 370
       GO TO 80                           XST 380
70     WRITE (IPRINT,140) NERROR           XST 390
80     LLIST=3                           XST 400
       WRITE (IPRINT,150)                 XST 410
       WRITE (IPRINT,160)                 XST 420
       RETURN                            XST 430
                                         XST 440
C
90     FORMAT (//19X,39H LIST OF COMMANDS, DATA AND DIAGNOSTICS//) XST 450
100    FORMAT (84A1)                        XST 460
110    FORMAT (20X,84A1)                    XST 470
120    FORMAT (18X,3A1,3X,80A1)            XST 480
130    FORMAT (///40X,20H ONLY ONE FATAL ERROR) XST 490
140    FORMAT (///40X,I4,7H ERRORS)        XST 500
150    FORMAT(1H0/33X,95H NATIONAL BUREAU OF STANDARDS, WASHINGTON, D. C.XST 510
1      20234, OMNITAB II VERSION 5.00 MAY 15, 1970 )          XST 520
160    FORMAT(1H1)                          XST 530
       END                                XST 690
                                         XST 700

```

## 8. Appendix – OMNIT Flow Chart

### Flow Chart Notes

(1) The flow chart of the OMNIT subprogram was generated by the FLOGEN program on the UNIVAC 1108 and drawn by the CAL-COMP plotter; see CAL-COMP [1968].

(2) When the flow of the diagram is broken, an arrow points to a box with two symbols. The top number refers to the page and the bottom symbol is the entry point. At each entry arrow, the number inside the box indicates the page from which the flow comes and the symbol outside is the entry position.

(3) The INPUT subprogram reads an OMNITAB instruction and stores the information in the array KARD. Each character is stored in one location starting with the character of the first column in KARD(3). See page 2 of the flow chart.

(4) The subprogram NNAME takes the first six or less characters of the first nonnumeric word and converts them into two numbers by combining the values assigned to each letter. (See NNAME for the values assigned to each letter.) The numbers are

stored in NAME(1) and NAME(2). See page 5 of the flow chart.

(5) Pages 5 through 11 of the flow chart check to see if the OMNITAB command is one of the following: OMNITAB, FINISH, FORMAT, NOTE, HEAD, TITLE or STOP.

(6) The subprogram EXPAND converts the values in ARGTAB, described on page 18 of the flow chart, to floating-point numbers and stores them in either the array IARGS or the array ARGS, depending upon whether an argument is an integer or a floating-point number. Also, EXPAND sets KIND(I)=0, if an argument is an integer, and sets KIND(I)=1, if an argument is a floating-point number; see pages 19 and 22 of the flow chart.

(7) A table look-up is done by the subprogram LOOKUP using NAME(1), NAME(2) and sometimes NAME(3) and NAME(4). Also, a set of unique numbers are assigned to the variables L1 and L2. If L1=0, no command was found; see page 20 of the flow chart.

(8) If a command is found, the subprogram XECUTE is called to execute the instruction; see page 22 of the flow chart.

SUBROUTINE OMNIT

C VERSION 5.00 OMNIT 5/15/70  
 C \*\*\*\* THIS IS THE MAIN OMNITAB ROUTINE \*\*\*\*

COMMON /BLOCKA/ MODE,M,KARD(83),KARG,ARG,ARG2,NEWCD(80),KRDEND  
 COMMON /BLOCKB/ NSTMT,NSTMTX,NSTMTH,NCOM,LCOM,I0VFL,COM(2000)  
 COMMON /BLOCKC/ KIO,INUNIT,ISCRAT,KBDOUT,KRDKNT,LLIST  
 COMMON /BLOCKD/ NRC,RC(12600)

COMMON /BLOCKD/ IARGS(100),KIND(100),ARGTAB(100),NRMAX,NRROW,NCOL,N  
 ARGS,VWXYZ(8),NERROR

DIMENSION ARGS(100)  
 EQUIVALENCE (ARGS(1),RC(12501))  
 COMMON /BLOCKE/ NAME(4),L1,L2,ISRFGL  
 COMMON/HEADER/NOCARD(80),ITLE(60,6),LNCNT,IPRINT,NPAGE,IPUNCH

C THE FOLLOWING CARDS ARE NEEDED ONLY FOR TAPE OPERATIONS

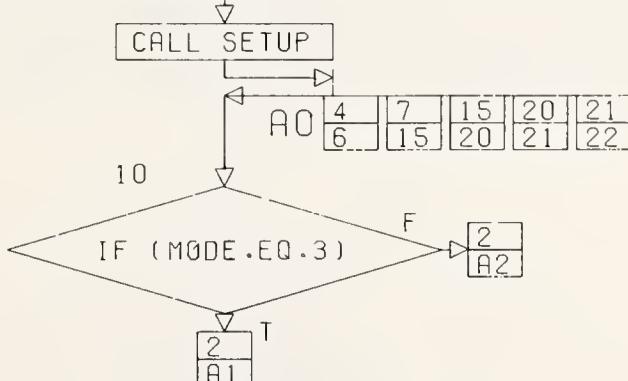
COMMON /TAPE/ NAME4(2),NTPCT,IPUNCP,INUNIP,L1TP

C \*\*\*\*

DATA IBLANK/1H/.LETSG0/-1/

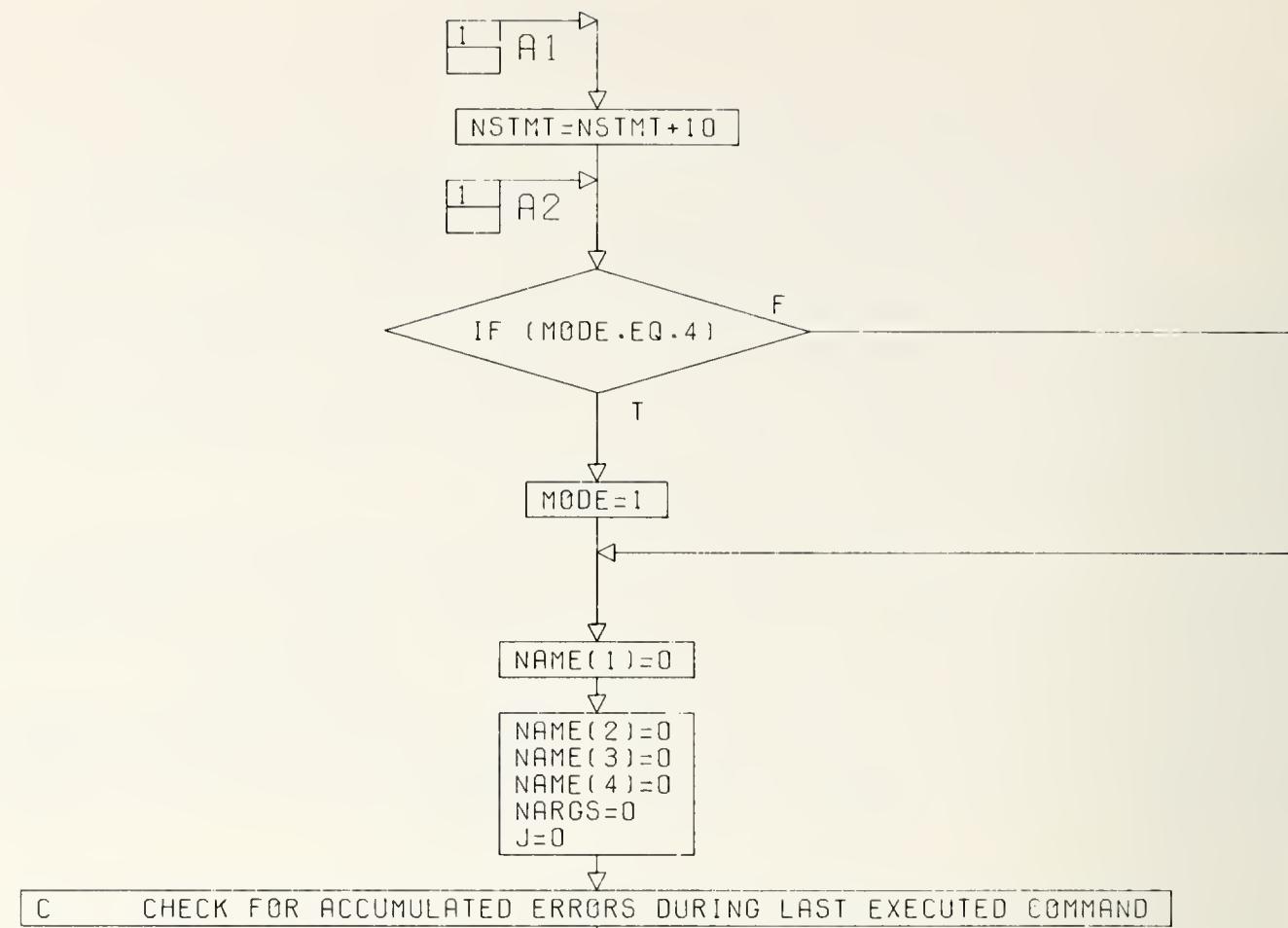
C THIS IS THE MAIN OMNITAB PROGRAM  
 C SUBROUTINES CALLED BY THIS PROGRAM..  
 C SETUP,INPUT,ERROR,STMT,NNAME,AARGS,ASTER,SETQ,READQ,STORE,XECUTE  
 C AERR,XOMNIT,XF0RMT,LOOKUP  
 C MOD = 1 INTERPRETIVE MODE  
 C = 2 DATA MODE (READ SET)  
 C = 3 STORAGE MODE (BETWEEN BEGIN AND FINISH)  
 C = 4 IMPLIED STORAGE MODE (STATEMENT NUMBER GIVEN)

C 0 = 0, 1 = 1, ETC., 9 = 9, A = 10, B = 11, ETC., Z = 35, / = 36  
 C . = 37, - = 38, + = 39, \* = 40, ( = 41, ) = 42, , = 43  
 C BLANK = 44, = = 45, \$ AND OTHERS = 46

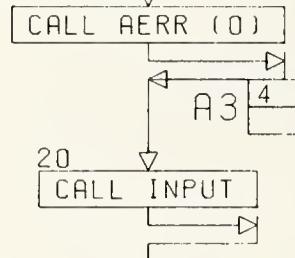


CONT. ON PG 2

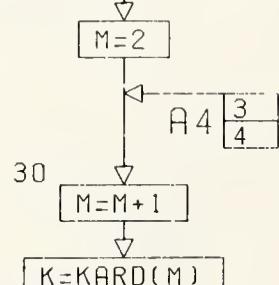
PG. 1 OF 22



C CHECK FOR ACCUMULATED ERRORS DURING LAST EXECUTED COMMAND

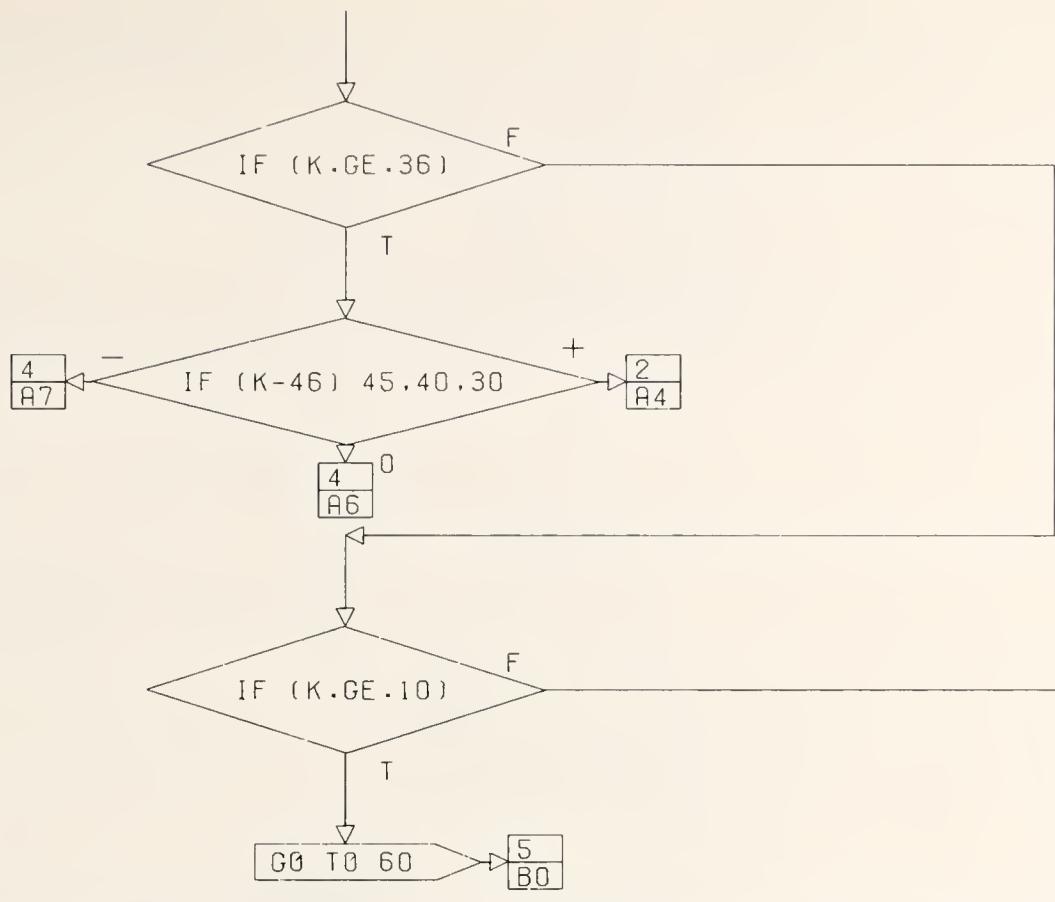


C SCANNING BEGINS WITH THE THIRD CHARACTER. THE FIRST TWO ARE DUMMY  
TO KEEP THE PROGRAM OUT OF TROUBLE. SCANNING TERMINATES WITH A \$  
A \$ HAS BEEN PLANTED IN THE (KRDEND+1)-TH POSITION.

