NBSIR 80-1982

BFIRES/Version 2: Documentation of Program Modifications

Fred I. Stahl

Environmental Design Research Division
Center for Building Technology
National Engineering Laboratory
National Bureau of Standards
U.S. Department of Commerce
Washington, D.C. 20234

March 1980

Developed with the partial assistance of the
Center for Fire Research, in support of the
HEW-NBS Fire/Life Safety Program

NBSIR 80-1982

BFIRES/VERSION 2: DOCUMENTATION OF PROGRAM MODIFICATIONS

Fred I. Stahl

Environmental Design Research Division
Center for Building Technology
National Engineering Laboratory
National Bureau of Standards
U.S. Department of Commerce
Washington, D.C. 20234

March 1980

Developed with the partial assistance of the
Center for Fire Research, in support of the
HEW-NBS Fire/Life Safety Program

U.S. DEPARTMENT OF COMMERCE, Philip M. Klutznick, ***Secretary***

Luther H. Hodges, Jr., ***Deputy Secretary***
Jordan J. Baruch, ***Assistant Secretary for Productivity, Technology, and Innovation***

NATIONAL BUREAU OF STANDARDS, Ernest Ambler, ***Director***

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

The Environmental Design Research Division, Center for Building Technology, National Bureau of Standards (NBS), is developing a more comprehensive understanding of pedestrian movement within buildings. The responses of building occupants during fire emergencies is an important part of this overall effort.

This report documents a continuation of an effort to develop a computer simulation of human egress behavior during fires, which was originally undertaken for the NBS Center for Fire Research through its HEW-NBS Fire/Life Safety Program. Modifications of the BFIRES computer discussed in this report were prepared under the sponsorship of the Center for Building Technology.

The author gratefully acknowledges Mr. Brian Pierman, Dr. Stephen Margulis and Dr. Robert Glass of the Center for Building Technology, and Mr. Jeff Shibe of the Center for Fire Research, for their reviews of the report's clarity and technical accuracy. The author also wishes to thank the staff of the Center for Building Technology Word Processing Center who prepared various versions of the manuscript.

ABSTRACT

Several limitations to the use of BFIRES/VERSION 1 are discussed. Chief among these are the program's inability to simulate rescue activities during fire events, and to simulate direct interactions between occupant behavior and toxic qualities of smoke filled environments. This report documents a revised program, BFIRES/VERSION 2, which contains new subroutines developed to mitigate these problems. These subroutines are grouped into two modules: (1) a "smoke" module designed to simulate the experience of inhabiting a smoke filled environment, and (2) a "rescue" module intended to permit the rescue of physically immobile occupants. Additional improvements incorporated into BFIRES/VERSION 2 involve more efficient file management and data input facilities, and expanded output capabilities.

Key Words: Architectural research; building Fires; computer-aided design, environmental psychology; fire research, fire safety, human behavior in fires; modeling technique; programming; simulation of human behavior.

BFIRES/VERSION 2: Documentation of Program Modifications

1.0 INTRODUCTION

1.1 Purpose of this Report

This report documents recent modifications to BFIRES, a computer program designed to simulate the emergency egress behavior of building occupants during fires. Documentation of BFIRES/VERSION 1 (BFIRES I) is available in a previous publication of the National Bureau of Standards (NBS) (Stahl, 1979). The current report presents BFIRES/VERSION 2 (BFIRES II), and documents modifications of and additions to the original computer program. This report is meant as a companion to the earlier publication. As a convenience to the reader, however, a complete FORTRAN listing of BFIRES is provided in an appendix to the current document.

This document is not a user's manual. It does not provide specific instructions for defining simulation problems, setting up BFIRES input files, executing simulation experiments, or for interpreting and using BFIRES output. A separate user's manual is currently in preparation.[1] However, Table 5.1 of this report contains basic information necessary for preparing input files and for using BFIRES II, although detailed questions concerning the program's use should be directed to the author.

1.2 Factors Limiting the Usefulness of BFIRES I

The technical approach underlying BFIRES is based on a stochastic dynamic model of building fire events. Human decisionmaking and behavior are simulated (a) in response to environmental and social stimuli which carry information about the state of the fire event over time, and (b) under the premise that perception and cognition bias rather than determine human actions in the environment.

Several factors severely limit the overall applicability of BFIRES I, and these are specifically addressed by the modifications documented in this report. For example, BFIRES I does not permit the user to simulate direct interactions between building occupants and toxic agents (as typically occur when an occupant inhabits a room or space which has been infiltrated by fire or smoke). Instead, that program simulates occupants' responses to information about a fire occurring in some other part of the building. Thus, the application of BFIRES I requires the user to assume that (a) the fire ignition point is reasonably distant from the floor or area under direct study, (b) occupants of the area under study have obtained information about the fire's presence, and (c) the emergency escape activity will not be noticeably affected by any migration of toxicants from the point of ignition to the occupied area. Another important

[1] The BFIRES II user's manual is expected to be available during the fall of 1980.

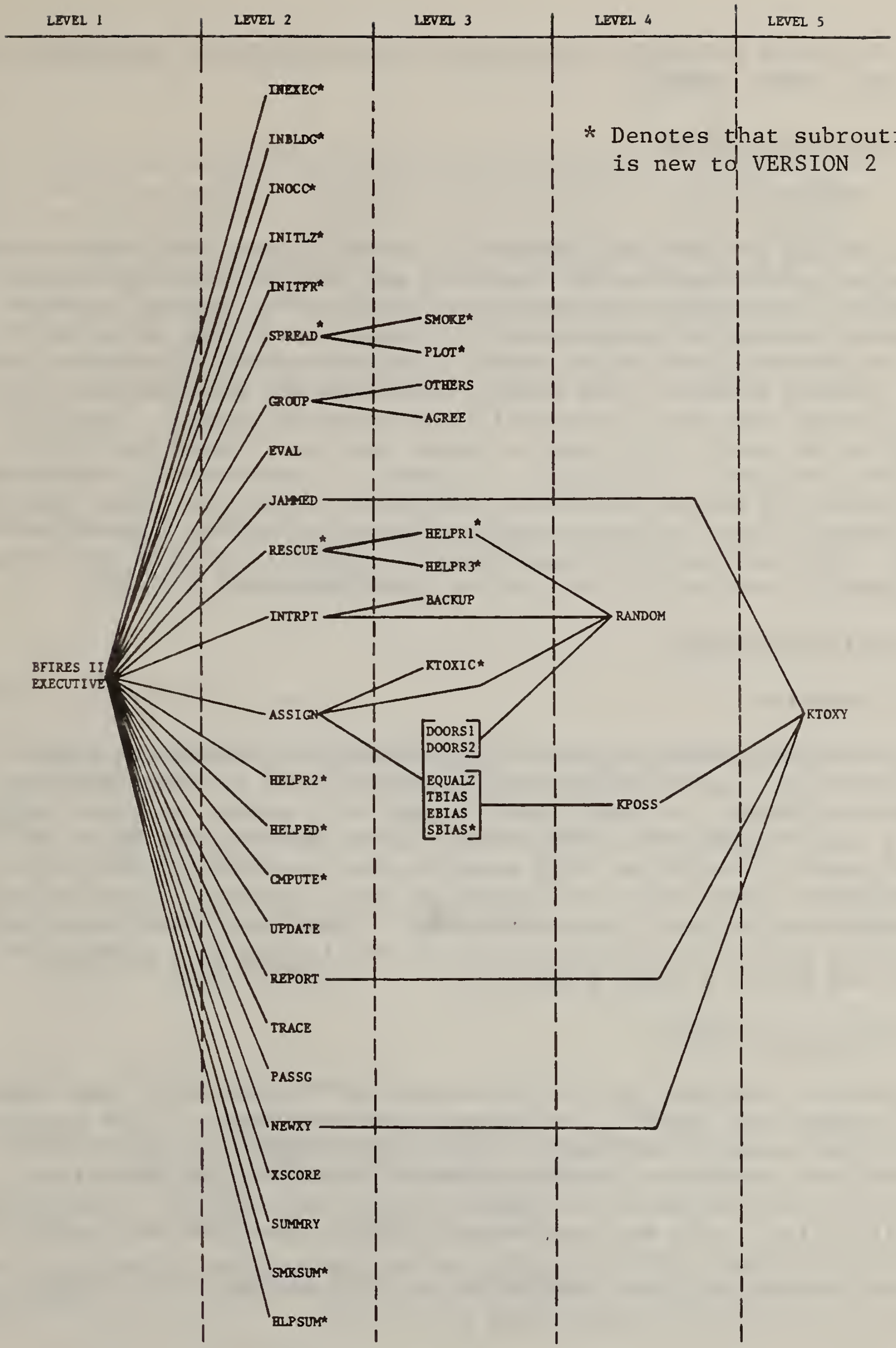
shortcoming of BFIRES I is its inability to simulate rescue activities, particularly those which might be required to remove nonambulatory occupants from healthcare facilities. Finally, the construction of BFIRES I data input files appears to be unecessarily complex from the user's viewpoint.

1.3 Specific Modifications of BFIRES I Included in BFIRES II and Covered by this Report

BFIRES II contains several new subroutines specifically designed to solve the problems described above. Most of these subroutines are clustered into two "modules." One module simulates direct interactions between occupants and toxicants (the SMOKE module), while the other addresses the problem of rescuing nonamubatory occupants (the RESCUE module). A number of additional subroutines enhance the input/output (I/O) capabilities of BFIRES. Finally BFIRES I subroutines EXEC and ASSIGN are restructured to accomodate the new modules and expand the program's I/O facilities.

New subroutines and modifications of existing ones are documented in Chapters 2 through 5. Subroutines are grouped for presentation in relation to their superordinate modules, and the function and operation of each subroutine is described. Flow diagrams and FORTRAN listings are also provided. The multi-level structure of BFIRES II is illustrated in Figure 1.1.[1]

[1] Figures and Tables appear at the ends of chapters to which they refer.



* Denotes that subroutine is new to VERSION 2

Figure 1.1 Multi-level Structure of BFIRES/VERSION 2

2.0 SIMULATION OF DIRECT INTERACTIONS BETWEEN OCCUPANTS AND SMOKE: THE "SMOKE" MODULE

2.1 Subroutine SPREAD

2.1.1 Function

The functions of subroutine SPREAD are to (1) determine the rate of smoke migration in the simulated environment, and (2) affect simulated smoke migration. To simplify the physical aspects of the problem, movement of toxicants through the environment is treated as a simple diffusion process, by which a point source expands into an increasingly larger "smoke" zone, as time advances. The rate of expansion may be adjusted by the user, through the specification of input parameter ISRATE. Thus, if ISRATE is set equal to 1, then the smoke zone will expand one spatial unit per time unit[1]. Similarly, if ISRATE is specified as 3, then the smoke zone will expand one spatial unit every three time units, and so on. SPREAD affects simulated smoke migration by calling subroutine SMOKE. This routine is discussed below. Refer to the flow chart shown in Figure 2.1 and the Fortran listing given in Table 2.1.

2.2 Subroutine SMOKE

2.2.1 Function

This subroutine causes person-occupiable spatial locations to become infiltrated by "smoke" (toxic fire products), at a rate determined by subroutine SPREAD. Although smoke migration is assumed to occur radially from a point source, it may be blocked by the intrusion of walls or closed doors. Subroutine SMOKE scans spatial locations in the path of the expanding smoke zone, and determines which potential locations will be blocked during the current time frame. Thus, only those points not blocked by walls or closed doors will be infiltrated at a given point in time. See Figure 2.2 and Table 2.2.

2.3 Subroutine KTOXIC

This routine maintains an up-to-date count of the number of time frames each occupant has spent in the smoke-filled environment. This function is provided under the assumption that as the length of time during which a person has occupied a toxic environment increases, the deleterious effects of the toxicants upon cognitive and motor behavior increase as well (Phillips, 1973; Berl and Halpin, 1978). Questions pertaining to visibility through smoke and to the actual composition of toxicants are separate issues which are treated neither within KTOXIC nor BFIRESII in general. See Figure 2.3 and Table 2.3

[1] This is acknowledged to be an oversimplification of smoke migration, which may vary in rate, direction and intensity over time.

2.4 Subroutine SBIAS

2.4.1 Function

This program biases an occupant's decision making behavior to favor movement toward smoke-free spatial locations. The occupant's immediate movement options are first scanned, and the location of infiltrated alternatives are identified. The probabilities of selecting the various options (first specified by subroutine TBIAS or EBIAS) are then reformulated, biasing in favor of the noninfiltrated spatial locations. This is accomplished by reducing the liklihood that the occupant will move toward a smoke-filled point, and by then redistributing the difference among noninfiltrated locations (so that the sum of the probabilities for all spatial moves is 1.00).

2.4.2 Computational Formulas

The reduction factor is computed by the equation

$$\mathrm{REDN(K) = P(K)/TSMOKE(N)} \qquad (1)$$

where:

REDN(K) = reduction factor,

P(K) = the probability that occupant$_n$ will move toward location K during time t (see Stahl, 1979, pp. 2-5), and

TSMOKE(N) = occupant$_n$'s tolerance for occupying a smoke-filled environment.

The distribution factor is computed by the equation

$$\mathrm{DISTR = SUMRED/NUMCLR} \qquad (2)$$

where:

DISTR = distribution factor,

SUMRED = the sum of the reduction factors computed for all smoke-infiltrated points, and

NUMCLR = the number of smoke-free points.

The revised probability of moving toward a smoke-free point is computed by the equation.

$$\mathrm{PNEW = P(K) + DISTR} \qquad (3)$$

where:

PNEW = the revised probability that occupant_n will approach location K during t,

P(K) = the original probability that location K will be approached during t, and

DISTR = the distribution factor.