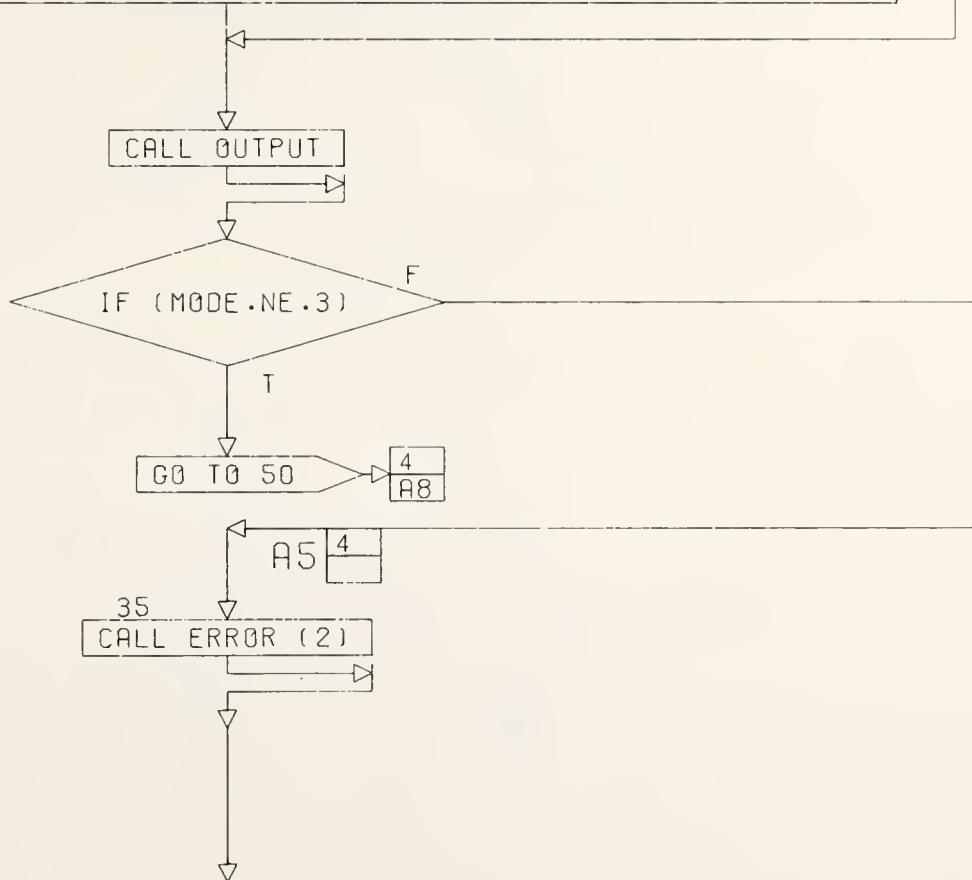


CONT. ON PG 3

PG 2 OF 22

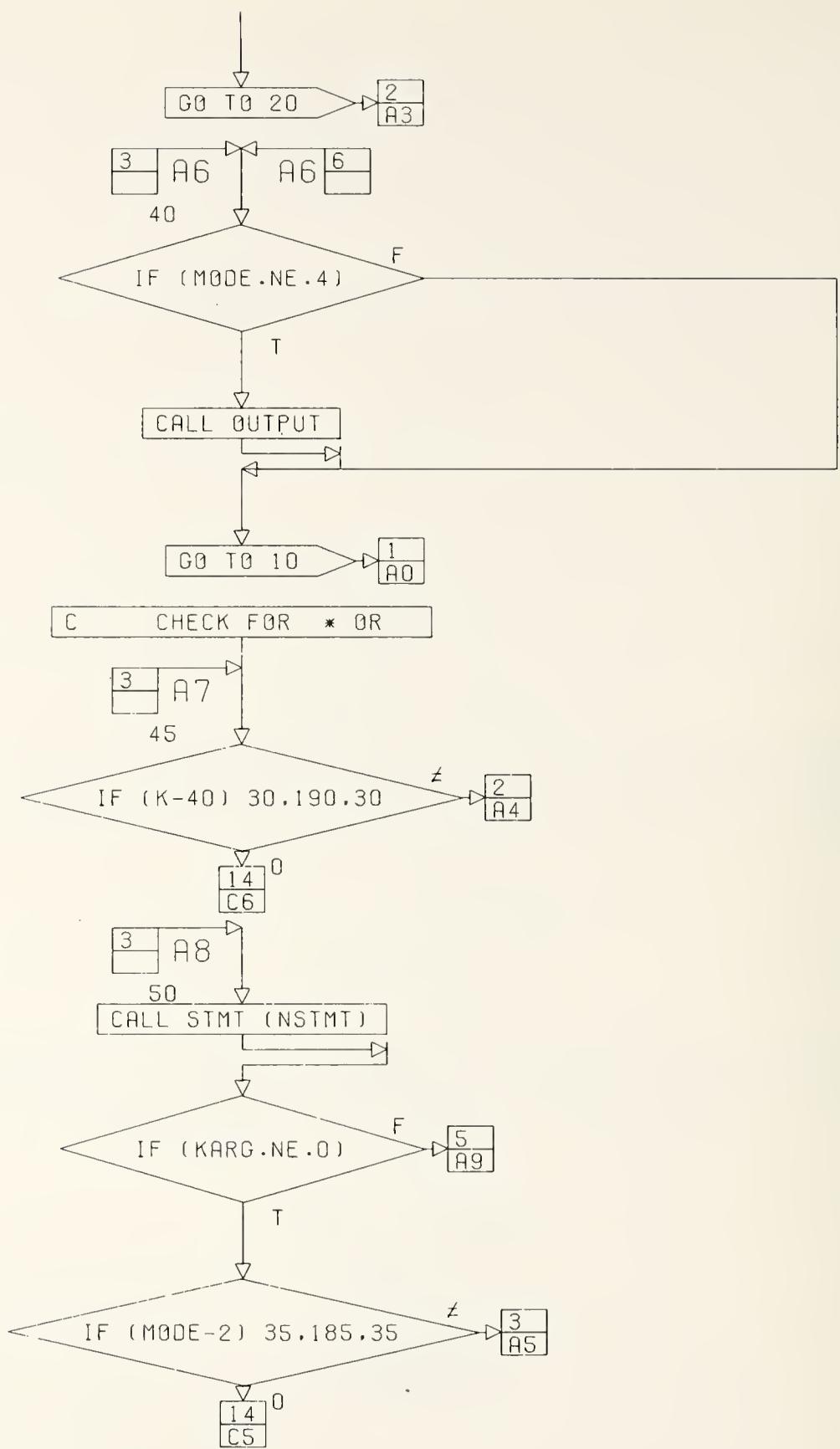


C      A NUMBER IS THE FIRST ALPHANUMERIC CHARACTER ENCOUNTERED, ERROR IF  
 C      IN MODE 3



CONT. ON PG      4

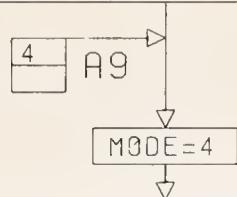
PG 3 OF 22



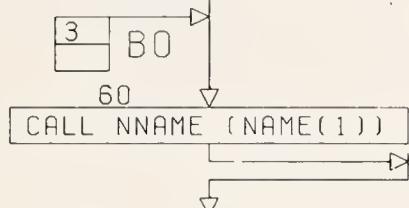
CONT. ON PG 5

PG 4 OF 22

C IF AN ILLEGAL STATEMENT NUMBER WAS FOUND, KARG = 1 (KARG = 0 IF  
C LEGAL)



C M IS POINTING AT THE FIRST LETTER ON THE CARD, ASSEMBLE NAME.



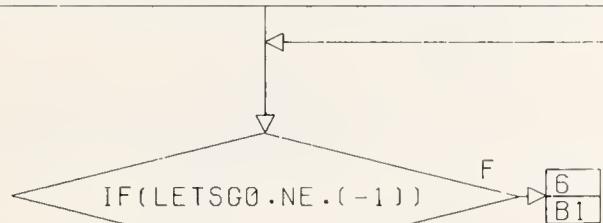
C CHECK THE FIRST NAME FOR SPECIAL NAMES...  
C OMNITAB, FORMAT, NOTE, FOOTNOTE, HEAD, TITLE  
C OMNITAB

IF (NAME(1).NE.11300.OR.NAME(2).NE.7102)

T



C IF NOT THE FIRST OMNITAB CARD, WRITE EOF RECORD.

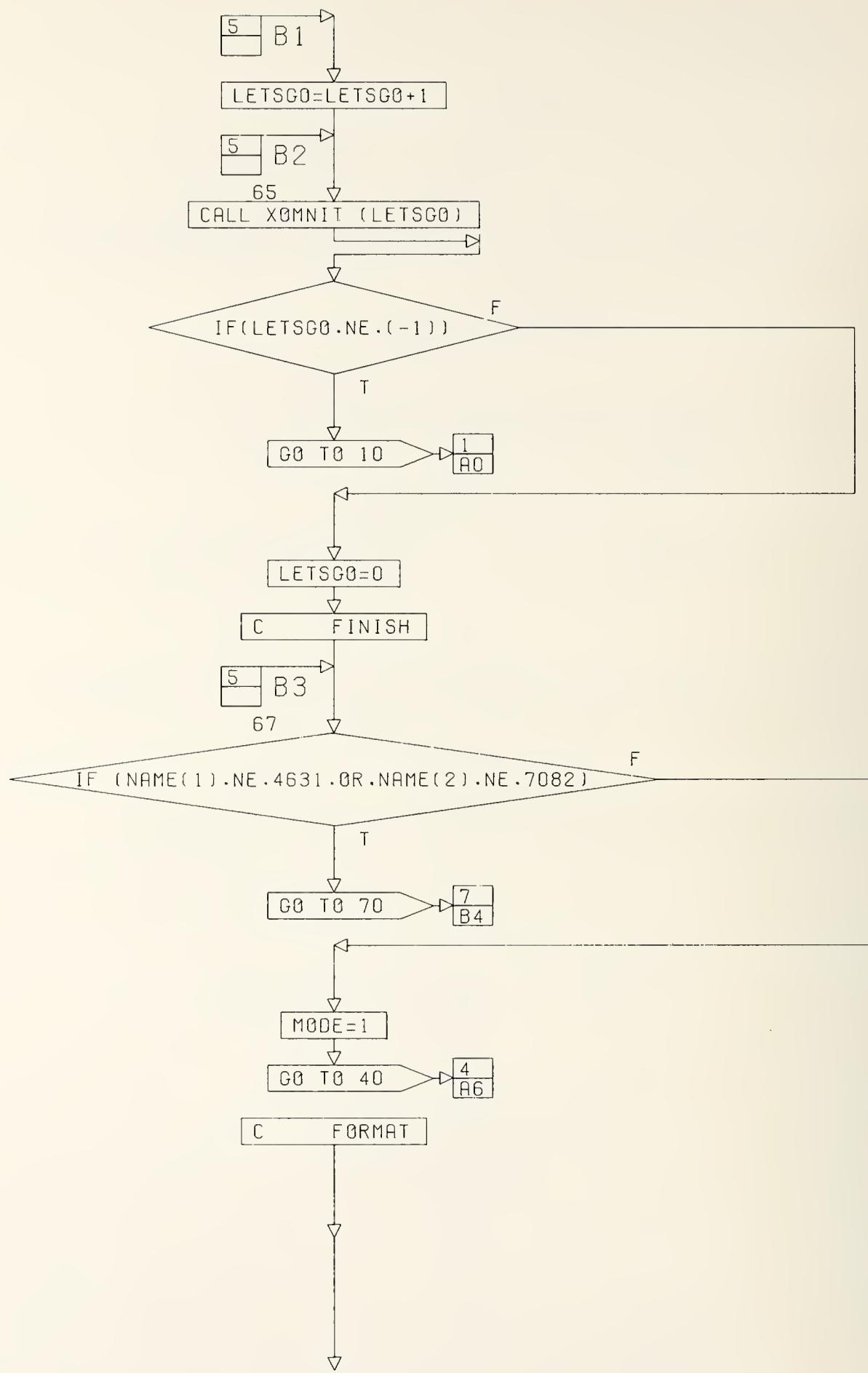


T

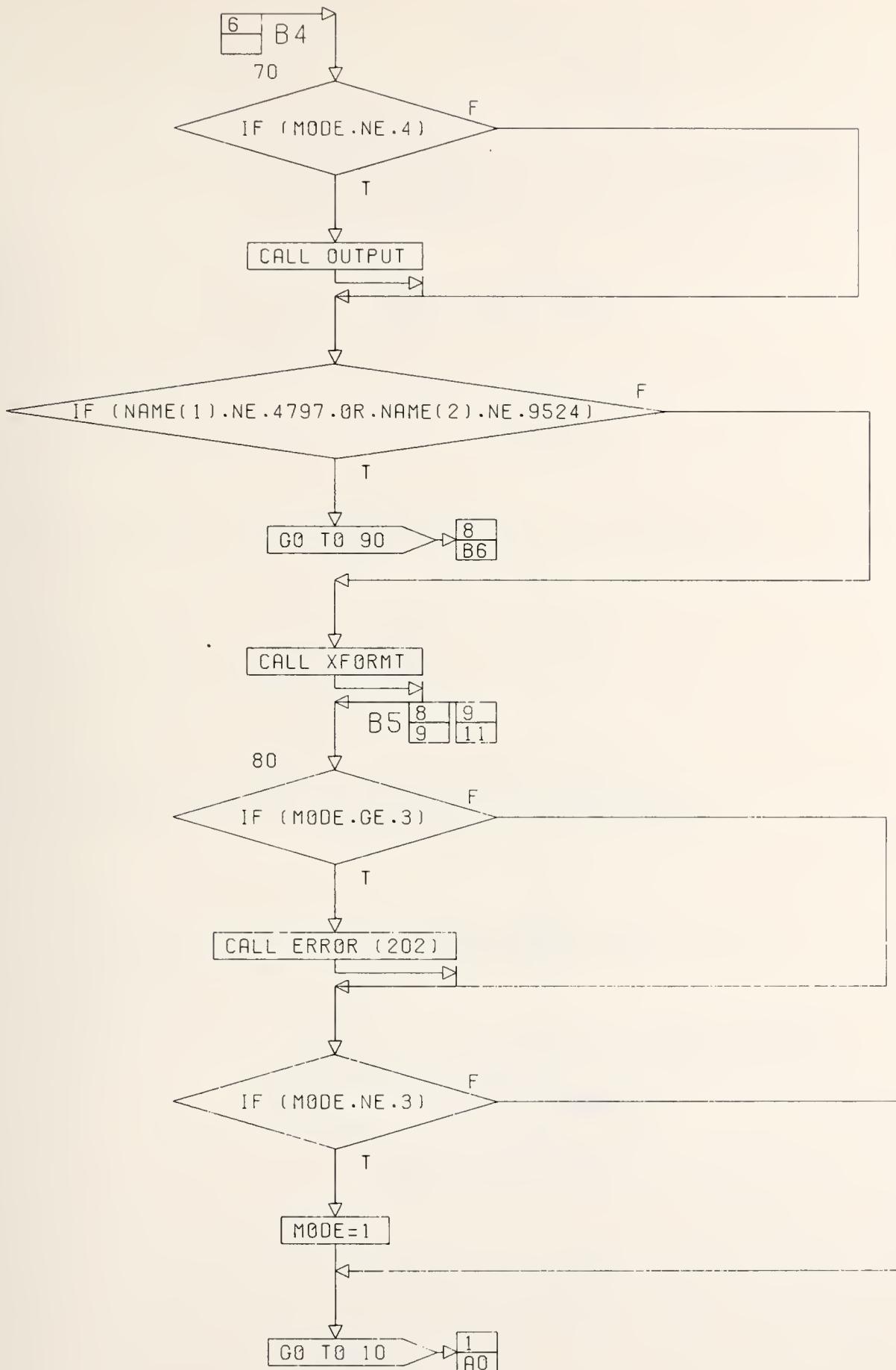
WRITE (ISCRAT,390)

CONT. ON PG 6

PG 5 OF 22

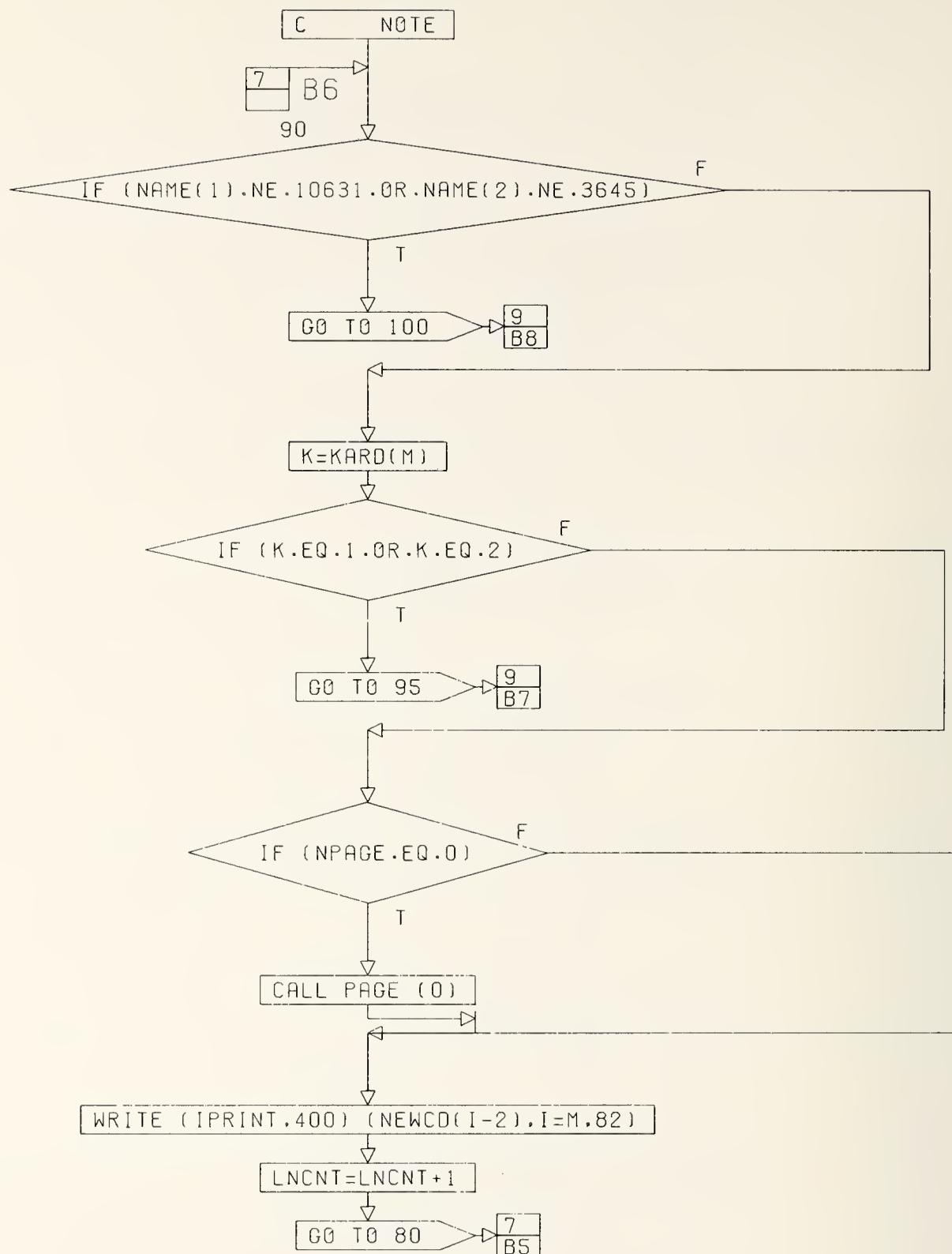


CONT. ON PG 7



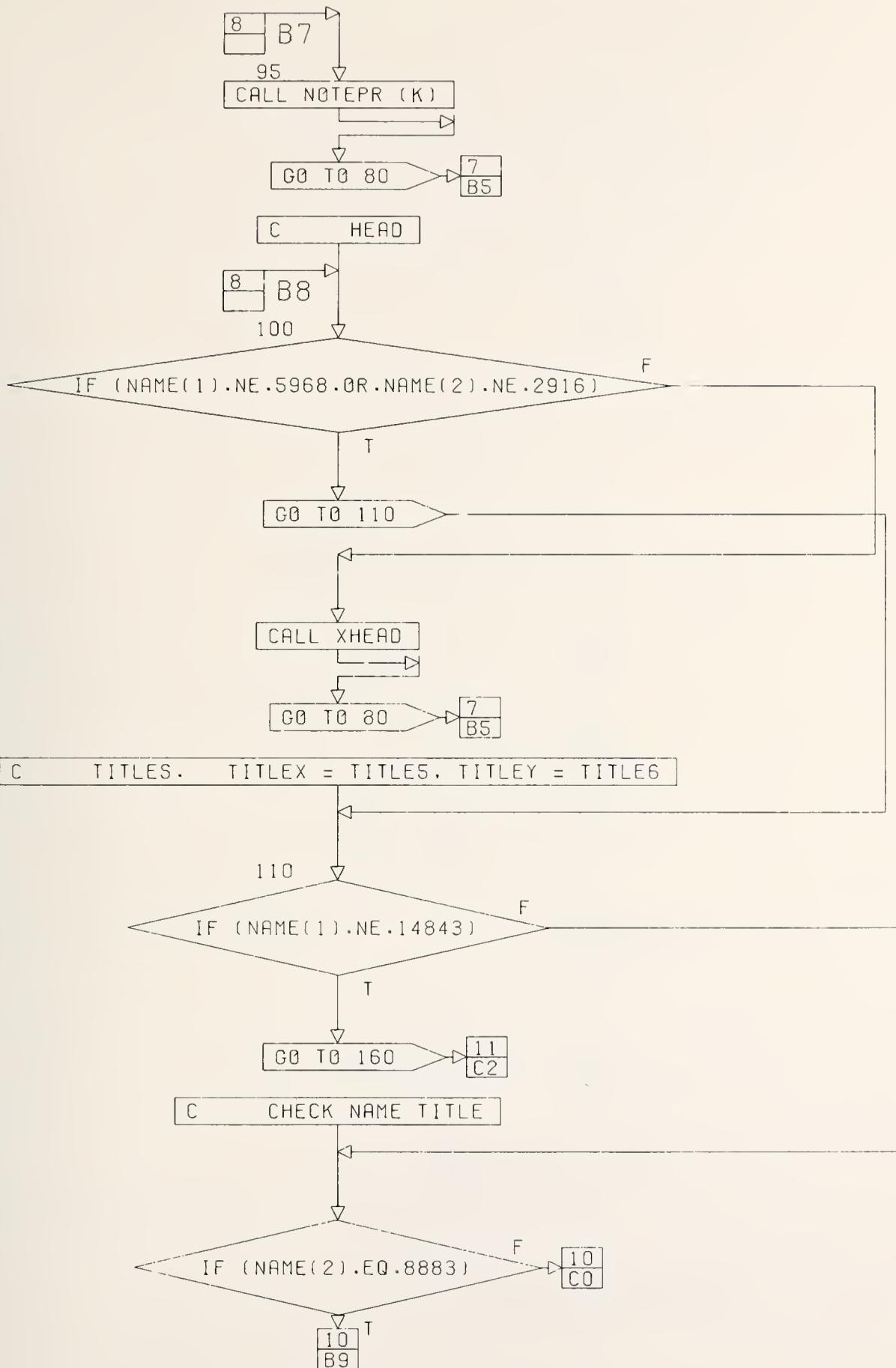
CONT. ON PG 8

PG 7 OF 22



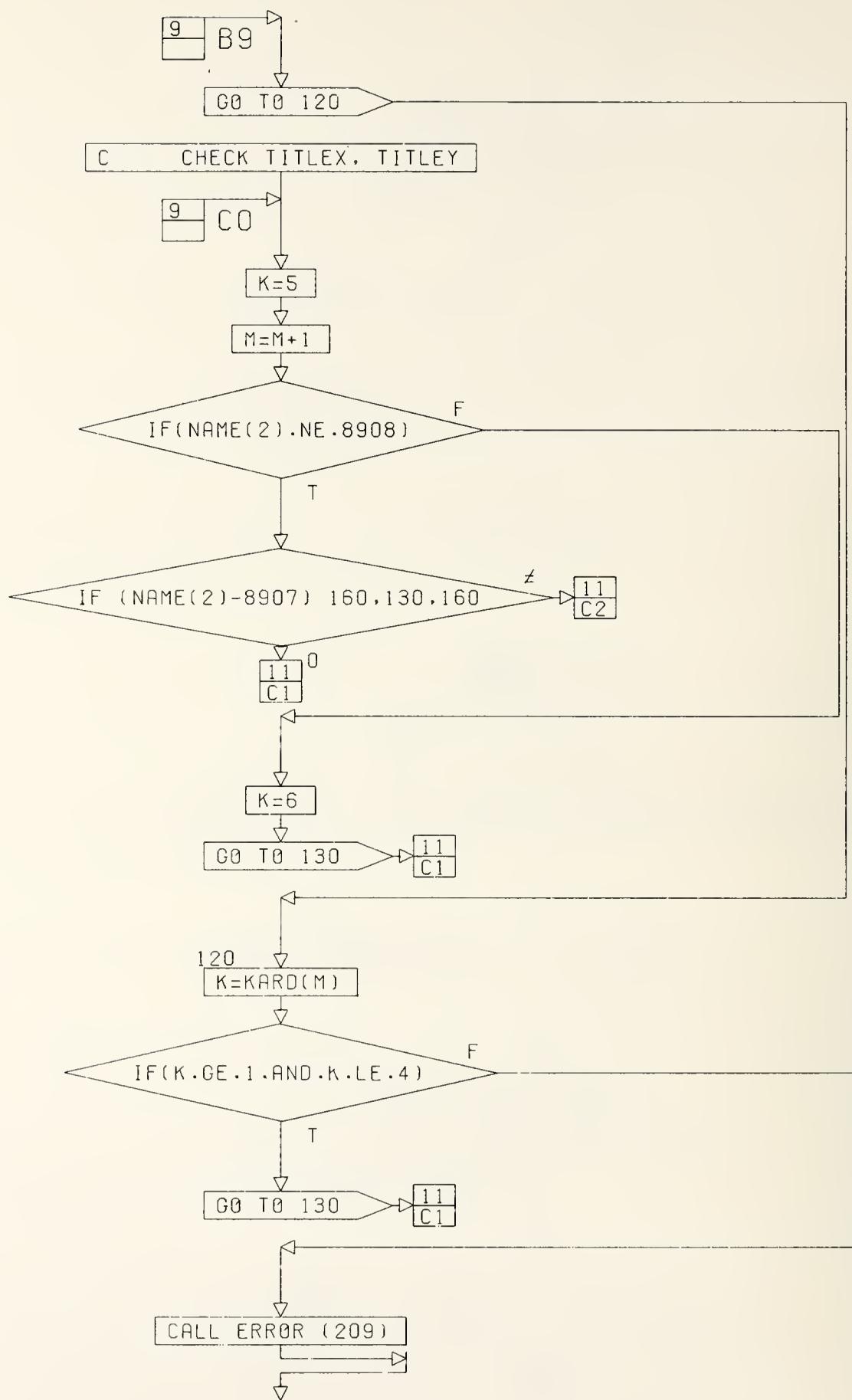
CONT. ON PG 9

PG 8 OF 22