The revised probability of moving toward a smoke-infiltrated location is computed by the equation

$$PNEW = P(K) - REDN(K) \tag{4}$$

where: PNEW = the revised probability that occupant_n will approach location K during t,

P(K) = the original probability that location K will be approached during t, and

REDN(K) = the reduction factor for location K.

Refer to Figure 2.4 and Table 2.4.

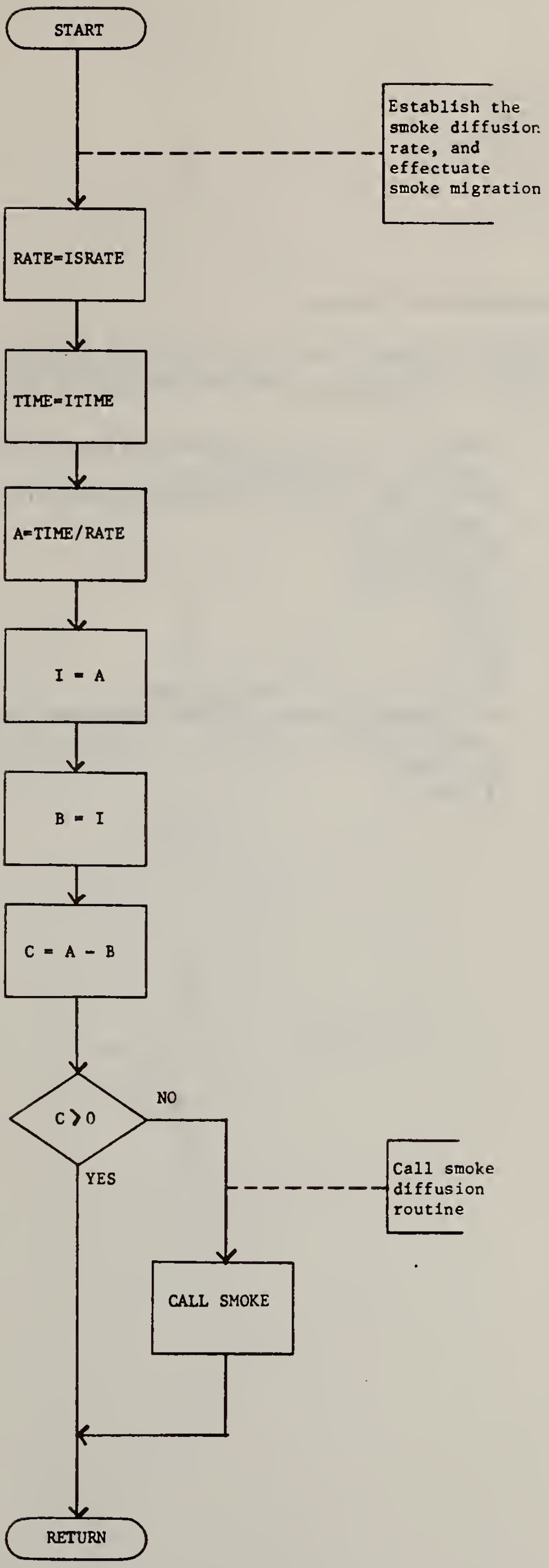

START
Establish the smoke diffusion rate, and effectuate smoke migration
RATE=ISRATE
TIME=ITIME
A=TIME/RATE
I = A
B = I
C = A - B
C > 0
NO
YES
Call smoke diffusion routine
CALL SMOKE
RETURN

Figure 2.1 Flow Chart for Subroutine SPREAD

```
C
C
C
C
C
C  SUBROUTINE SPREAD
C
C  THIS ROUTINE DETERMINES RATE OF SMOKE SPREAD, BASED ON A USER
C  INPUT PARAMETER.  THE ROUTINE THEN CALLS 'SMOKE', WHICH SPREADS
C  SMOKE AS REQUIRED.
C
      SUBROUTINE SPREAD (ISMOKE,IBAR,IDOOR,NPOINT,MAXX,MAXY,
     1  NSPACE,ND,ISRATE,ITIME)
      DIMENSION ISMOKE(100,100),IBAR(20,75,2),IDOOR(30,4),
     1  NPOINT (20)
      RATE=ISRATE
      TIME=ITIME
      A=TIME/RATE
      I=A
      B=I
      C=A-B
      IF (C.GT.0.001) GO TO 100
      CALL SMOKE (ISMOKE,IBAR,IDOOR,NPOINT,MAXX,MAXY,
     1  NSPACE,ND)
100   CONTINUE
      RETURN
      END
```