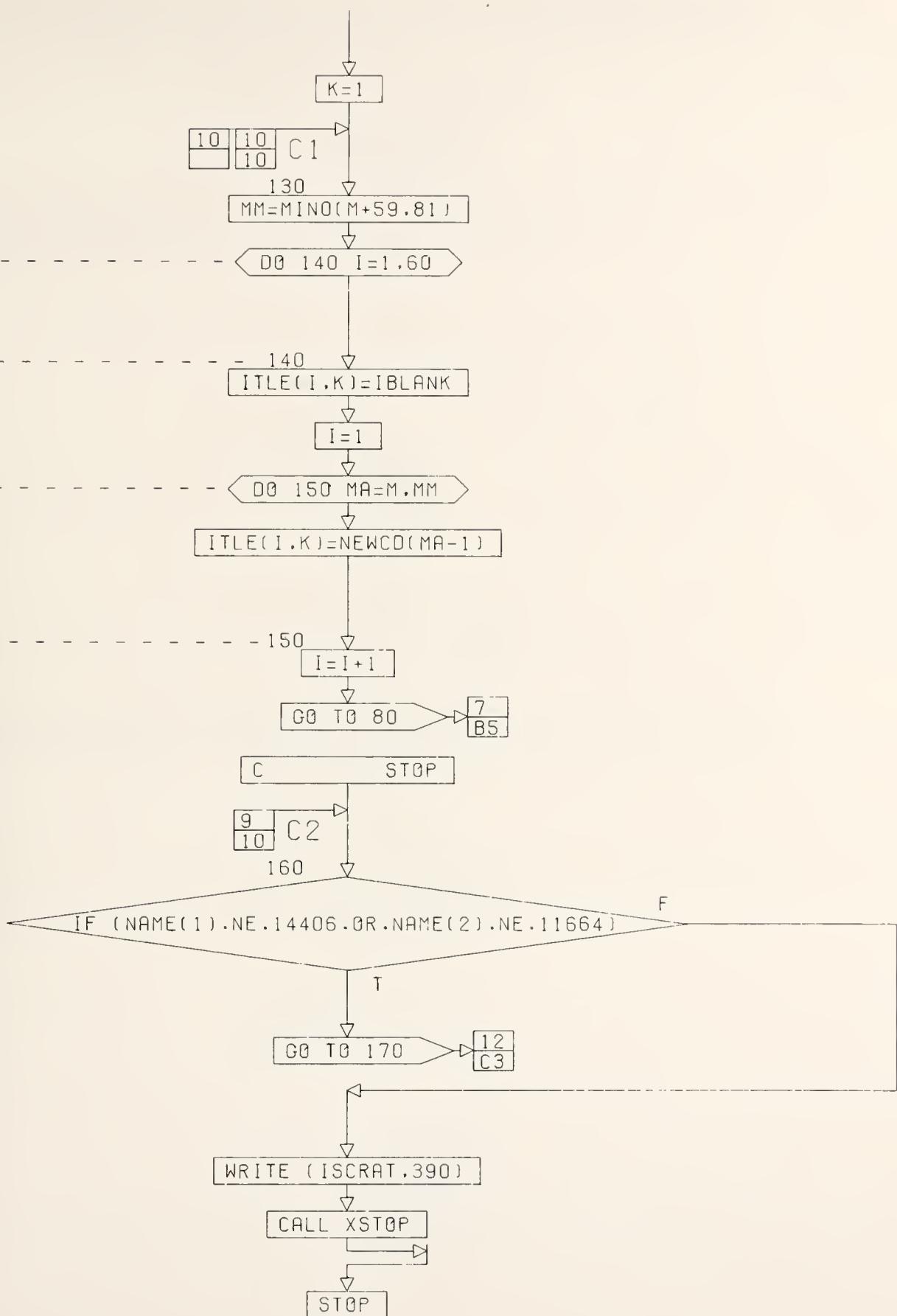
CONT. ON PG 10

PG 9 OF 22



CONT. ON PG 11

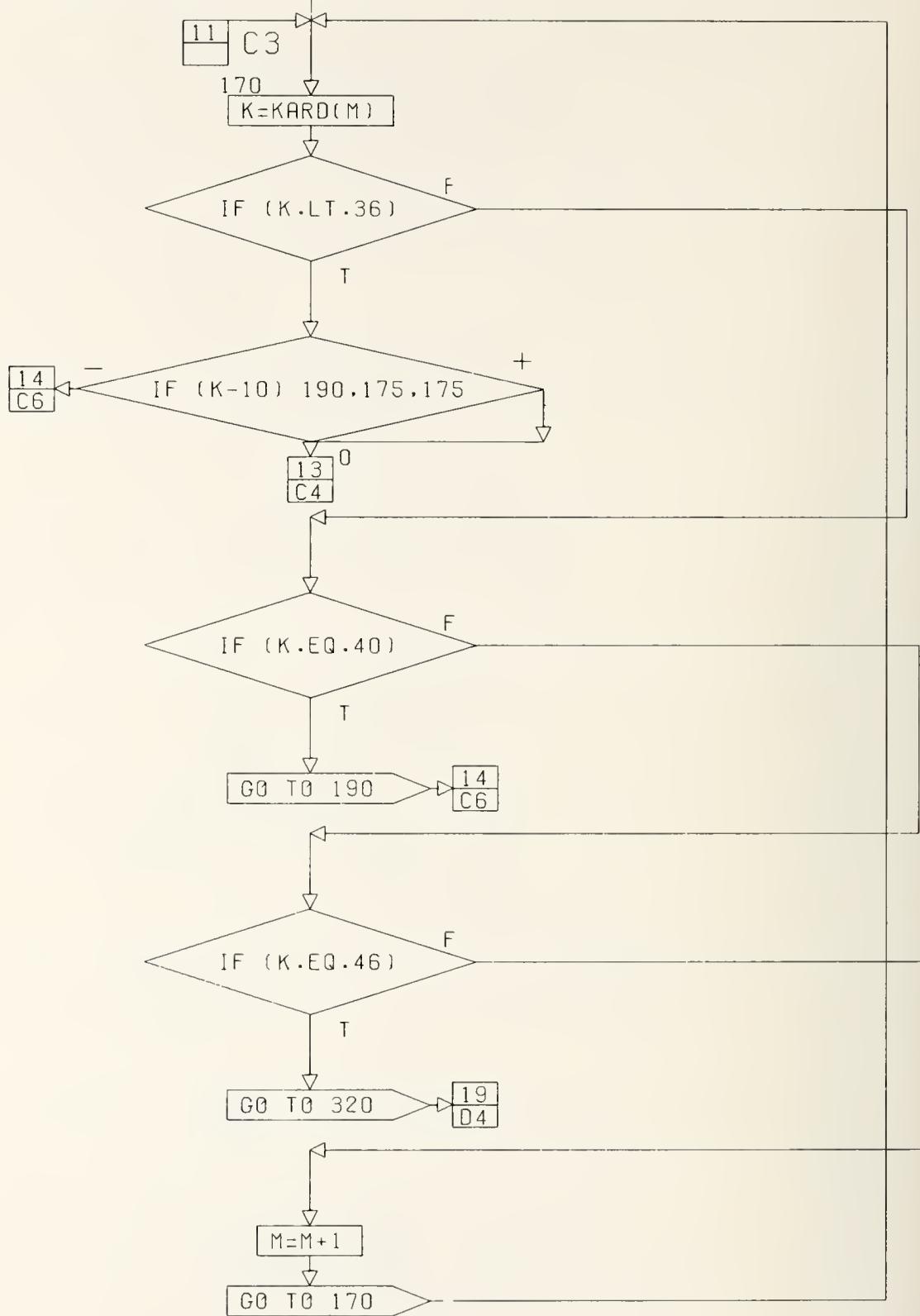
PG 10 OF 22



CONT. ON PG 12

PG 11 OF 22

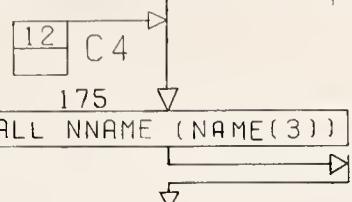
C C M IS POINTING AT THE FIRST NON-LETTER AFTER NAME. LOOK FOR POSSIBLE NAME QUALIFIER OR ARGUMENTS OR END OF CARD.



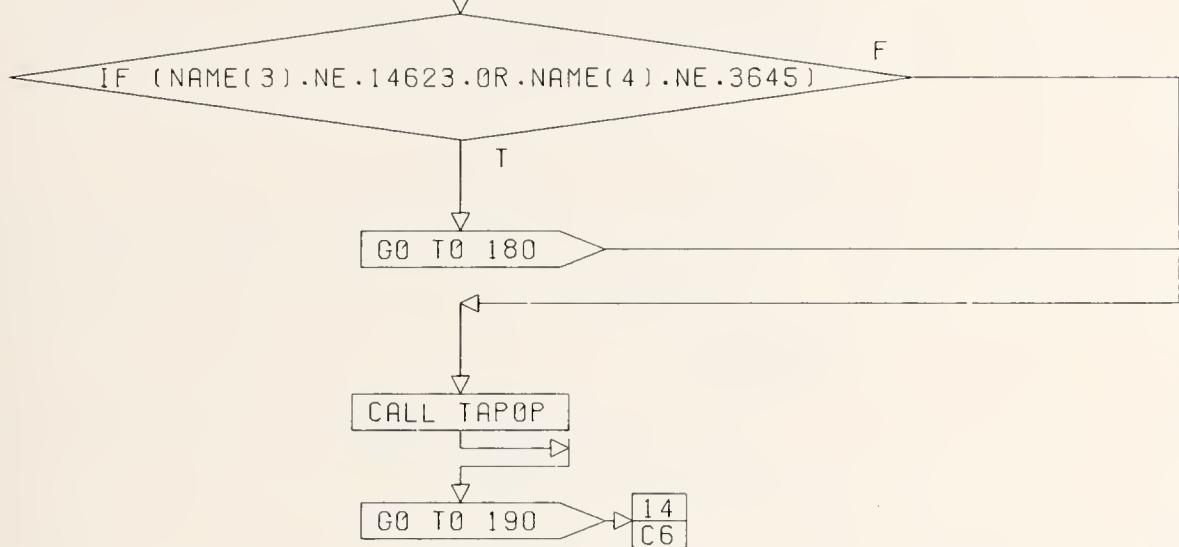
CONT. ON PG 13

PG 12 OF 22

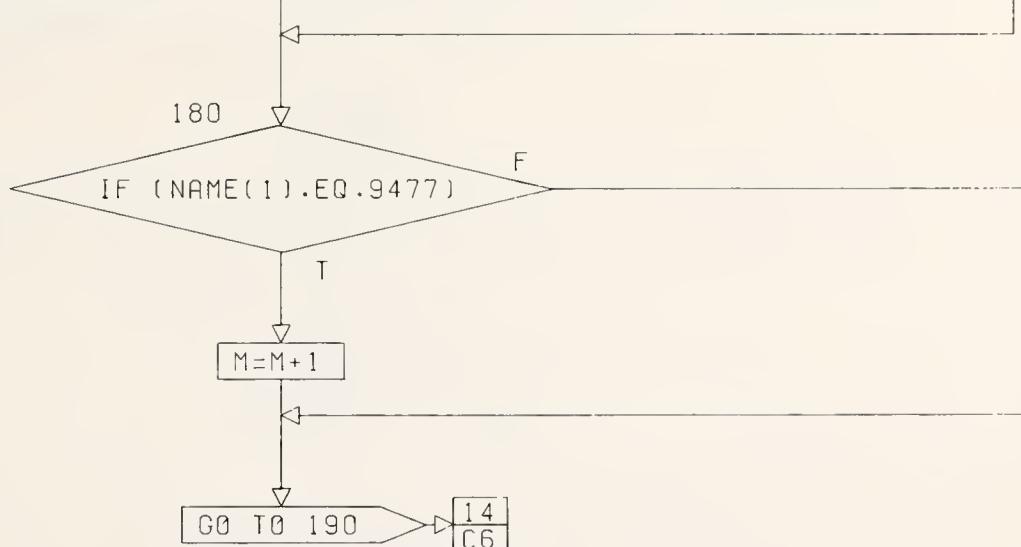
C A LETTER FOUND. ASSEMBLE SECOND NAME (COMMAND QUALIFIER).



C CHECK SPECIAL CASE OF NAMES M(X'AX'), M(X'AX), M(XX'), M(X'X)  
C SKIP ONE CHARACTER (.) IF FIRST NAME = (M )  
C THE FOLLOWING CARD IS NEEDED ONLY FOR TAPE OPERATIONS  
C IS NAME(3) EQUAL TO TAP AND NAME(4)=E



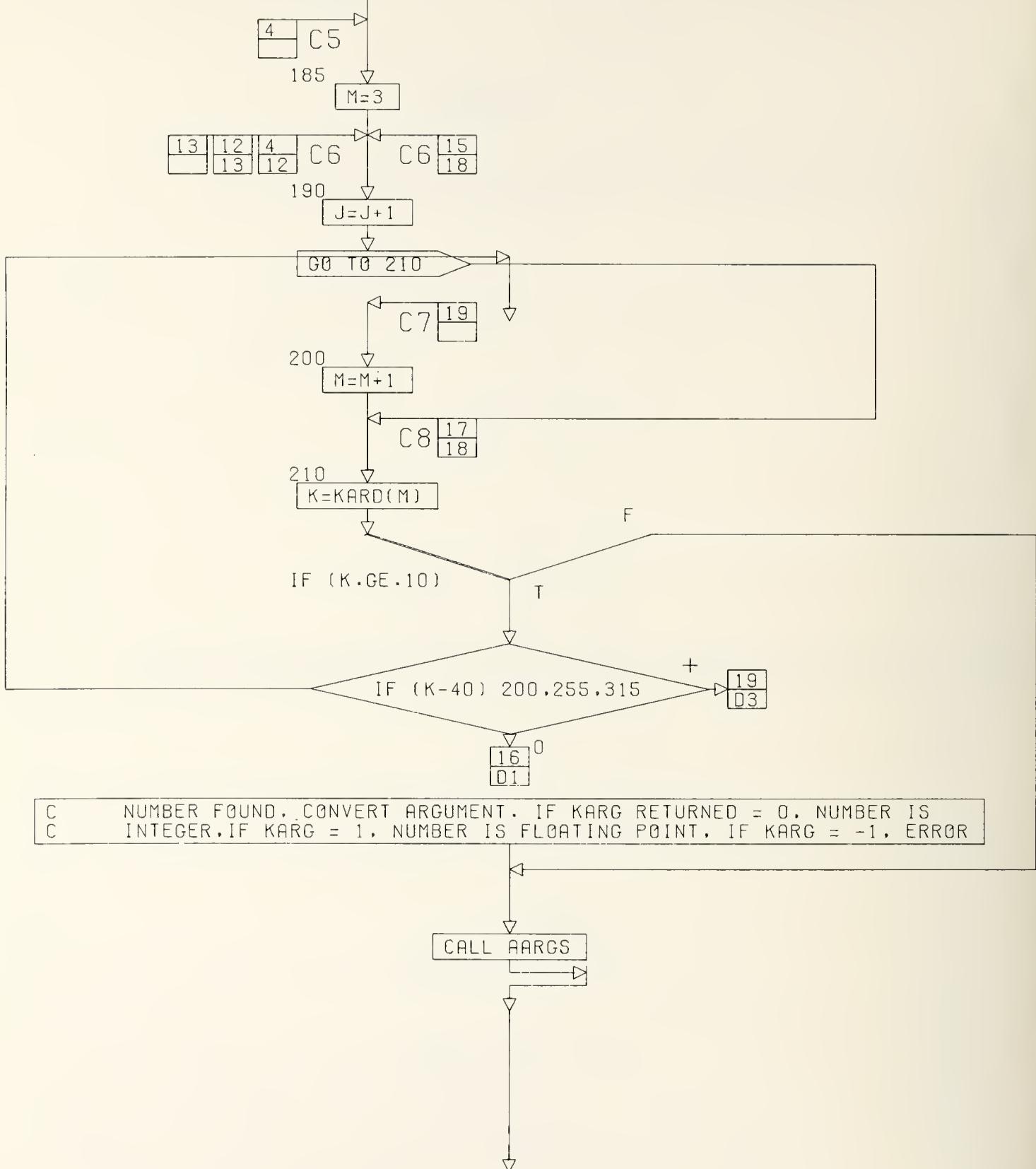
C \*\*\*\*\*



CONT. ON PG 14

PG 13 OF 22

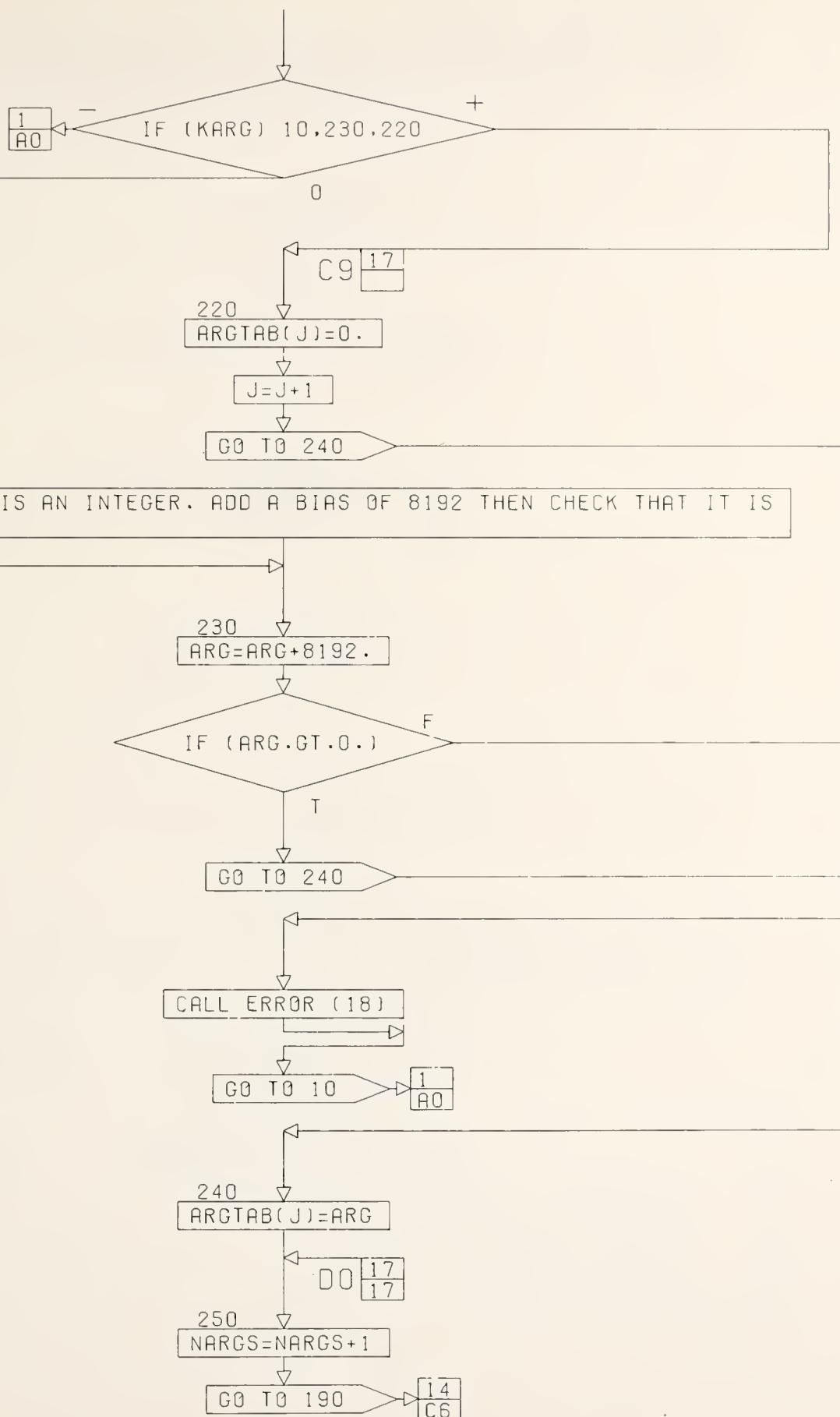
C SCAN FOR ARGUMENTS AND END OF CARD



C NUMBER FOUND. CONVERT ARGUMENT. IF KARG RETURNED = 0, NUMBER IS  
C INTEGER, IF KARG = 1, NUMBER IS FLOATING POINT, IF KARG = -1, ERROR

CONT. ON PG 15

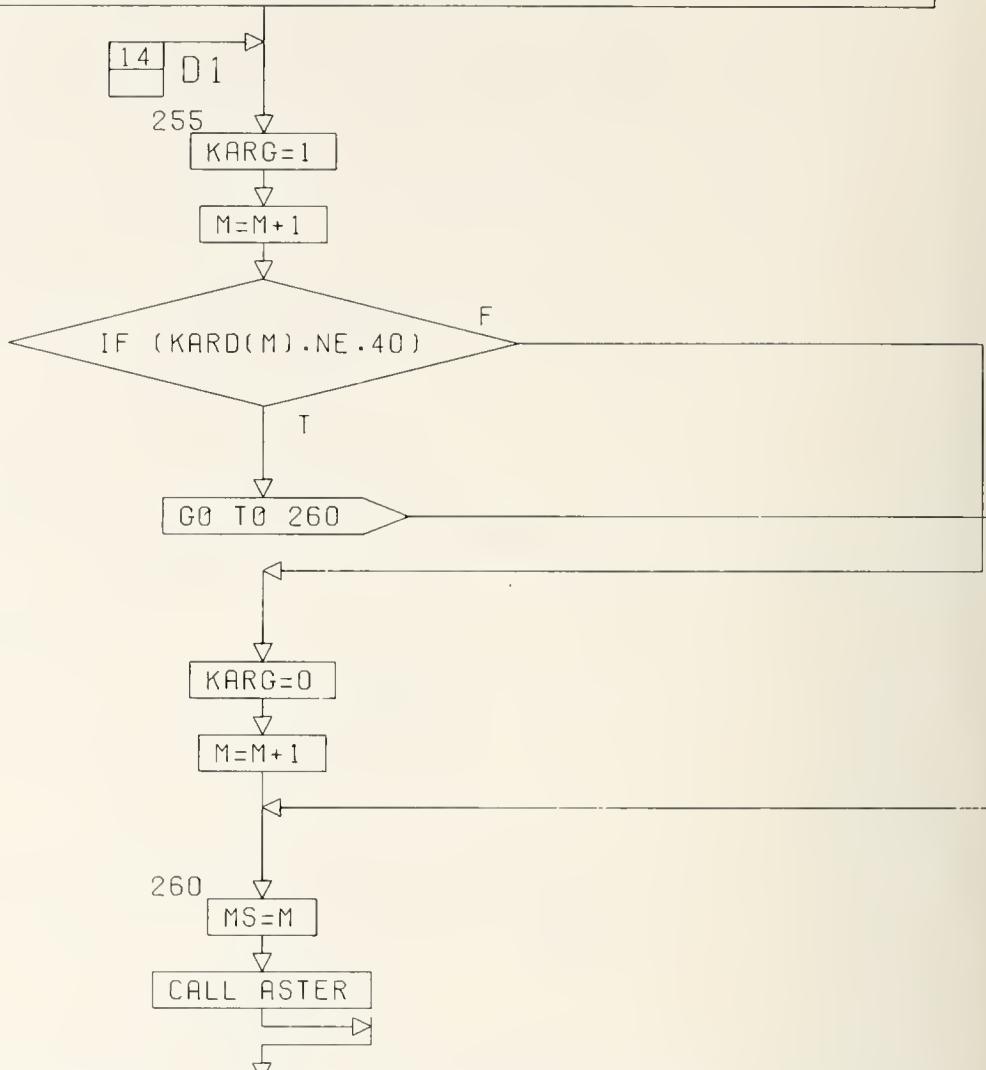
PG 14 OF 22



CONT. ON PG 16

PG 15 OF 22

C ASTERISK FOUND. CONVERT  
C IF BRACKETED BY SINGLE ASTERISKS, QUANTITY IS TO BE USED AS A  
C FLOATING POINT ARGUMENT. IF BRACKETED BY DOUBLE ASTERISKS, QUANTITY  
C IS TO BE TRUNCATED AND USED AS AN INTEGER ARGUMENT.

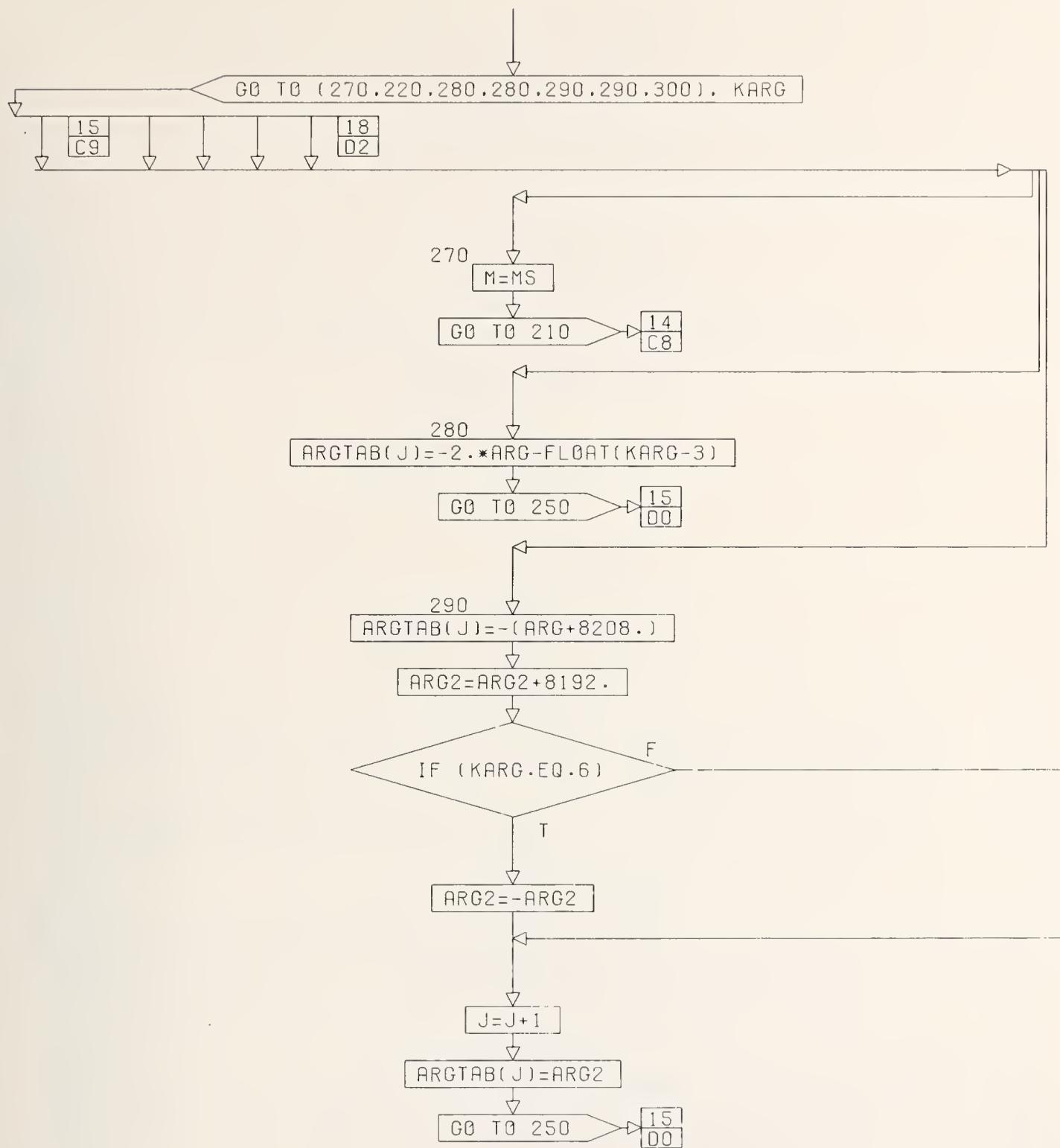