Table 2.1 Subroutine SPREAD: FORTRAN Listing

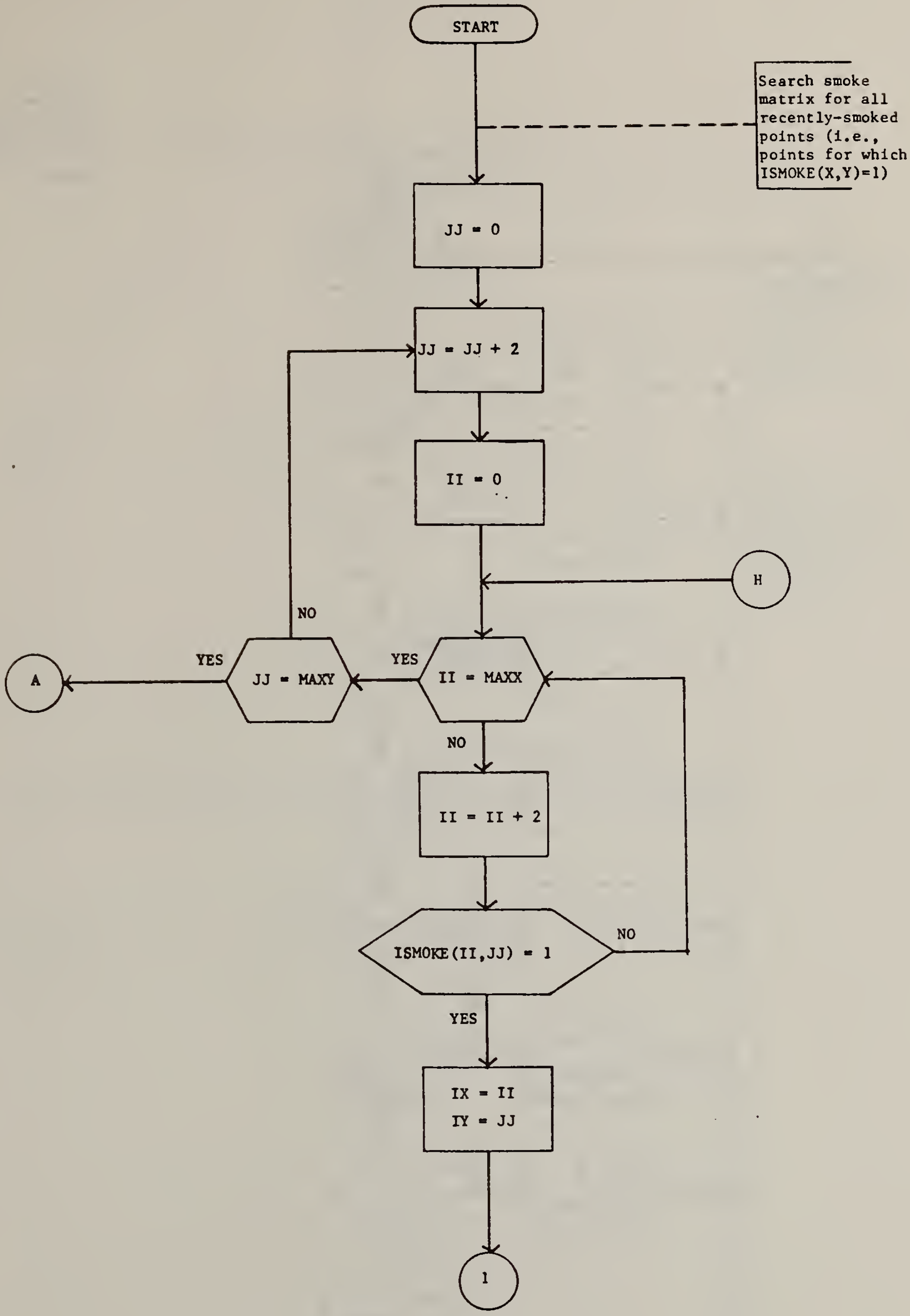
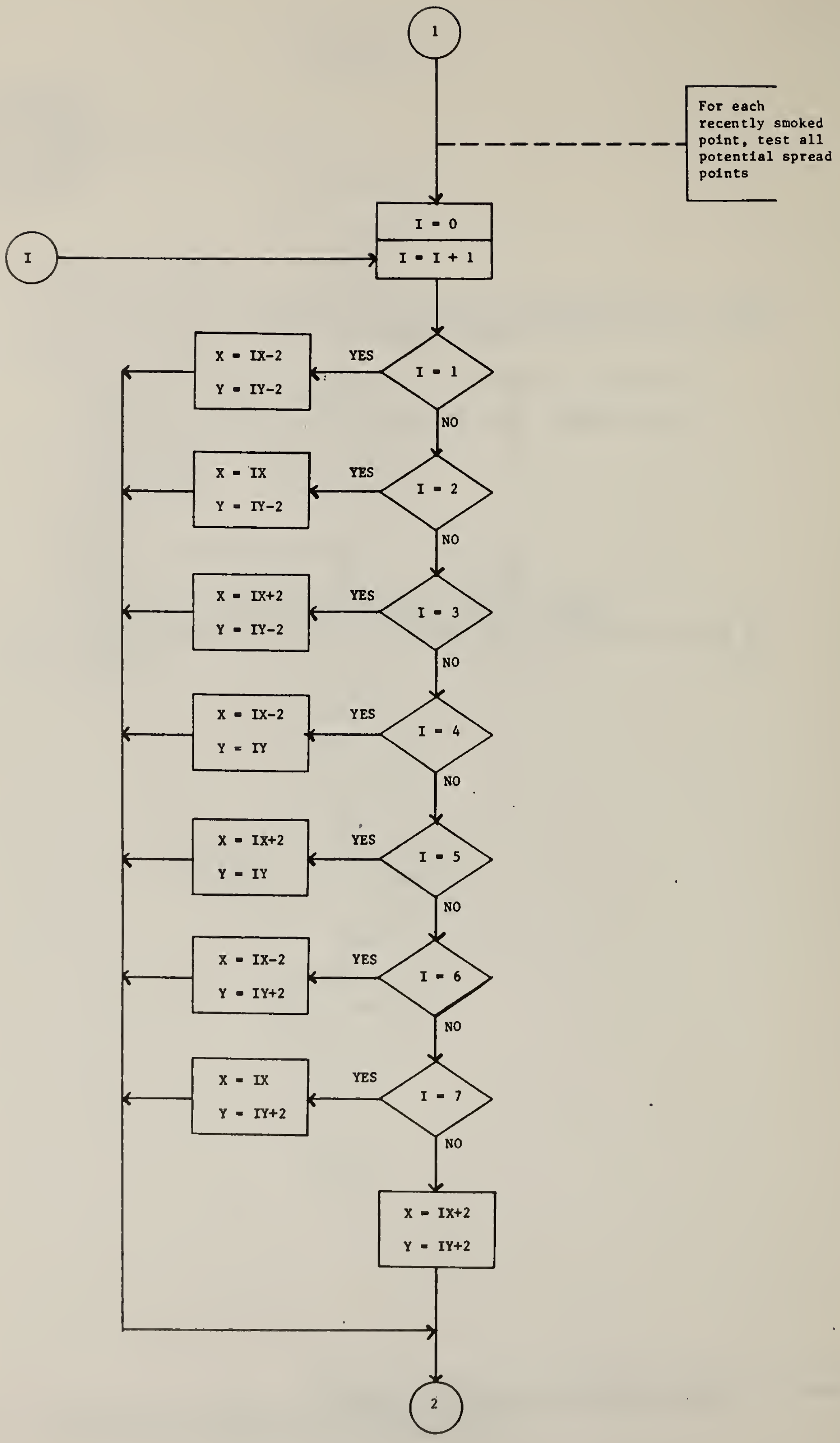
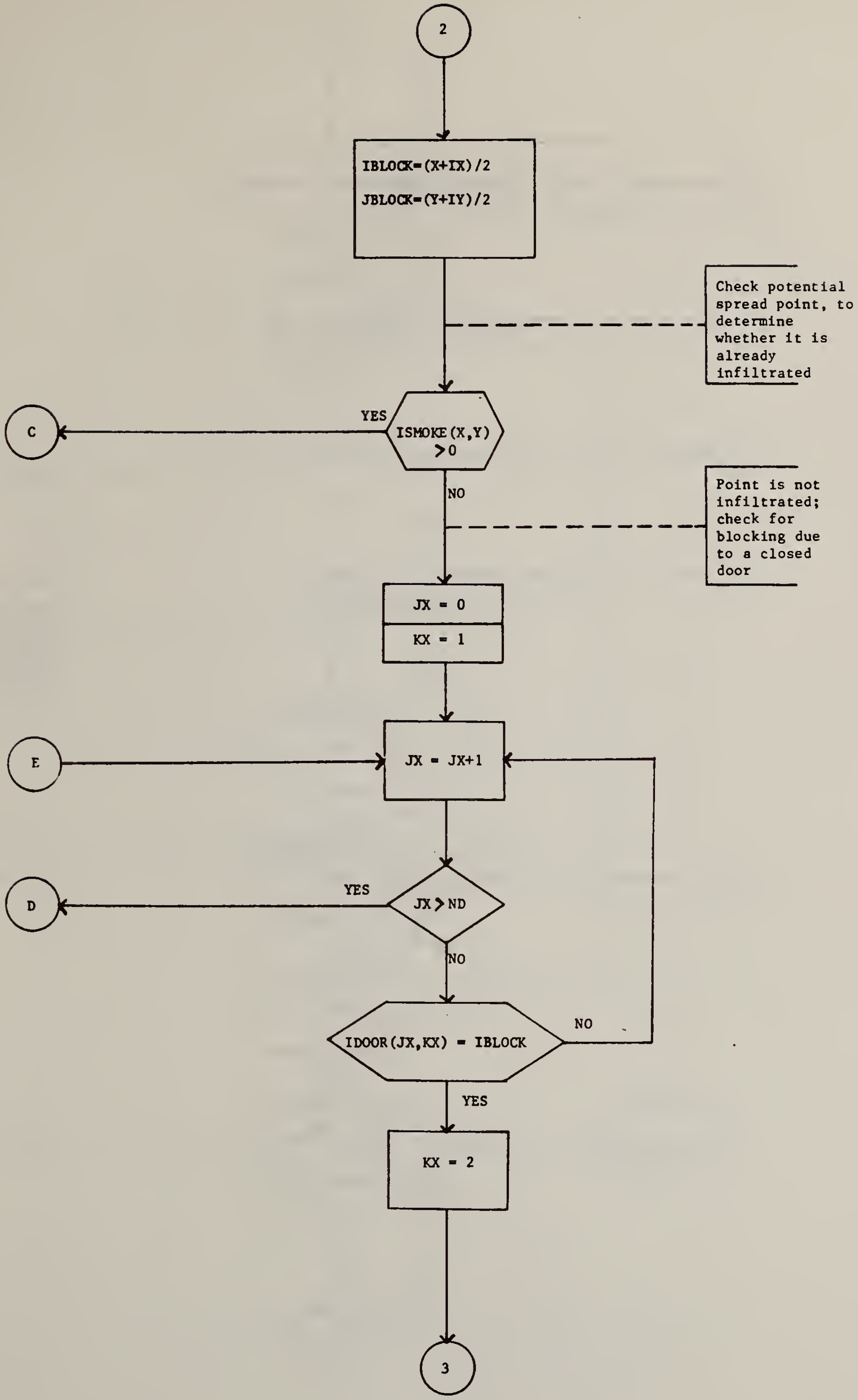
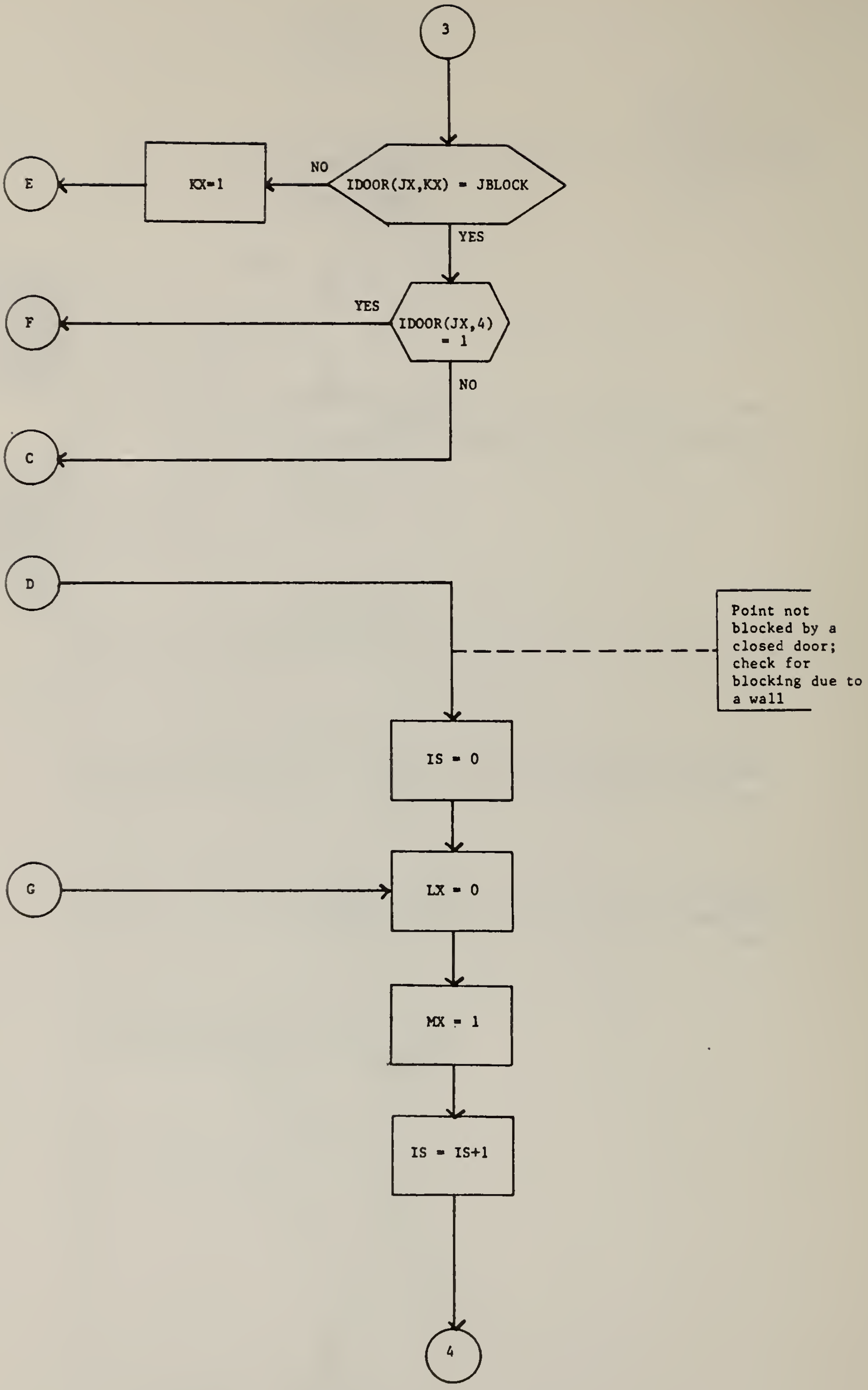
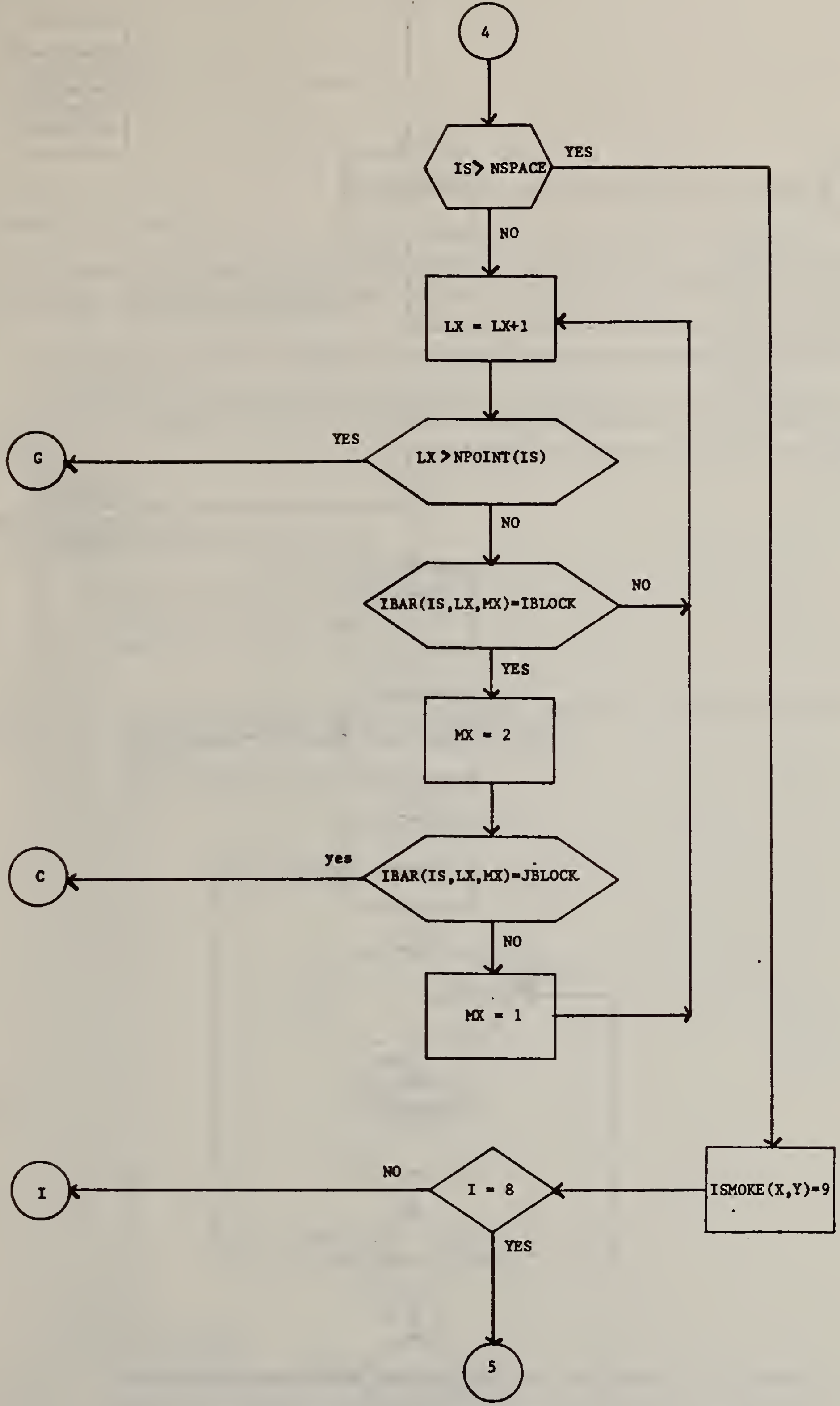


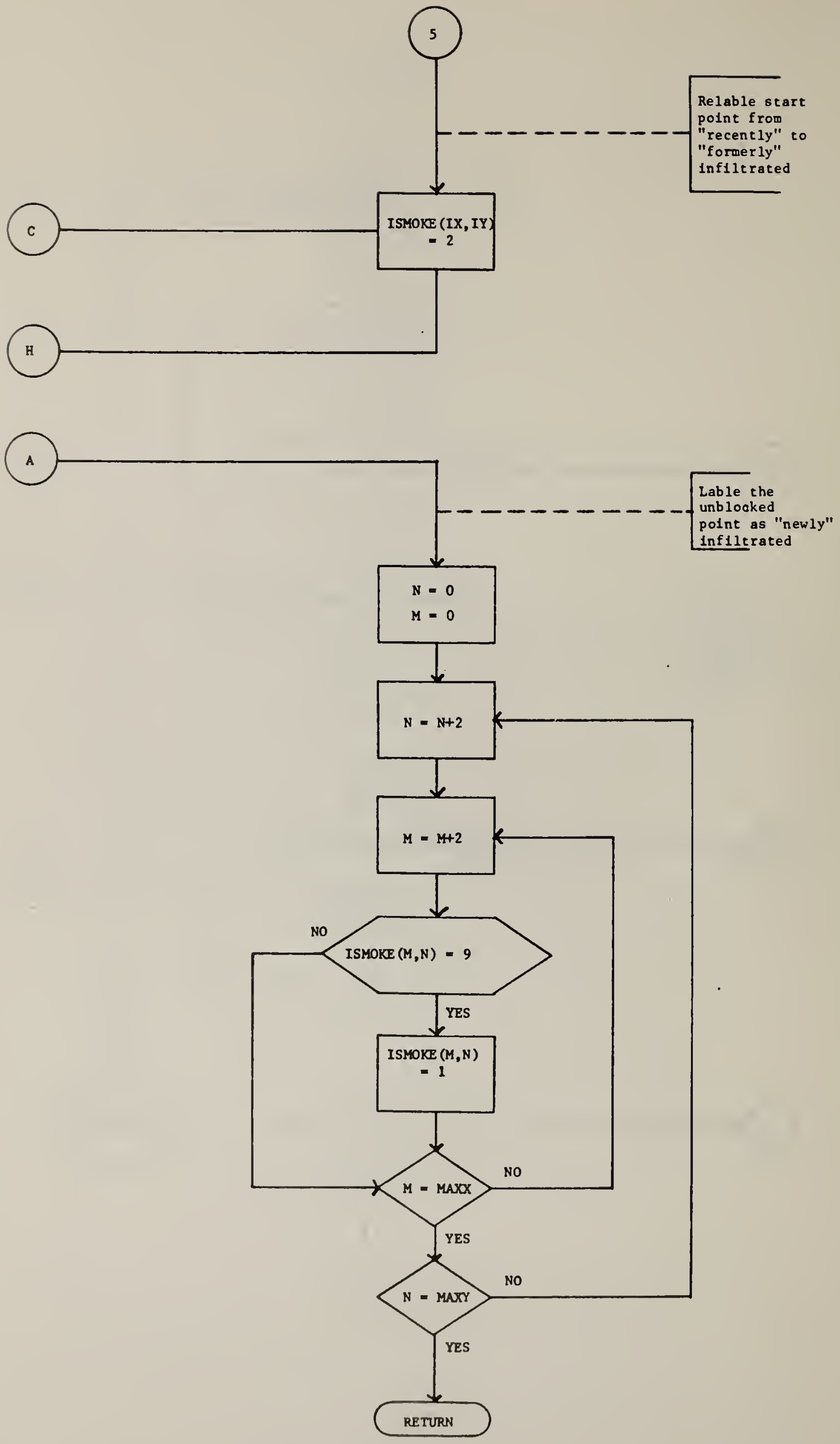
Figure 2.2 Flow Chart for Subroutine SMOKE