C THE TERMINAL ASTERISK(S) HAVE BEEN CHECKED TO BE THE SAME AS THE  
C INITIAL SET (IF NO ERROR) AND M IS POINTING AT THE FIRST CHARACTER  
C AFTER THE LAST ASTERISK.  
C KARG RETURNED AS  
C 1 = ERROR FOUND  
C 2 = FLOATING POINT CONSTANT, Z.B. \*PI\*  
C 3 = INTEGER NAMED VARIABLE, Z.B. \*\*NRMAX\*\*  
C 4 = FL. PT. NAMEO VARIABLE, Z.B. \*NRMAX\*  
C 5 = INTEGER ROW-COLUMN, Z.B. \*\*3,40\*\*

C  
C 6 = FL. PT. ROW-COLUMN, Z.B. \*1,2\*  
C 7 = STRING OF ASTERISKS Z.B. \*\*\*  
C A STRING OF THREE OR MORE ASTERISKS IMPLIES -THRU-  
C EXAMPLE..  
C ERASE 1 2 3 4 12 13 14 15 16 20 IS EQUIVALENT TO  
C ERASE 1 \*\*\* 4, 12 \*\*\* 16, 20  
C PRINT 1 20 19 18 17 16 15 14 IS EQUIVALENT TO  
C PRINT 1, 20 \*\*\* 14

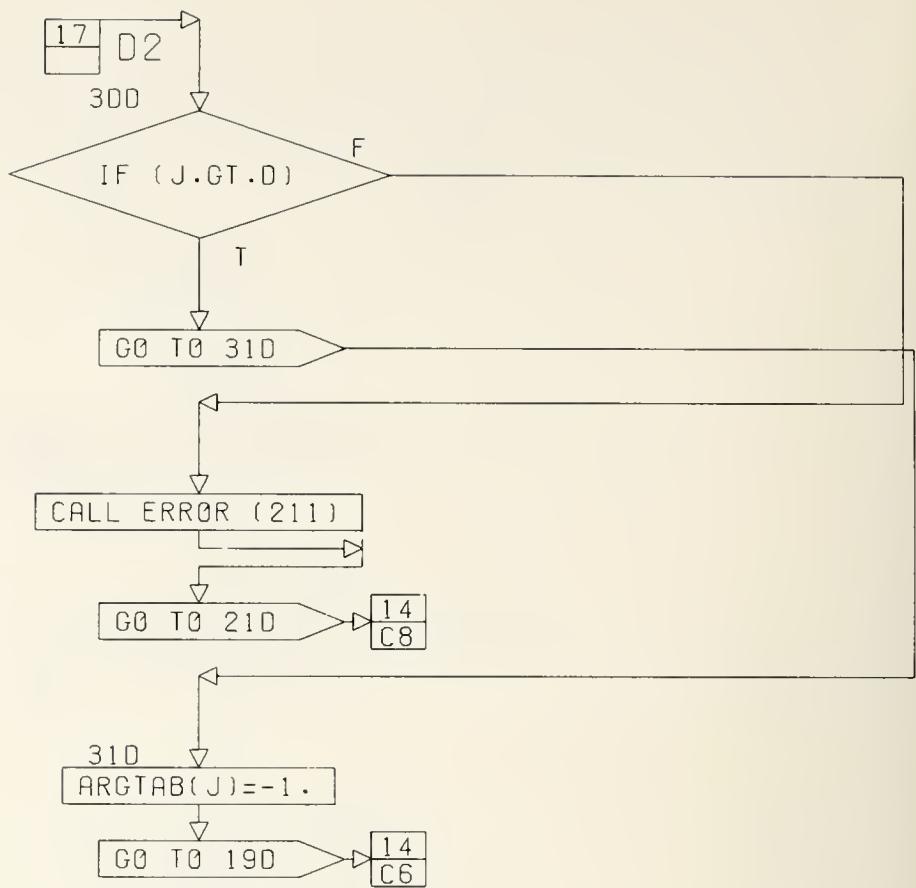
CONT. ON PG 17

PG 16 OF 22



CONT. ON PG 18

PG 17 OF 22



C ARGTAB SETUP  
 C IF ENTRY .GT. D, IT IS AN INTEGER CONSTANT (Z.B. COLUMN NUMBER)  
 C TO WHICH A BIAS OF 8192 HAS BEEN ADDED. THIS IS TO SAY THAT A  
 C NEGATIVE INTEGER ARGUMENT MAY NOT BE EXPLICITLY GIVEN OR MODIFIED  
 C TO BE LESS THAN -8191.  
 C IF ENTRY .EQ.D, THE NEXT ENTRY IS A FLOATING POINT CONSTANT.  
 C IF ENTRY .LT. D, ARGUMENT IS A VARIABLE. SET SIGN POSITIVE AND..  
 C IF ENTRY .LT. 16, IT IS A NAMED VARIABLE REFERENCE NUMBER

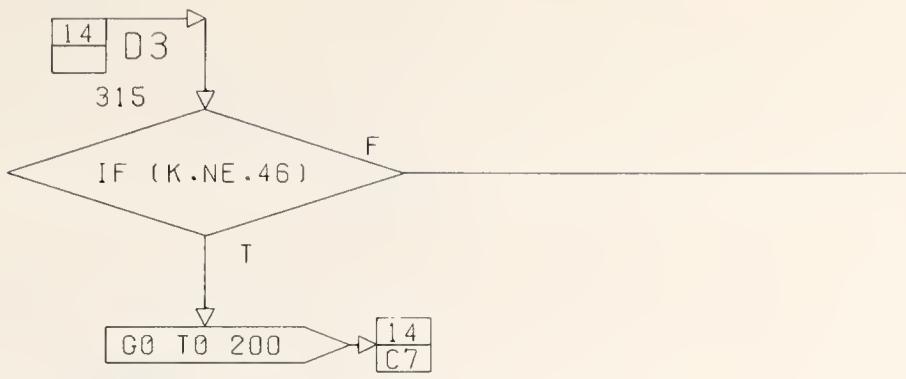
C 2,3 NRMAX 6.7 V 1D.11 X  
 C 8.9 W 12.13 Y  
 C 14.15 Z  
 C V,W,X,Y,Z, ARE FOR PROGRAMMING CONVENIENCE ONLY AND DO NOT  
 C AFFECT THE OPERATION OF OMNITAB  
 C IF ENTRY IS EVEN, CURRENT VALUE TO BE TRUNCATED AND USED  
 C AS AN INTEGER ARGUMENT.  
 C IF ENTRY IS ODD, THE CURRENT VALUE IS TO BE USED AS A

C FLOATING POINT ARGUMENT.  
 C IF ENTRY .GT. 16, IT IS A WORKSHEET REFERENCE (ROW,COLUMN) TO  
 C WHICH A BIAS OF 8192. HAS BEEN ADDED.  
 C ENTRY - 82D8 = ROW NUMBER  
 C ABS(NEXT ENTRY) = COLUMN NUMBER TO WHICH A BIAS OF 8192.  
 C HAS BEEN ADDED.  
 C IF NEXT ENTRY IS NEGATIVE, WORKSHEET CONTENTS ARE TO BE  
 C USED AS A FLOATING POINT CONSTANT. IF +, WORKSHEET VALUE

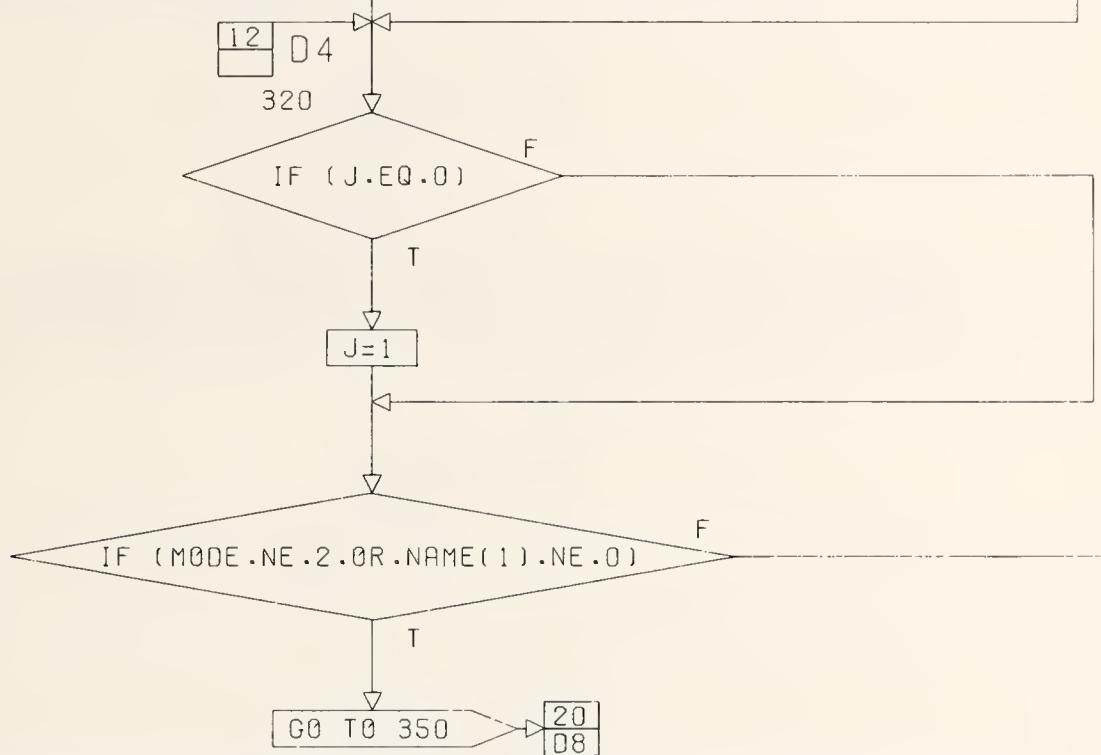
C TO BE TRUNCATED AND USED AS AN INTEGER ARGUMENT.

CONT. ON PG 19

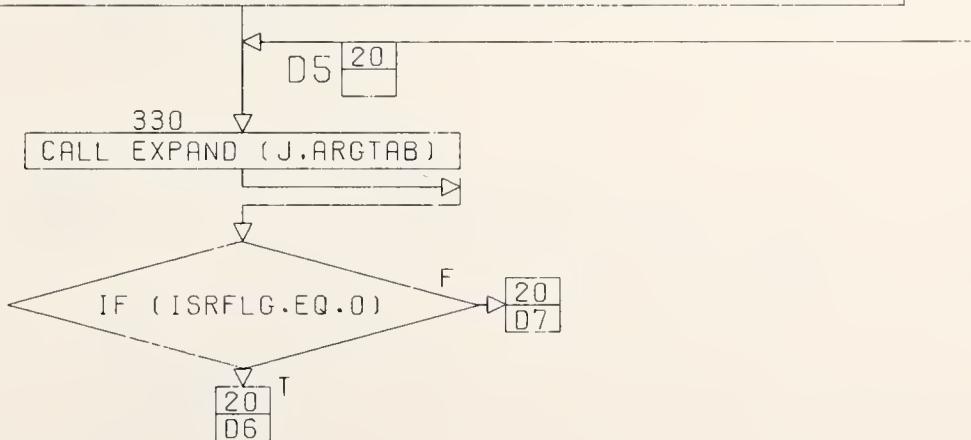
PG 18 OF 22



C THE TERMINATION OF CARD FOUND ( \$ ENCOUNTERED)

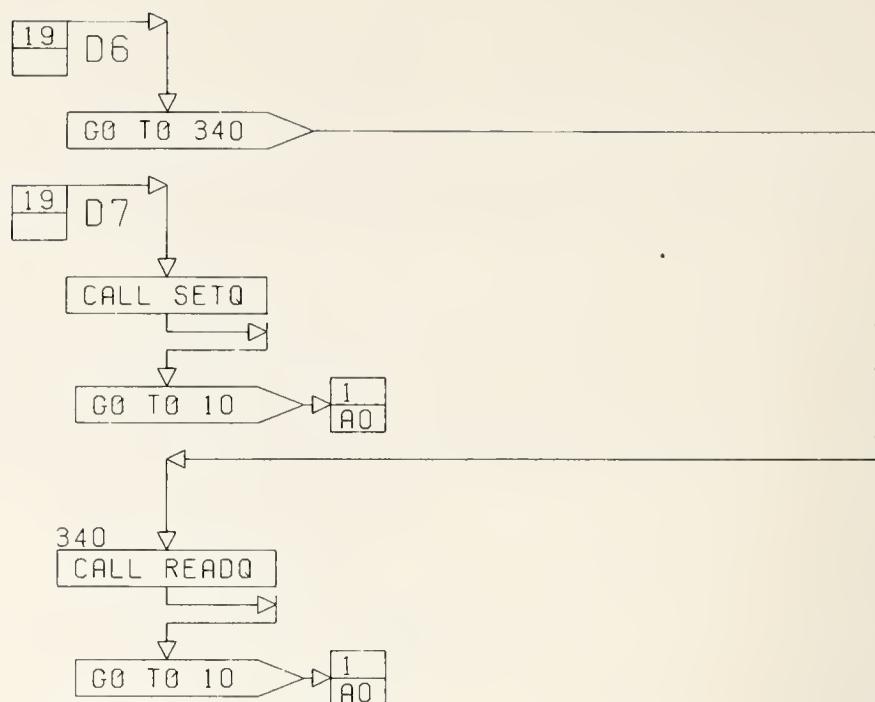


C IN INPUT MODE AND NO POSSIBLE NAME, RETURN TO SET OR READ ROUTINE

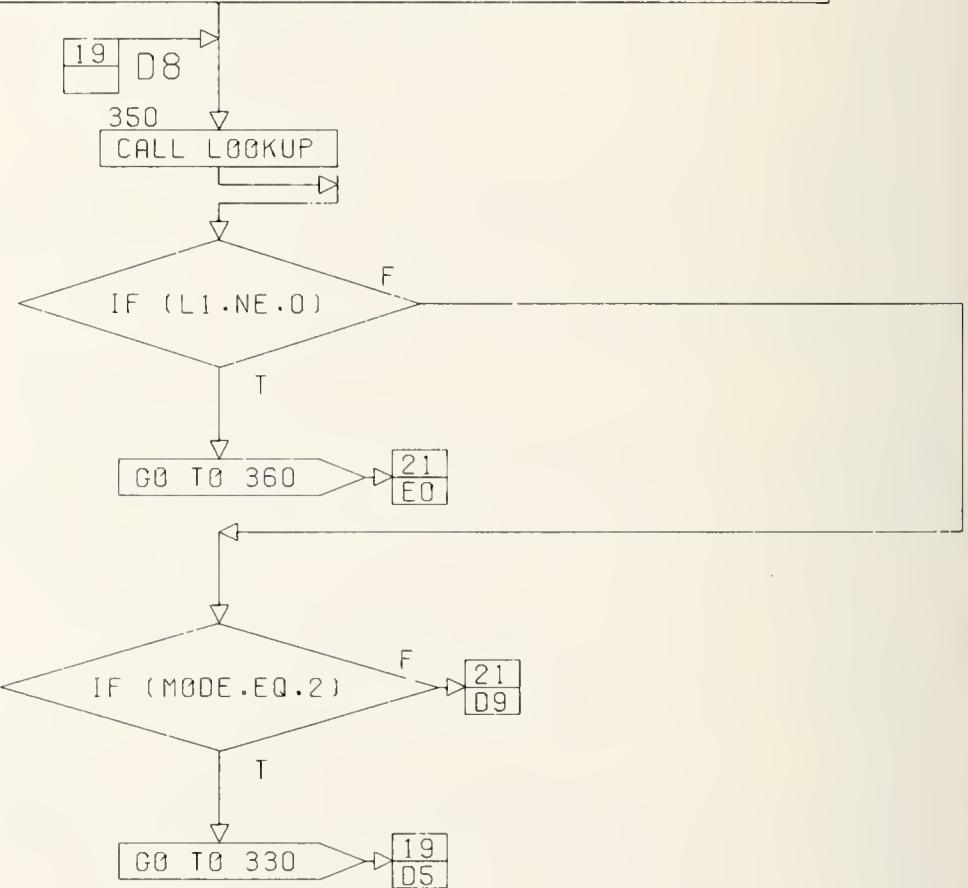


CONT. ON PG 20

PG 19 OF 22

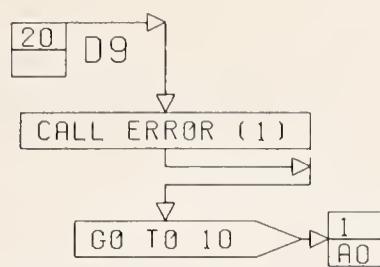


C LOOK UP NAME (AND POSSIBLE QUALIFIER) IN DICTIONARY. RETURN  
 C COORDINATES OF ENTRY. IF L1 = 0, NAME NOT FOUND

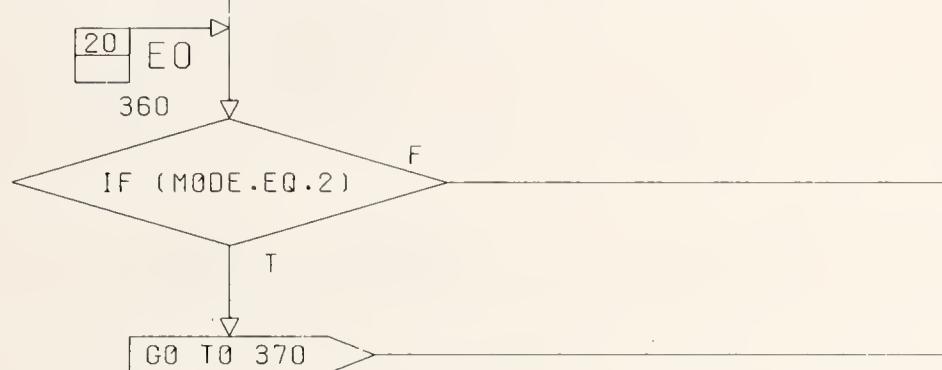


CONT. ON PG 21

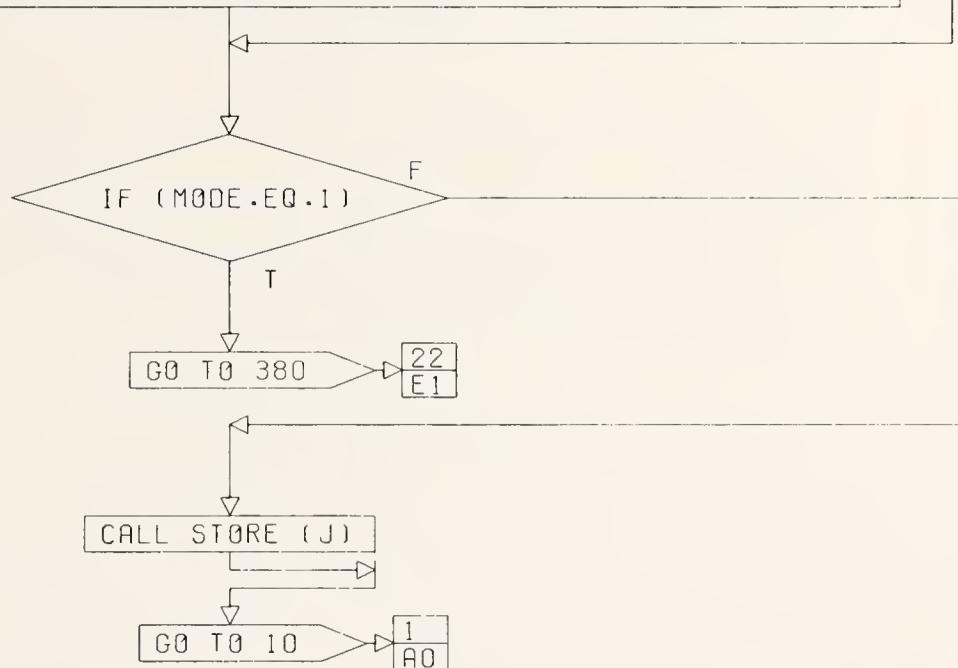
PG 20 OF 22



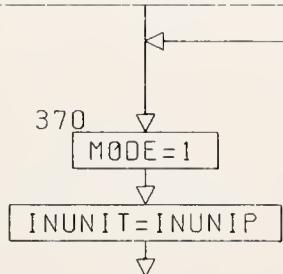
C NAME FOUND  
C THE FOLLOWING CARDS ARE NEEDED ONLY FOR TAPE OPERATIONS  
C STATEMENT WAS 220 IF (MODE.EQ.2) MODE=1



C \*\*\*\*

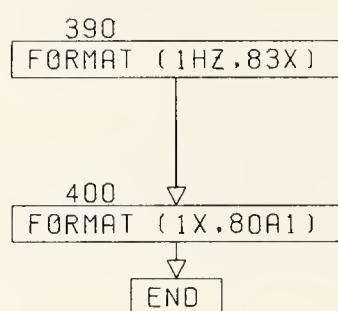
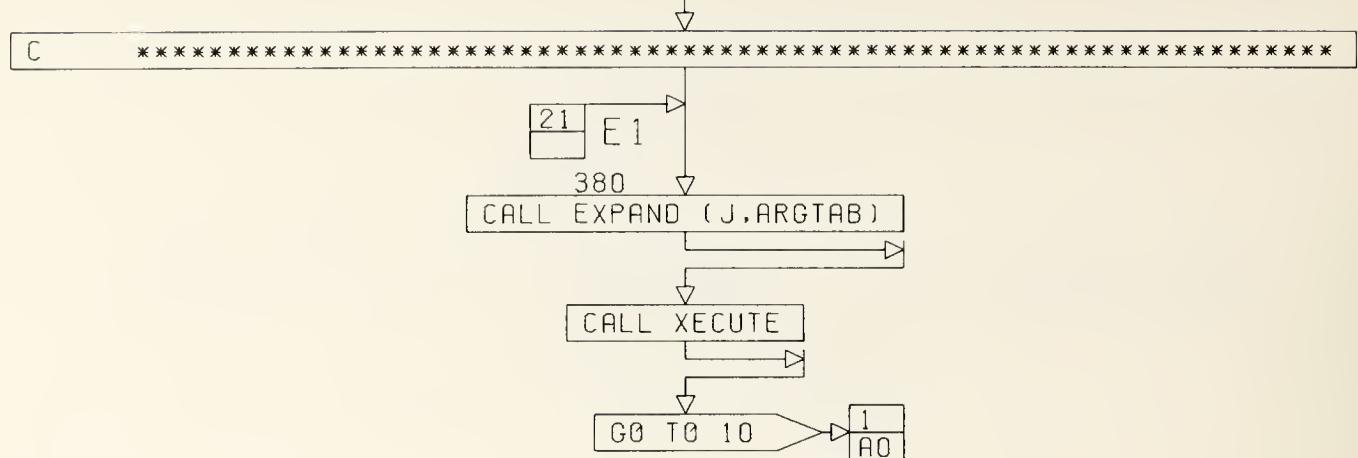


C THE FOLLOWING CARDS ARE NEEDED ONLY FOR TAPE OPERATIONS



CONT. ON PG 22

PG 21 OF 22



**Latest developments in the subject area of this publication, as well as  
in other areas where the National Bureau of Standards is active, are  
reported in the NBS Technical News Bulletin. See following page.**

## HOW TO KEEP ABREAST OF NBS ACTIVITIES

Your purchase of this publication indicates an interest in the research, development, technology, or service activities of the National Bureau of Standards.

The best source of current awareness in your specific area, as well as in other NBS programs of possible interest, is the TECHNICAL NEWS BULLETIN, a monthly magazine designed for engineers, chemists, physicists, research and product development managers, librarians, and company executives.

If you do not now receive the TECHNICAL NEWS BULLETIN and would like to subscribe, and/or to review some recent issues, please fill out and return the form below.

(cut here)

Mail to: Office of Technical Information and Publications  
National Bureau of Standards  
Washington, D. C. 20234

Name \_\_\_\_\_

Affiliation \_\_\_\_\_

Address \_\_\_\_\_

City \_\_\_\_\_ State \_\_\_\_\_ Zip \_\_\_\_\_

Please send complimentary past issues of the Technical News Bulletin.

Please enter my 1-yr subscription. Enclosed is my check or money order for \$3.00 (additional \$1.00 for foreign mailing).

*Check is made payable to: SUPERINTENDENT OF DOCUMENTS.*

SP 339

## PERIODICALS

**JOURNAL OF RESEARCH** reports National Bureau of Standards research and development in physics, mathematics, chemistry, and engineering. Comprehensive scientific papers give complete details of the work, including laboratory data, experimental procedures, and theoretical and mathematical analyses. Illustrated with photographs, drawings, and charts.

*Published in three sections, available separately:*