1
For each recently smoked point, test all potential spread points
I = 0
I = I + 1
I
I = 1
YES
X = IX-2
Y = IY-2
NO
I = 2
YES
X = IX
Y = IY-2
NO
I = 3
YES
X = IX+2
Y = IY-2
NO
I = 4
YES
X = IX-2
Y = IY
NO
I = 5
YES
X = IX+2
Y = IY
NO
I = 6
YES
X = IX-2
Y = IY+2
NO
I = 7
YES
X = IX
Y = IY+2
NO
X = IX+2
Y = IY+2
2


2
IBLOCK=(X+IX)/2
JBLOCK=(Y+IY)/2
Check potential spread point, to determine whether it is already infiltrated
YES
C
ISMOKE(X,Y) >0
NO
Point is not infiltrated; check for blocking due to a closed door
JX = 0
KX = 1
E
JX = JX+1
YES
D
JX > ND
NO
IDOOR(JX,KX) = IBLOCK
NO
YES
KX = 2
3


3
IDOOR(JX,KX) = JBLOCK
NO
KX=1
E
YES
IDOOR(JX,4)
= 1
YES
F
NO
C
D
Point not blocked by a closed door; check for blocking due to a wall
IS = 0
G
LX = 0
MX = 1
IS = IS+1
4


4
IS> NSPACE
YES
NO
LX = LX+1
YES
G
LX > NPOINT(IS)
NO
IBAR(IS,LX,MX)=IBLOCK
NO
YES
MX = 2
yes
C
IBAR(IS,LX,MX)=JBLOCK
NO
MX = 1
NO
I
I = 8
ISMOKE(X,Y)=9
YES
5


5
Relable start point from "recently" to "formerly" infiltrated
C
ISMOKE(IX,IY) = 2
H
A
Lable the unbloacked point as "newly" infiltrated
N = 0
M = 0
N = N+2
M = M+2
NO
ISMOKE(M,N) = 9
YES
ISMOKE(M,N) = 1
NO
M = MAXX
YES
NO
N = MAXY
YES
RETURN

```
C
C
C
C
C  SUBROUTINE SMOKE
C
C  THIS SUBROUTINE CAUSES PERSON-OCCUPIABLE LOCATIONS TO BECOME INFILTRATED
C  BY SMOKE.  SMOKE MIGRATION MAY BE BLOCKED BY WALLS (DETERMINISTIC), AND
C  BY CLOSED DOORS (STOCHASTIC).
C
      SUBROUTINE SMOKE (ISMOKE,IBAR,IDOOR,NPOINT,MAXX,MAXY,NSPACE,ND)
      INTEGER X,Y
      DIMENSION ISMOKE(100,100),IDOOR(30,4),IBAR(20,75,2),NPOINT(20)
C  SEARCH SMOKE MATRIX FOR ALL RECENTLY SMOKED POINTS, I.E., POINTS FOR WHICH
C  ISMOKE=1
      JJ=0
300   JJ=JJ+2
      II=0
301   IF (II.EQ.MAXX) GO TO 302
      II=II+2
      IF (ISMOKE(II,JJ).EQ.1) GO TO 101
      GO TO 301
302   IF (JJ.EQ.MAXY) GO TO 100
      GO TO 300
101   IX=II
      IY=JJ
C  FOR EACH RECENTLY SMOKED POINT, TEST ALL POTENTIAL-SPREAD-POINTS.  SPREAD
C  SMOKE AS DETERMINED BY BLOCKING CONDITIONS
      DO 200 I=1,8
      GO TO (1,2,3,4,5,6,7,8),I
1     X=IX-2
      Y=IY-2
      GO TO 9
2     X=IX
      Y=IY-2
      GO TO 9
3     X=IX+2
      Y=IY-2
      GO TO 9
4     X=IX-2
      Y=IY
      GO TO 9
5     X=IX+2
      Y=IY
      GO TO 9
6     X=IX-2
      Y=IY+2
         GO TO 9
7     X=IX
      Y=IY+2
      GO TO 9
8     X=IX+2
      Y=IY+2
9     CONTINUE
      IBLOCK=(X+IX)/2
      JBLOCK=(Y+IY)/2
C  CHECK POTENTIAL-SPREAD-POINT TO DETERMINE WHETHER IT IS ALREADY SMOKED.
      IF (ISMOKE(X,Y).GT.0) GO TO 200
C  CHECK POTENTIAL-SPREAD-POINT FOR BLOCKING DUE TO A CLOSED DOOR
      JX=0
```

Table 2.2 Subroutine SMOKE: FORTRAN Listing

```
      KX=1
   11 JX=JX+1
      IF (HX.GT.ND) GO TO 41
      IF (IDOOR(JX,KX).EQ.IBLOCK) GO TO 10
      GO TO 11
   10 KX=2
      IF (IDOOR(JX,KX).EQ.JBLOCK) GO TO 12
      KX=1
      GO TO 11
   12 IF (IDOOR(JX,4).EQ.1) GO TO 50
      GO TO 200
   41 CONTINUE
C  CHECK FOR BLOCKING DUE TO A WALL
      IS=0
   15 LX=0
      MX=1
   13 IS=IS+1
      IF (IS.GT.NSPACE) GO TO 42
   14 LX=LX+1
      IF (LX.GT.NPOINT(IS)) GO TO 15
      IF (IBAR(IS,LX,MX).EQ.IBLOCK) GO TO 16
      GO TO 14
   16 MX=2
      IF (IBAR(IS,LX,MX).EQ.JBLOCK) GO TO 200
      MX=1
      GO TO 14
   42 CONTINUE
   50 ISMOKE(X,Y)=9
  200 CONTINUE
C  RE-LABEL STARTING POINT FROM RECENTLY-SMOKED TO FORMERLY-SMOKED
      ISMOKE(IX,IY)=2
      GO TO 301
C  LABEL UNBLOCKED POINT AS NEWLY SMOKED
  100 CONTINUE
      DO 400 N=2,MAXY,2
      DO 400 M=2,MAXX,2
      IF (ISMOKE(M,N).EQ.9) GO TO 401
      GO TO 400
  401 ISMOKE(M,N)=1
  400 CONTINUE
      RETURN
      END
```

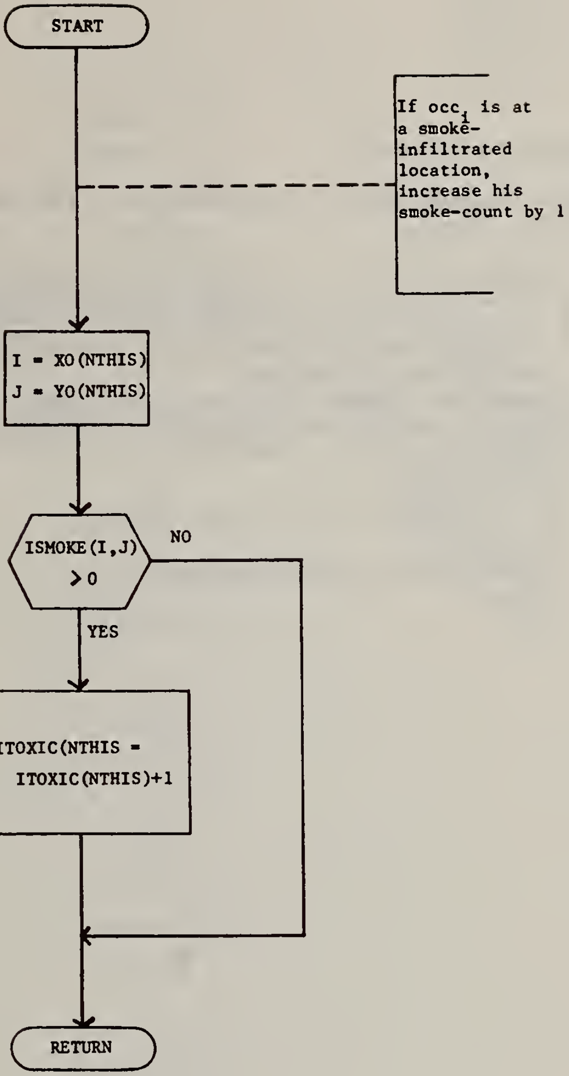

START
If occ_i is at a smoke-infiltrated location, increase his smoke-count by 1
I = XO(NTHIS)
J = YO(NTHIS)
ISMOKE(I,J) > 0
NO
YES
ITOXIC(NTHIS = ITOXIC(NTHIS)+1
RETURN

Figure 2.3 Flow Chart for Subroutine KTOXIC

```
C
C
C
C
C  SUBROUTINE KTOXIC
C
C  THIS PROGRAM COUNTS THE NUMBER OF TIME FRAMES EACH OCCUPANT HAS SPENT AT
C  SMOKED POINTS.
C
      SUBROUTINE KTOXIC (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1  XO,YO,IBAR,XT,YT,NAGREE,XE,YE,IAGREE,IRAND,P,MOVE,XK,
     2  YK,K,L,IS,IGOALX,IGOALY,IENTER,X,IDOOR,POPEN,ND,MDOOR,
     3  PCLOSE,NOEXP,LOWEXP,IHIEXP,ITOTAL,ISMOKE,TSMOKE,ITOXIC)
      INTEGER XO(20),YO(20)
      DIMENSION ISMOKE(100,100),ITOXIC(20)
C  DETERMINE WHETHER THE OCCUPANT IS CURRENTLY AT A SMOKED POINT.  IF SO,
C  INCREASE HIS TOTAL COUNT BY 1.
      I=XO(NTHIS)
      J=YO(NTHIS)
      IF (ISMOKE(I,J).GT.0) GO TO 100
      GO TO 110
100   ITOXIC(NTHIS)=ITOXIC(NTHIS)+1
110   RETURN
      END
```