### ● Physics and Chemistry

Papers of interest primarily to scientists working in these fields. This section covers a broad range of physical and chemical research, with major emphasis on standards of physical measurement, fundamental constants, and properties of matter. Issued six times a year. Annual subscription: Domestic, \$9.50; foreign, \$11.75\*.

### ● Mathematical Sciences

Studies and compilations designed mainly for the mathematician and theoretical physicist. Topics in mathematical statistics, theory of experiment design, numerical analysis, theoretical physics and chemistry, logical design and programming of computers and computer systems. Short numerical tables. Issued quarterly. Annual subscription: Domestic, \$5.00; foreign, \$6.25\*.

### ● Engineering and Instrumentation

Reporting results of interest chiefly to the engineer and the applied scientist. This section includes many of the new developments in instrumentation resulting from the Bureau's work in physical measurement, data processing, and development of test methods. It will also cover some of the work in acoustics, applied mechanics, building research, and cryogenic engineering. Issued quarterly. Annual subscription: Domestic, \$5.00; foreign, \$6.25\*.

## TECHNICAL NEWS BULLETIN

The best single source of information concerning the Bureau's research, developmental, cooperative and publication activities, this monthly publication is designed for the industry-oriented individual whose daily work involves intimate contact with science and technology—for engineers, chemists, physicists, research managers, product-development managers, and company executives. Annual subscription: Domestic, \$3.00; foreign, \$4.00\*.

\* Difference in price is due to extra cost of foreign mailing.

Order NBS publications from:

Superintendent of Documents  
Government Printing Office  
Washington, D.C. 20402

## NONPERIODICALS

**Applied Mathematics Series.** Mathematical tables, manuals, and studies.

**Building Science Series.** Research results, test methods, and performance criteria of building materials, components, systems, and structures.

**Handbooks.** Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

**Special Publications.** Proceedings of NBS conferences, bibliographies, annual reports, wall charts, pamphlets, etc.

**Monographs.** Major contributions to the technical literature on various subjects related to the Bureau's scientific and technical activities.

**National Standard Reference Data Series.** NSRDS provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated.

**Product Standards.** Provide requirements for sizes, types, quality and methods for testing various industrial products. These standards are developed cooperatively with interested Government and industry groups and provide the basis for common understanding of product characteristics for both buyers and sellers. Their use is voluntary.

**Technical Notes.** This series consists of communications and reports (covering both other agency and NBS-sponsored work) of limited or transitory interest.

**Federal Information Processing Standards Publications.** This series is the official publication within the Federal Government for information on standards adopted and promulgated under the Public Law 89-306, and Bureau of the Budget Circular A-86 entitled, Standardization of Data Elements and Codes in Data Systems.