Table 2.3 Subroutine KTOXIC: FORTRAN Listing

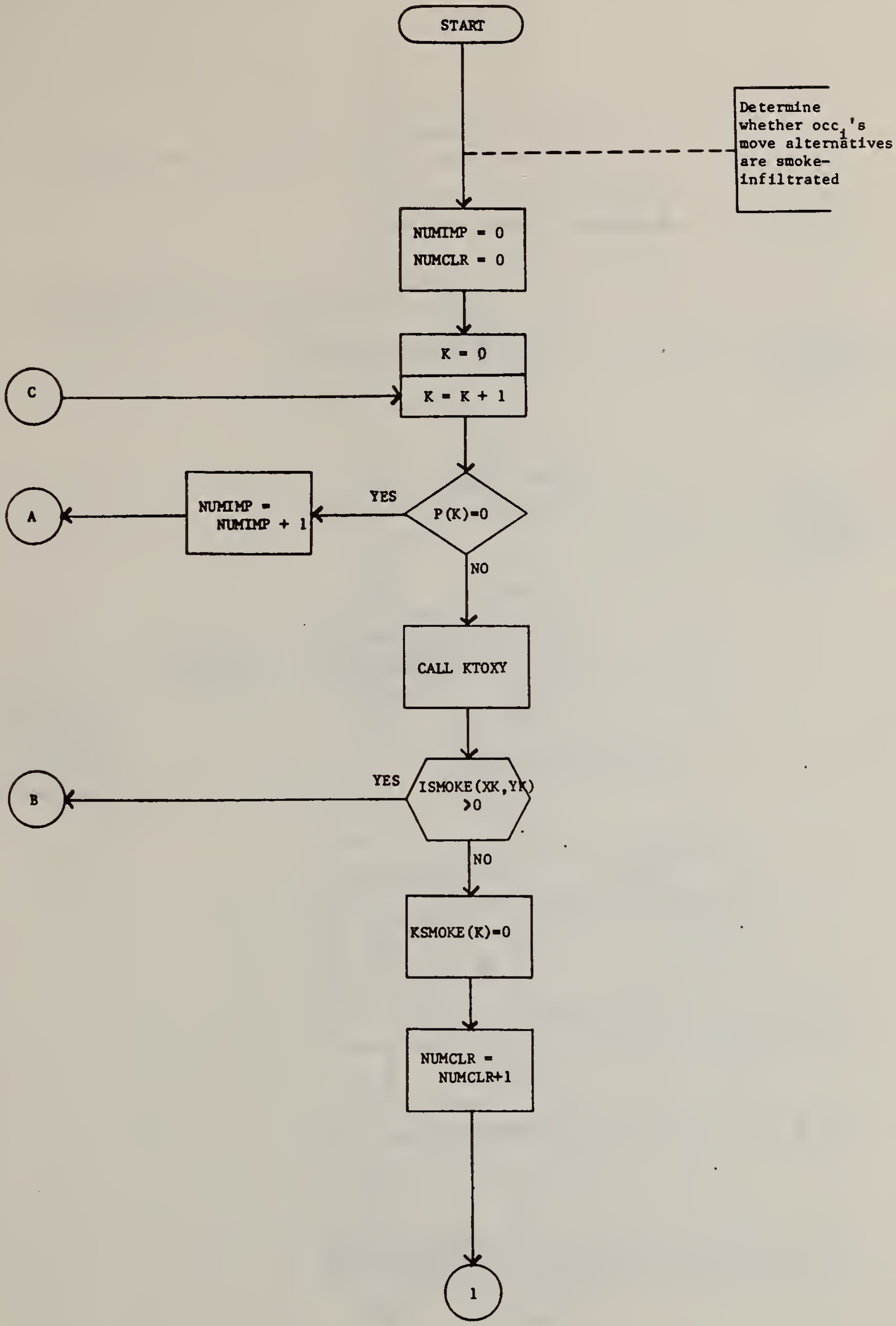
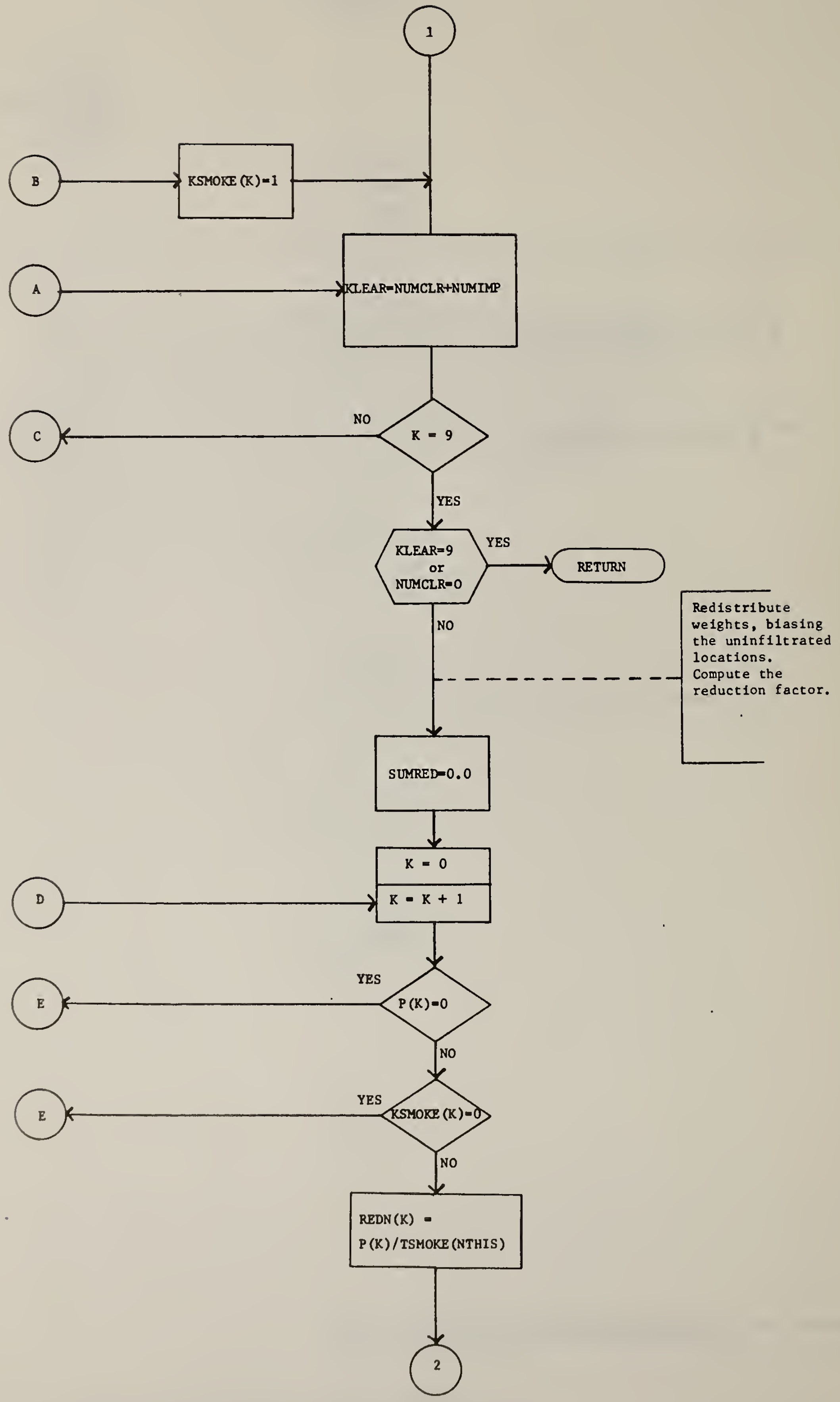
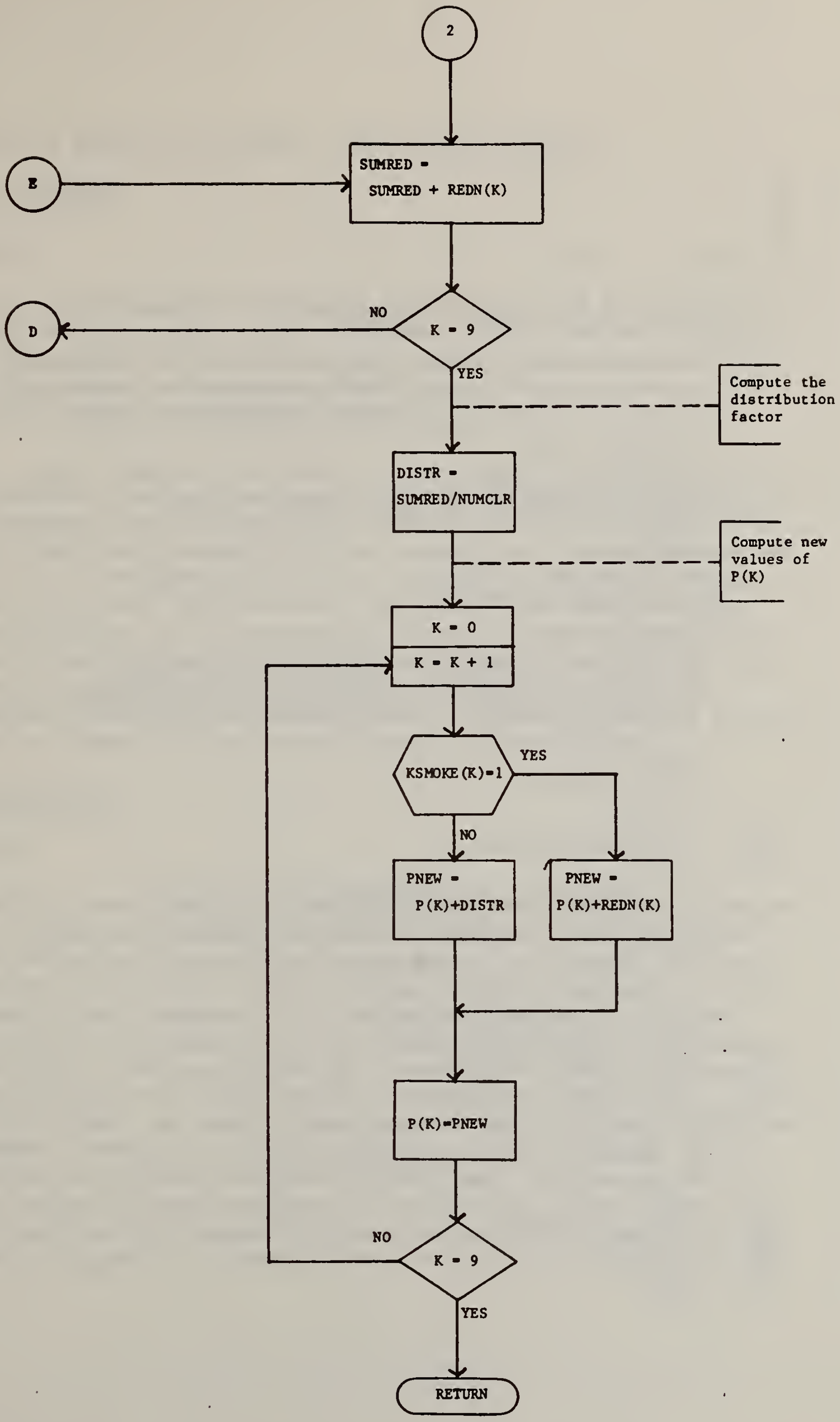


Figure 2.4 Flow Chart for Subroutine SBIAS


1
B
KSMOKE(K)=1
A
KLEAR=NUMCLR+NUMIMP
NO
C
K = 9
YES
KLEAR=9
or
NUMCLR=0
YES
RETURN
NO
Redistribute weights, biasing the uninfiltrated locations. Compute the reduction factor.
SUMRED=0.0
K = 0
D
K = K + 1
YES
E
P(K)=0
NO
YES
E
KSMOKE(K)=0
NO
REDN(K) =
P(K)/TSMOKE(NTHIS)
2


2
E
SUMRED =
SUMRED + REDN(K)
D
NO
K = 9
YES
Compute the distribution factor
DISTR =
SUMRED/NUMCLR
Compute new values of P(K)
K = 0
K = K + 1
KSMOKE(K)=1
YES
NO
PNEW =
P(K)+DISTR
PNEW =
P(K)+REDN(K)
P(K)=PNEW
NO
K = 9
YES
RETURN

```
C
C
C
C
C  SUBROUTINE SBIAS
C
C  THIS ROUTINE BIASES AN OCCUPANT'S DECISION MAKING TO
C  FAVOR NON-SMOKED SPATIAL LOCATIONS.
C
      SUBROUTINE SBIAS (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1  XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,IRAND,
     2  P,MOVE,XK,YK,K,L,IS,IGOALX,IGOALY,IENTER,X,IDOOR,
     3  POPEN,ND,MDOOR,PCLOSE,NOEXP,LOWEXP,IHIEXP,ITOTAL,
     4  ISMOKE,TSMOKE)
      DIMENSION P(9),ISMOKE(100,100),KSMOKE(9),REDN(9)
      INTEGER TSMOKE(20),XK,YK
C  DETERMINE WHETHER MOVE ALTERNATIVES ARE SMOKED
      NUMIMP=0
      NUMCLR=0
      DO 100 K=1,9
      IF (P(K).EQ.0.) GO TO 110
      GO TO 120
110   NUMIMP=NUMIMP+1
      GO TO 140
120   CALL KTOXY (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1  XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,IRAND,
     2  P,MOVE,XK,YK,K)
      IF (ISMOKE(XK,YK).GT.0) GO TO 130
      KSMOKE(K)=0
      NUMCLR=NUMCLR+1
      GO TO 140
130   KSMOKE(K)=1
140   KLEAR=NUMCLR+NUMIMP
100   CONTINUE
      IF ((KLEAR.EQ.9).OR.(NUMCLR.EQ.0)) GO TO 400
C  REDISTRIBUTE WEIGHTS, FAVORING UNSMOKED LOCATIONS
C   COMPUTE REDUCTION FACTOR, REDN(K), AND SUMRED
      SUMRED=0.
      DO 200 K=1,9
      IF (P(K).EQ.0.) GO TO 200
      IF (KSMOKE(K).EQ.0) GO TO 200
      REDN(K)=P(K)/TSMOKE(NTHIS)
200   SUMRED=SUMRED+REDN(K)
C  COMPUTE DISTR
      DISTR=SUMRED/NUMCLR
C  COMPUTE NEW P(K) VALUES
      DO 300 K=1,9
      IF (KSMOKE(K).EQ.1) GO TO 310
      PNEW=P(K)+DISTR
      GO TO 320
310   PNEW=P(K)-REDN(K)
320   P(K)=PNEW
300   CONTINUE
400   RETURN
      END
```

Table 2.4 Subroutine SBIAS: FORTRAN Listing

3.0 SIMULATION OF RESCUE ACTIVITIES: THE "RESCUE" MODULE

3.1 Subroutine RESCUE

3.1.1 Function

This routine is called whenever a preassigned rescuer (or "helper") is being processed, during a time frame. Helpers differ from other occupants in the sense that the user must specify for each helper the location of a single person-to-be-helped (PTBH), and a non-zero probability of actually engaging in the helping activity.

If the current time frame is the initial frame (i.e., if t=1), then RESCUE first establishes (probabilistically) whether helper_n engages in the rescue task (through a Call to subroutine HELPR1, described below). If not, then this simulated occupant will be treated as a non-helper during the remainder of the simulated fire event. If the helper does engage in the rescue activity, then repeated calls to RESCUE, as time advances, will serve the functions of (1) moving the helper to the location of the PTBH, and then (2) moving the helper-PTBH pair toward the exit goal[1]. These tasks are accomplished within RESCUE by calls to subroutines HELPER1 and HELPR3, which are discussed below. Subroutine RESCUE is further described in Figure 3.1 and Table 3.1.

3.2 Subroutine HELPR1

3.2.1 Function

During the initial time frame (t=1), RESCUE calls upon HELPER1 to establish whether a preassigned rescuer engages in helping. This is accomplished by comparing the user-defined probability that helper_n will in-fact engage in the rescue task (input parameter PHELP) against a random number generated by the computer. If the helper "decides" to rescue, then HELPR1 changes his movement goal to assure that subsequent movement decisions lead him to the location of his preassigned PTBH. As the simulated event progresses over time, HELPR1 maintains the helper on a proper course toward the PTBH. As with a pedestrian movement generated by other aspects of BFIRES, movement by a helper toward a PTBH is biased rather than fully determined. Thus, simulated helpers may traverse less-than-perfect routes while locating occupants to be rescued. This phenomenon is expected to conform with actual behavior during real emergencies, as noted in anecdotal reports of actual fires (refer also to Archea, 1979).

[1] The term "exit goal" may refer to entry into stair enclosure, passage through a horizontal exit, or to having reached some preassigned staging area.

3.2.2 Computational Formulas

Whether or not a helper engages in the rescue task is computed in the following manner:

$$q = PHELP(N) - X \qquad (5)$$

where:

q = either a positive, negative, or zero result, such that for negative and zero outcomes, the helper will not engage in rescue; for positive outcomes, the helper will rescue,

$PHELP(N)$ = the user-defined probability (between 0 and 1) that $helper_n$ will engage in the rescue task, and

(X) = a random number between 0 and 1 generated by the computer.

Refer to Figure 3.2 and Table 3.2.

3.3 Subroutine HELPR2

3.3.1 Function

This routine identifies the time frame during which a helper reaches his PTBH. See Figure 3.3 and Table 3.3

3.4 Subroutine HELPR3

3.4.1 Function

After a helper reaches his preassigned PTBH, three tasks must be accomplished. First, the helper's movement objective must be changed from the location of the PTBH back to the location of the exit goal. Second, the helper's movement speed must be slowed in recognition of the fact that assisting a nonambulatory occupant may be difficult and energy consuming (see Archea, 1979)[1]. Finally, the PTBH must be "moved along with" the rescuer, at the reduced rate of speed. This last task is accomplished through Subroutine HELPED, described below. See Figure 3.4 and Table 3.4.

3.5 Subroutine HELPED

3.5.1 Function

This routine causes the person-to-be-helped (PTBH) to remain with the assigned helper throughout the remainder of the simulated event. Under

[1] To simplify this problem, BFIRES assumes a 50% reduction of the helper's movement speed.

actual conditions, the helper would cause the PTBH to move, change direction, etc. BFIRES simulates this activity indirectly, by requiring the PTBH to follow-after the helper. Refer to Figure 3.5 Table 3.5. Finally, the helping function is limited to those situations where (1) the helper is familair with available egress routes, (2) the person-to-be-helped requires no preparation time once the helper has arrived to render assistance (as discussed by Archea, 1979), and (3) one helper is sufficient to move any nonambulatory occupant.

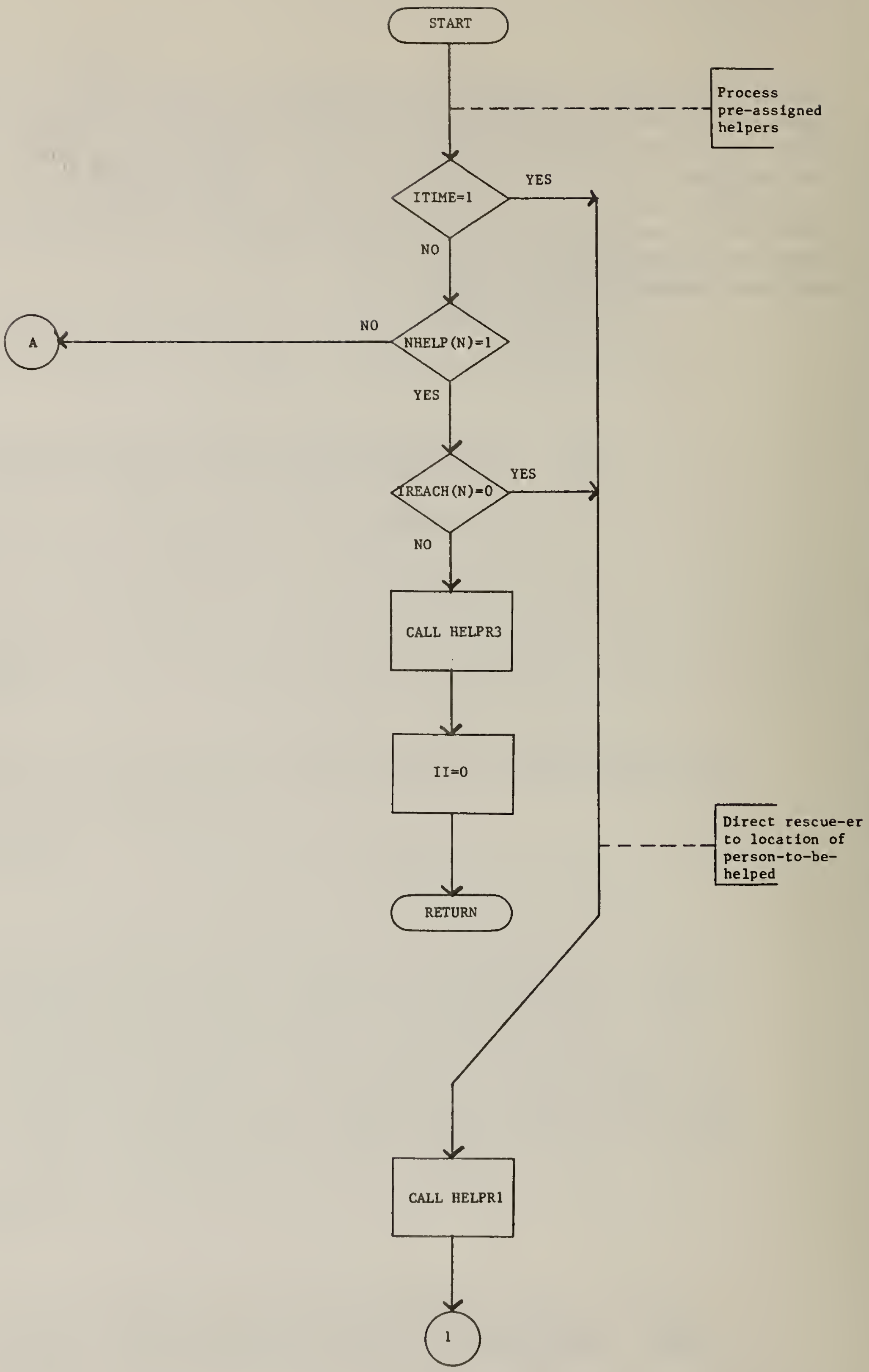
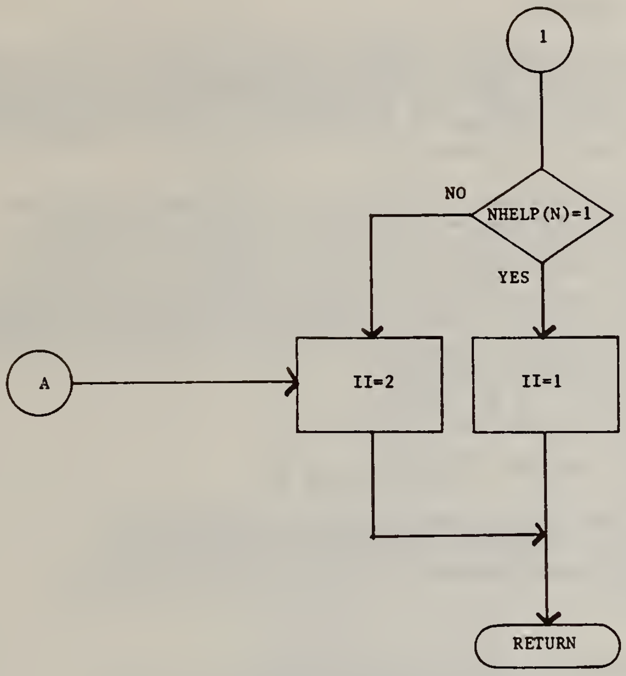


Figure 3.1 Flow Chart for Subroutine RESCUE


1
NO
NHELP(N)=1
YES
A
II=2
II=1
RETURN

```
C
C
C
C
C  SUBROUTINE RESCUE
C
C  THIS PROGRAM PROCESSES RESCUE-ER ACTIVITIES.  IT DETERMINES
C  WHETHER A PRE-ASSIGNED RESCUE-ER DECIDES TO ENGAGE IN RESCUING,
C  AND IF SO, DETERMINES THE COURSE OF THE RESCUE ACTIVITY.
C  BEHAVIOR BY THE PERSON TO BE HELPED (PTBH) IS GENERATED BY
C  SUBROUTINE HELPED, LATER ON.
C
      SUBROUTINE RESCUE (ITIME,NTHIS,PHELP,NHELP,XO,YO,
     1  IXHELP,IYHELP,NSPACE,XLO,XHI,YLO,YHI,NE,IGOALX,
     2  IGOALY,IAGREE,IGX,IGY,IAG,IXX,IREACH,IHANDI,II)
      DIMENSION PHELP(20),NHELP(20),IXHELP(20),IYHELP(20),
     1  NE(20),IGOALX(20,10,20),IGOALY(20,10,20),IAGREE(20),
     2  IGX(20),IGY(20),IAG(20),IXX(20),IREACH(20),IHANDI(20)
      INTEGER XO(20),YO(20),XLO(20),XHI(20),YLO(20),YHI(20)
C  PROCESS PRE-ASSIGNED HELPERS:
      IF (ITIME.EQ.1) GO TO 20
      IF (NHELP(NTHIS).EQ.1) GO TO 15
      GO TO 10
15    IF (IREACH(NTHIS).EQ.0) GO TO 20
      GO TO 25
25    CALL HELPR3 (ITIME,NTHIS,PHELP,NHELP,XO,YO,
     1  IXHELP,IYHELP,NSPACE,XLO,XHI,YLO,YHI,NE,
     2  IGOALX,IGOALY,IAGREE,IGX,IGY,IAG,IXX,IREACH,
     3  IHANDI)
35    II=0
      RETURN
C  DIRECT RESCUER TO LOCATION OF PTBH:
20    CALL HELPR1 (ITIME,NTHIS,PHELP,NHELP,XO,YO,
     1  IXHELP,IYHELP,NSPACE,XLO,XHI,YLO,YHI,NE,
     2  IGOALX,IGOALY,IAGREE,IGX,IGY,IAG,IXX)
      IF (NHELP(NTHIS).EQ.1) GO TO 30
      GO TO 10
30    II=1
      RETURN
10    II=2
      RETURN
      END
```

Table 3.1 Subroutine RESCUE: FORTRAN Listing

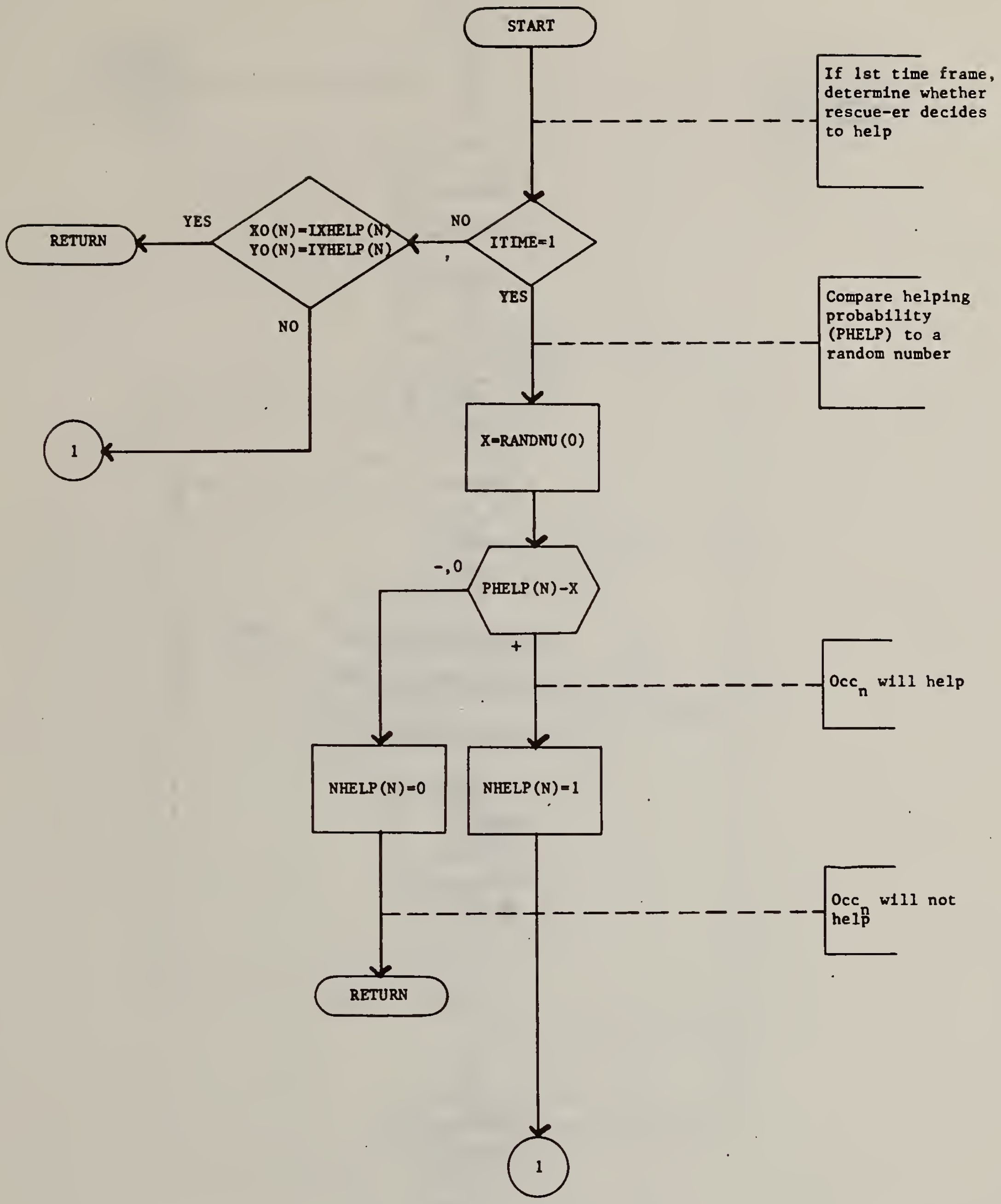
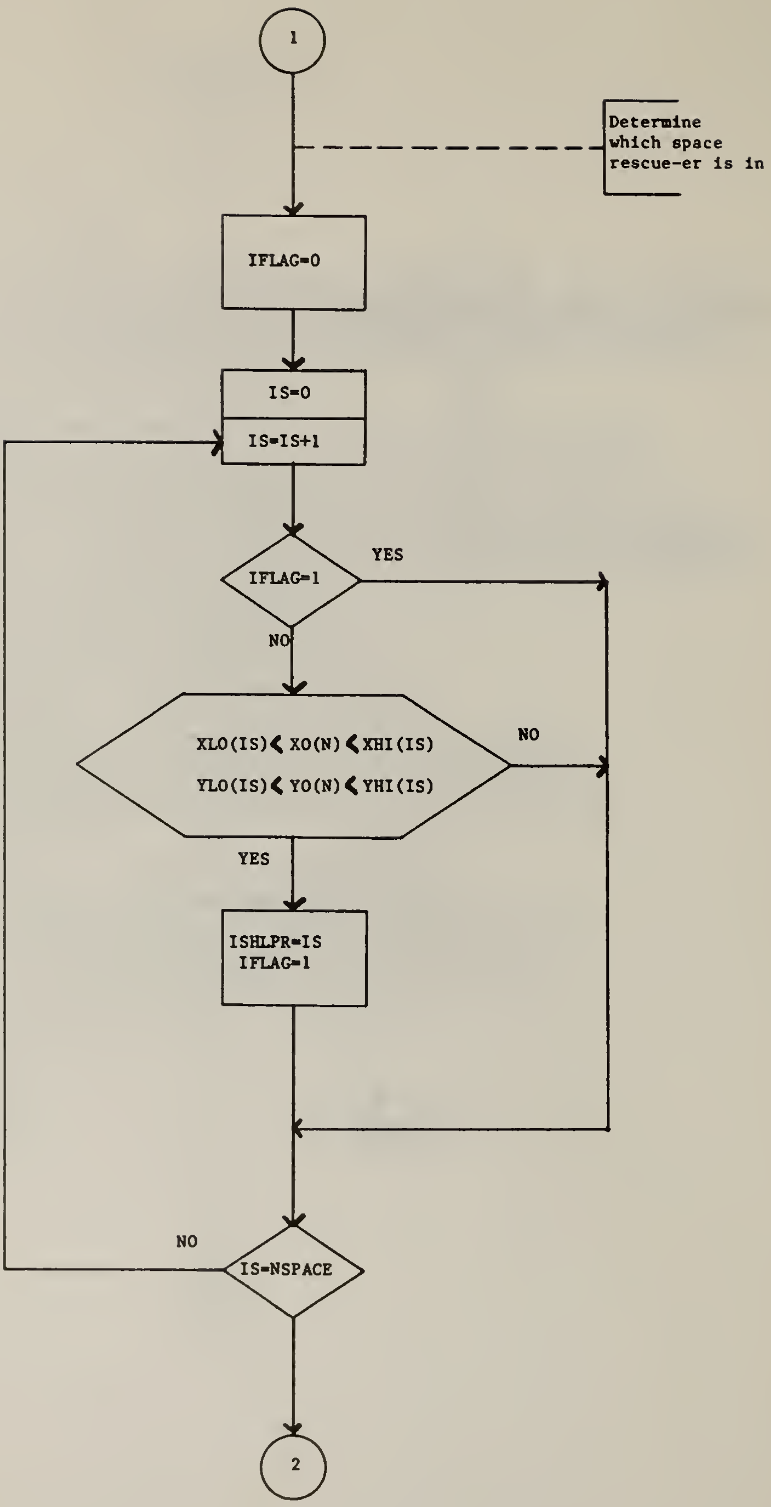
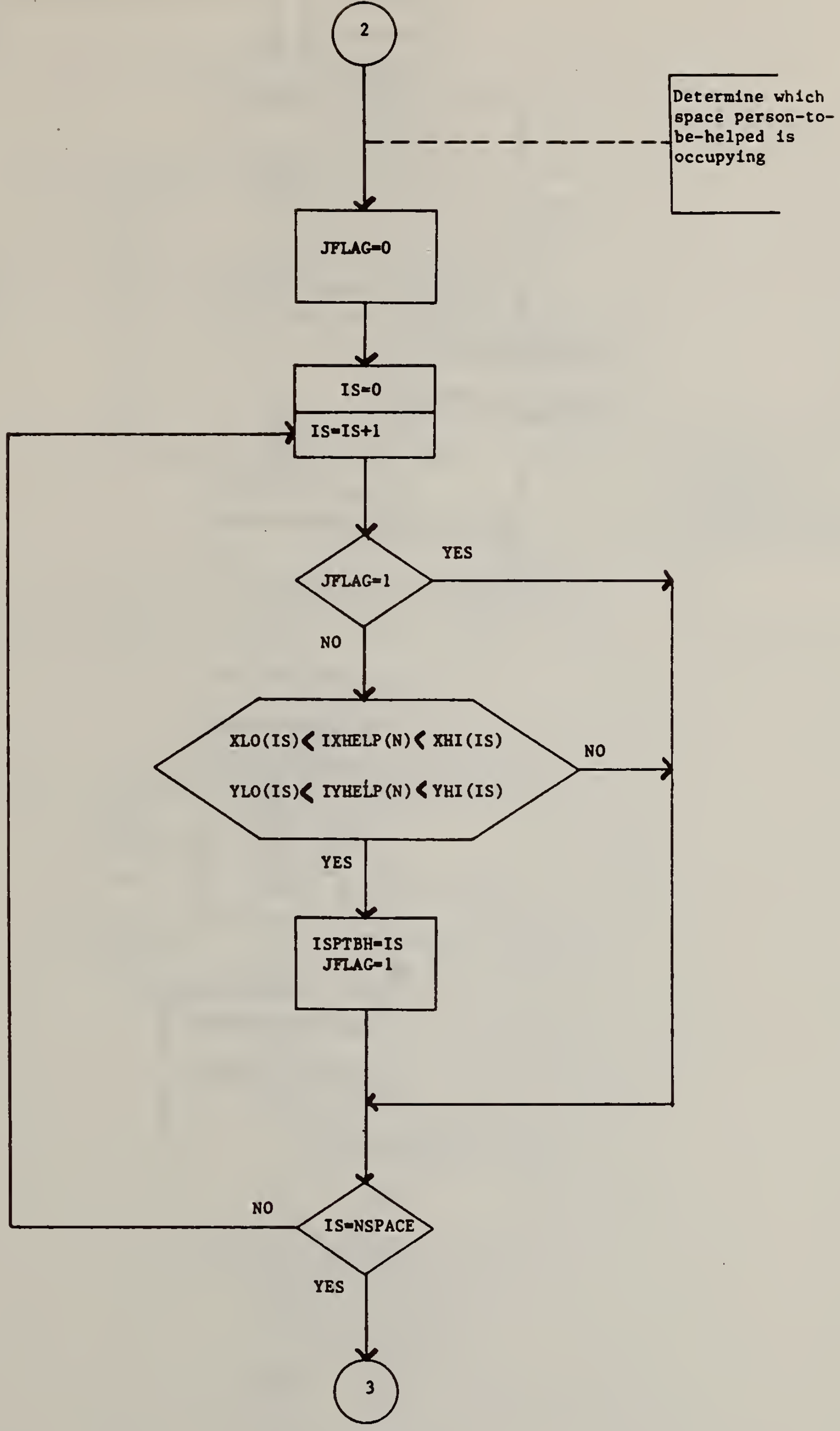
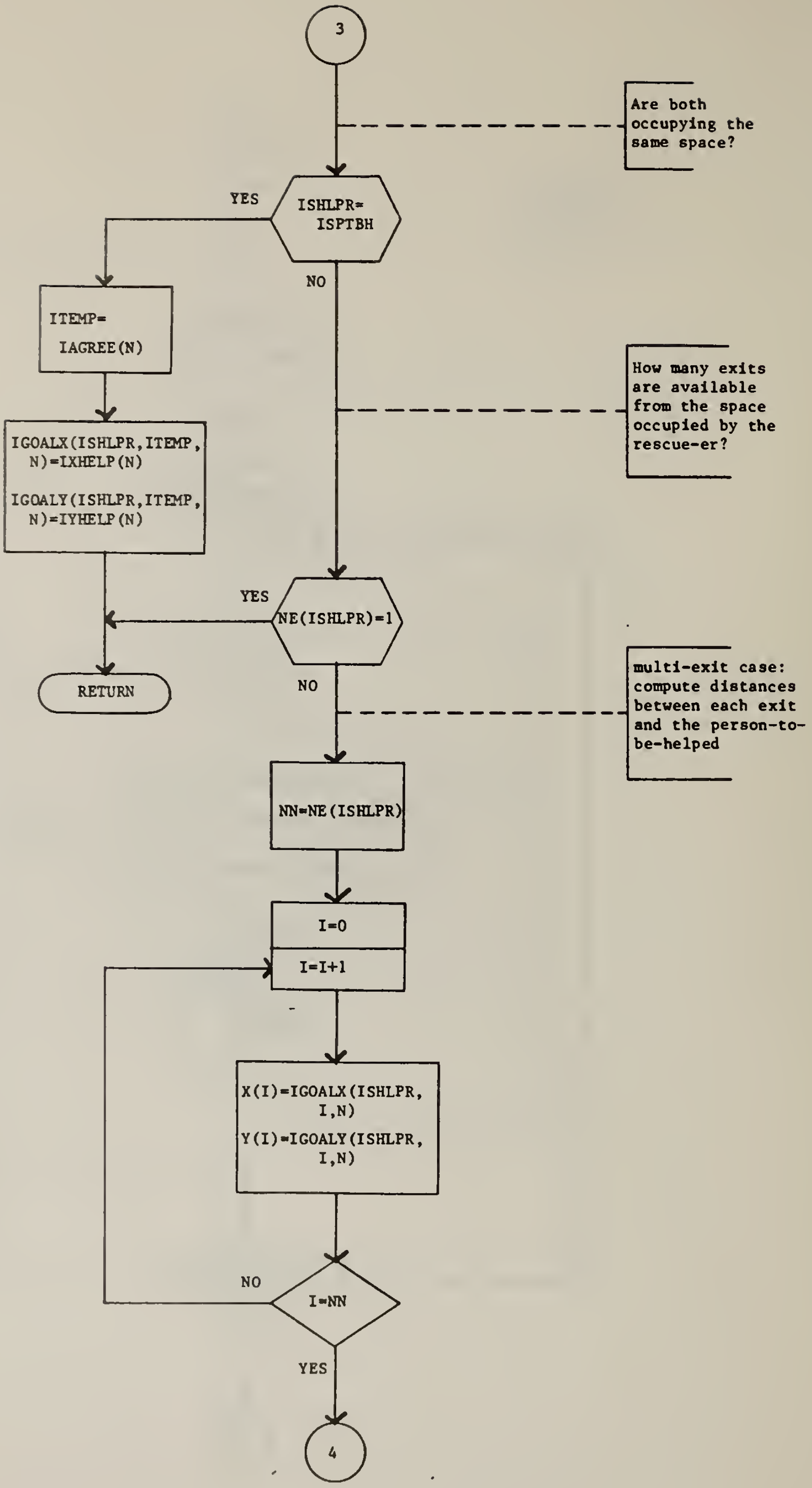
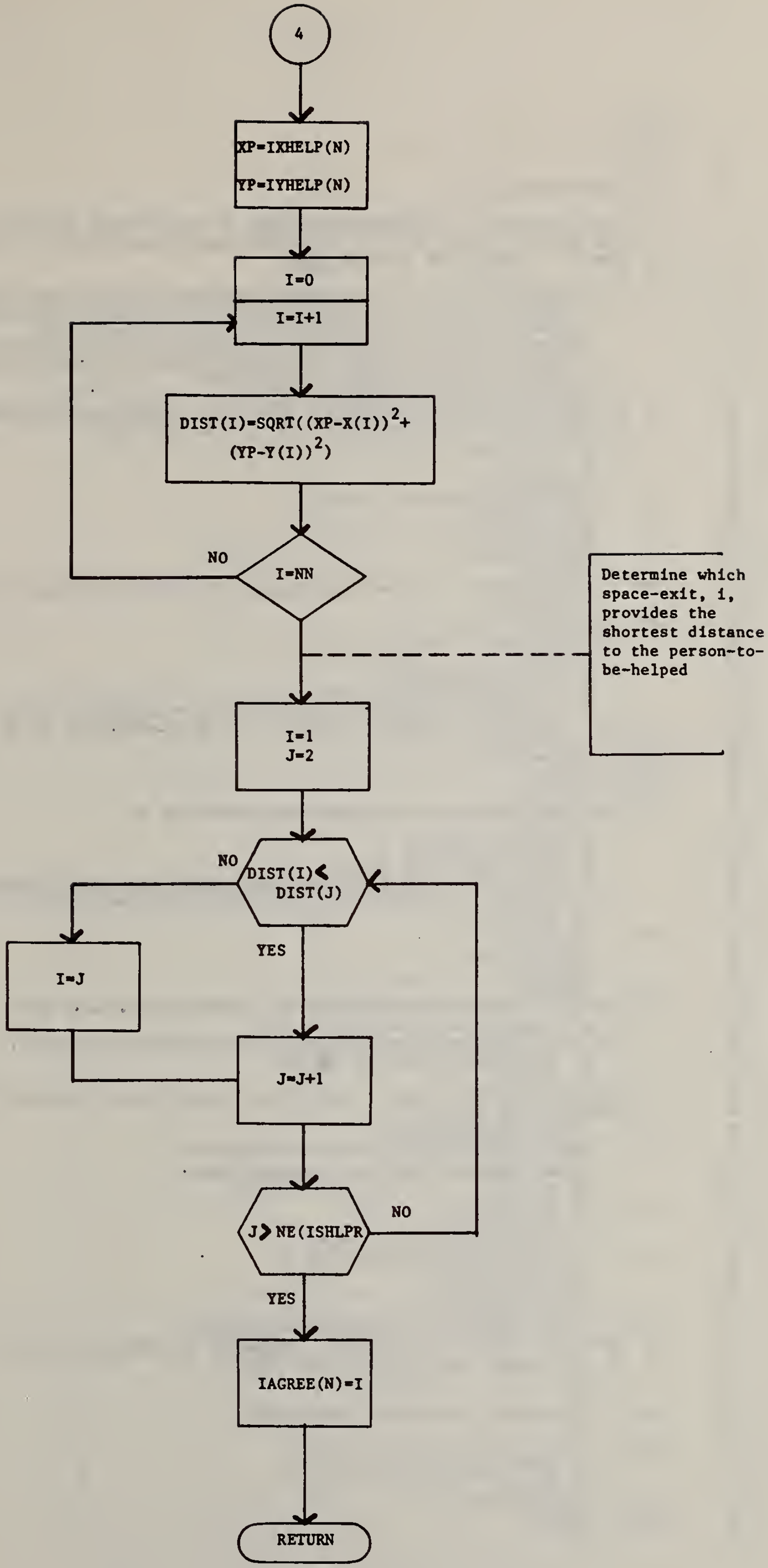


Figure 3.2 Flow Chart for Subroutine HELPR1


1
Determine which space rescue-er is in
IFLAG=0
IS=0
IS=IS+1
IFLAG=1
YES
NO
XLO(IS)< XO(N)< XHI(IS)
YLO(IS)< YO(N)< YHI(IS)
NO
YES
ISHLPR=IS
IFLAG=1
NO
IS=NSPACE
2


2
Determine which space person-to-be-helped is occupying
JFLAG=0
IS=0
IS=IS+1
JFLAG=1
YES
NO
XLO(IS)< IXHELP(N)< XHI(IS)
YLO(IS)< IYHELP(N)< YHI(IS)
NO
YES
ISPTBH=IS
JFLAG=1
NO
IS=NSPACE
YES
3


3
Are both occupying the same space?
YES
ISHLPR=ISPTBH
NO
ITEMP=IAGREE(N)
IGOALX(ISHLPR,ITEMP,N)=IXHELP(N)
IGOALY(ISHLPR,ITEMP,N)=IYHELP(N)
How many exits are available from the space occupied by the rescue-er?
YES
NE(ISHLPR)=1
RETURN
NO
multi-exit case: compute distances between each exit and the person-to-be-helped
NN=NE(ISHLPR)
I=0
I=I+1
X(I)=IGOALX(ISHLPR,I,N)
Y(I)=IGOALY(ISHLPR,I,N)
NO
I=NN
YES
4


4
XP=IXHELP(N)
YP=IYHELP(N)
I=0
I=I+1
DIST(I)=SQRT((XP-X(I))^2+ (YP-Y(I))^2)
NO
I=NN
Determine which space-exit, i, provides the shortest distance to the person-to-be-helped
I=1
J=2
NO
DIST(I)< DIST(J)
YES
I=J
J=J+1
J > NE(ISHLPR
NO
YES
IAGREE(N)=I
RETURN

```
C
C
C
C
C   SUBROUTINE HELPR1
C
C   THIS ROUTINE  (1) DETERMINES WHETHER A PRE-ASSIGNED HELPER IN FACT ENGAGES IN
C   THE HELPING TASK, AND  (2) RE-ESTABLISHES ENGAGED HELPER'S MOVEMENT GOALS IN
C   ORDER TO ASSURE THE HELPING BIAS.
C
      SUBROUTINE HELPR1 (ITIME,NTHIS,PHELP,NHELP,XO,YO,IXHELP,IYHELP,
     1  NSPACE,XLO,XHI,YLO,YHI,NE,IGOALX,IGOALY,IAGREE,IGX,IGY,IAG,IXX)
      INTEGER XO(20),YO(20),XLO(20),XHI(20),YLO(20),YHI(20)
      DIMENSION PHELP(20),NHELP(20),IXHELP(20),IYHELP(20),NE(20),
     1  IGOALX(20,10,20),IGOALY(20,10,20),IAGREE(20),IGX(20),IGY(20),
     2  IAG(20),IXX(20),X(20),Y(20),DIST(20)
      IAG(NTHIS)=IAGREE(NTHIS)
C   IF THIS IS THE 1ST TIME-FRAME, DETERMINE WHETHER THE PRE-ASSIGNED
C   HELPER IN FACT DECIDES TO GO INTO THE HELPING MODE
      IF (ITIME.EQ.1) GO TO 10
      GO TO 25
10    X=RANDNU(0)
      IF (PHELP(NTHIS)-X) 20,20,15
15    NHELP(NTHIS)=1
      GO TO 30
20    NHELP(NTHIS)=0
      RETURN
25    IF (XO(NTHIS).EQ.IXHELP(NTHIS).AND.YO(NTHIS).EQ.IYHELP(NTHIS)) GO
     1  TO 900
30    CONTINUE
C   DETERMINE WHICH SPACE HELPER IS IN
      IFLAG=0
      DO 100 IS=1,NSPACE
      IF (IFLAG.EQ.1) GO TO 100
      IF ((XLO(IS).LT.XO(NTHIS).AND.XHI(IS).GT.XO(NTHIS)).AND.
     1  (YLO(IS).LT.YO(NTHIS).AND.YHI(IS).GT.YO(NTHIS))) GO TO 101
      GO TO 100
101   ISHLPR=IS
      IFLAG=1
100   CONTINUE
C   DETERMINE WHICH SPACE PERSON-TO-BE-HELPED IS IN
      JFLAG=0
      DO 200 IS=1,NSPACE
      IF (JFLAG.EQ.1) GO TO 200
      IF ((XLO(IS).LT.IXHELP(NTHIS).AND.XHI(IS).GT.IXHELP(NTHIS)).AND.
     1  (YLO(IS).LT.IYHELP(NTHIS).AND.YHI(IS).GT.IYHELP(NTHIS)))GOTO 201
      GO TO 200
201   ISPTBH=IS
      IXX(NTHIS)=IS
      JFLAG=1
200   CONTINUE
C   DETERMINE WHETHER BOTH PERSONS ARE OCCUPYING THE SAME SPACE
      IF (ISHLPR.EQ.ISPTBH) GO TO 500
C   HOW MANY EXITS ARE AVAILABLE FROM SPACE OCCUPIED BY HELPER
      IF (NE(ISHLPR).EQ.1) GO TO 900
      GO TO 550
C   FOR THE MULTI-EXIT CASE, COMPUTE DISTANCES BETWEEN EACH EXIT AND THE
C   PERSON-TO-BE-HELPED
500   ITEMP=IAGREE(NTHIS)
      IGOALX(ISHLPR,ITEMP,NTHIS)=IXHELP(NTHIS)
      IGOALY(ISHLPR,ITEMP,NTHIS)=IYHELP(NTHIS)
      GO TO 900
550   NN=NE(ISHLPR)
      DO 701 I=1,NN
      X(I)=IGOALX(ISHLPR,I,NTHIS)
701   Y(I)=IGOALY(ISHLPR,I,NTHIS)
      XP=IXHELP(NTHIS)
      YP=IYHELP(NTHIS)
      DO 702 I=1,NN
702   DIST(I)=SQRT((XP-X(I))**2+(YP-Y(I))**2)
C   DETERMINE WHICH SPACE-EXIT, I, PROVIDES THE SHORTEST DISTANCE
C   TO THE PERSON-TO-BE-HELPED (THE PTBH):
      I=1
      J=2
601   IF (DIST(I).LT.DIST(J)) GO TO 602
      I=J
602   J=J+1
      IF (J.GT.NE(ISHLPR)) GO TO 603
      GO TO 601
603   IAGREE(NTHIS)=I
900   RETURN
      END
```

Table 3.2 Subroutine HELPR1: FORTRAN Listing

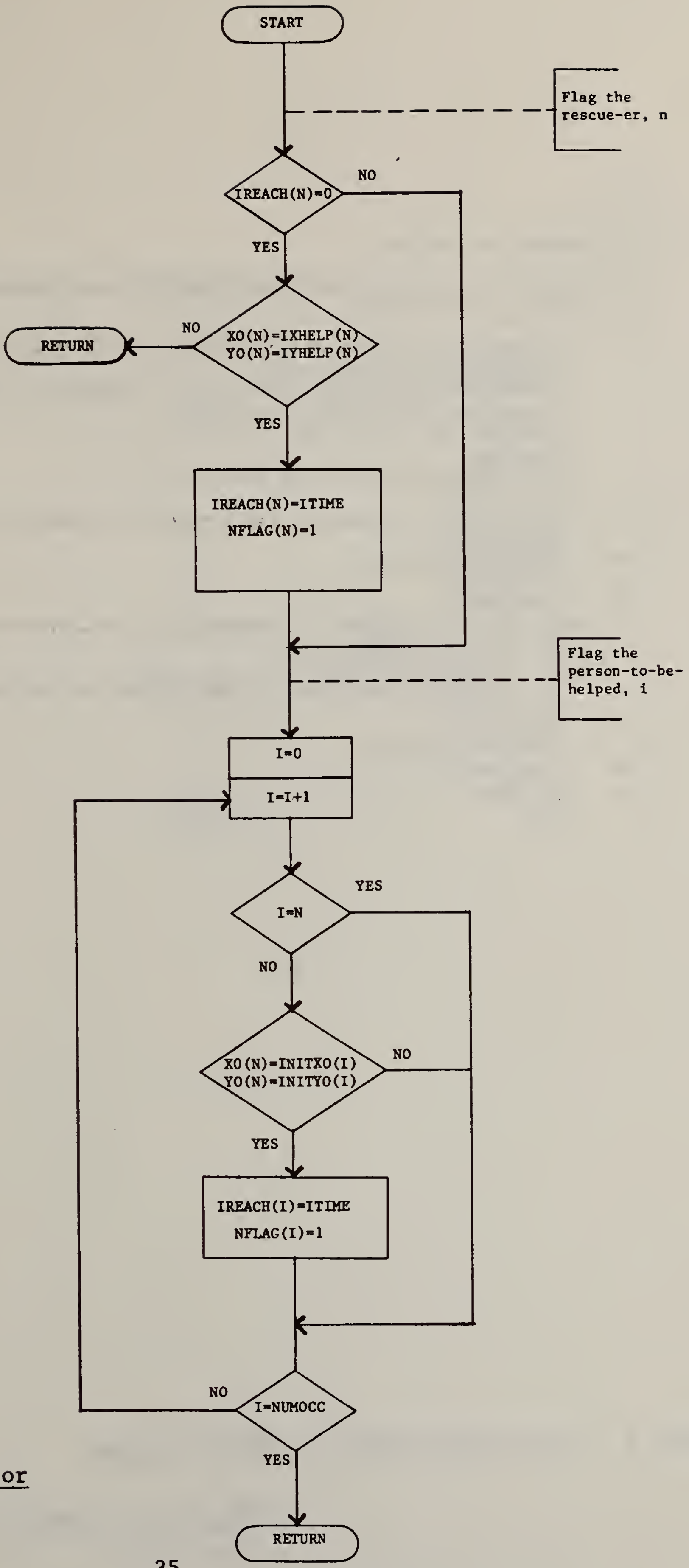


Figure 3.3 Flow Chart for Subroutine HELPR2

```
C
C
C
C
C   SUBROUTINE HELPR2
C
C   THIS ROUTINE FLAGS TEMPORAL POINT AT WHICH HELPER ACTUALLY REACHES
C   PERSON TO BE HELPED.
C
          SUBROUTINE HELPR2 (IREACH,NTHIS,XO,YO,IXHELP,IYHELP,ITIME,NFLAG,
         1  NUMOCC,INITXO,INITYO)
          DIMENSION IXHELP(20),IYHELP(20),IREACH(20),NFLAG(20)
          DIMENSION INITXO(20),INITYO(20)
          INTEGER XO(20),YO(20)
          N=NTHIS
          IF (IREACH(N).EQ.0) GO TO 10
          GO TO 30
10        IF (XO(N).EQ.IXHELP(N).AND.YO(N).EQ.IYHELP(N)) GO TO 20
          GO TO 50
20        IREACH(N)=ITIME
          NFLAG(N)=1
30        CONTINUE
C   FLAG PTBH TOO.  SCAN ALL OTHER OCC'S TO IDENTIFY CORRECT PTBH:
          DO 35 I=1,NUMOCC
          IF (I.EQ.N) GO TO 35
          IF ((XO(N).EQ.INITXO(I)).AND.(YO(N).EQ.INITYO(I))) GO TO 40
          GO TO 35
40        NFLAG(I)=1
          IREACH(I)=ITIME
35        CONTINUE
50        RETURN
          END
```

Table 3.3 Subroutine HELPR2: FORTRAN Listing

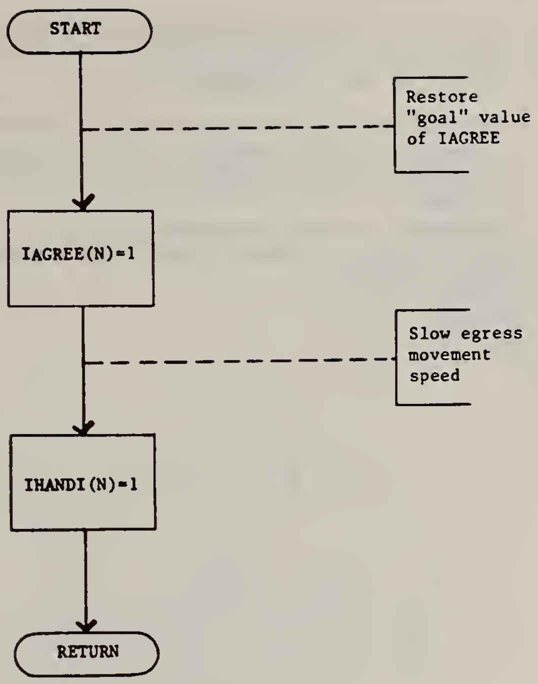
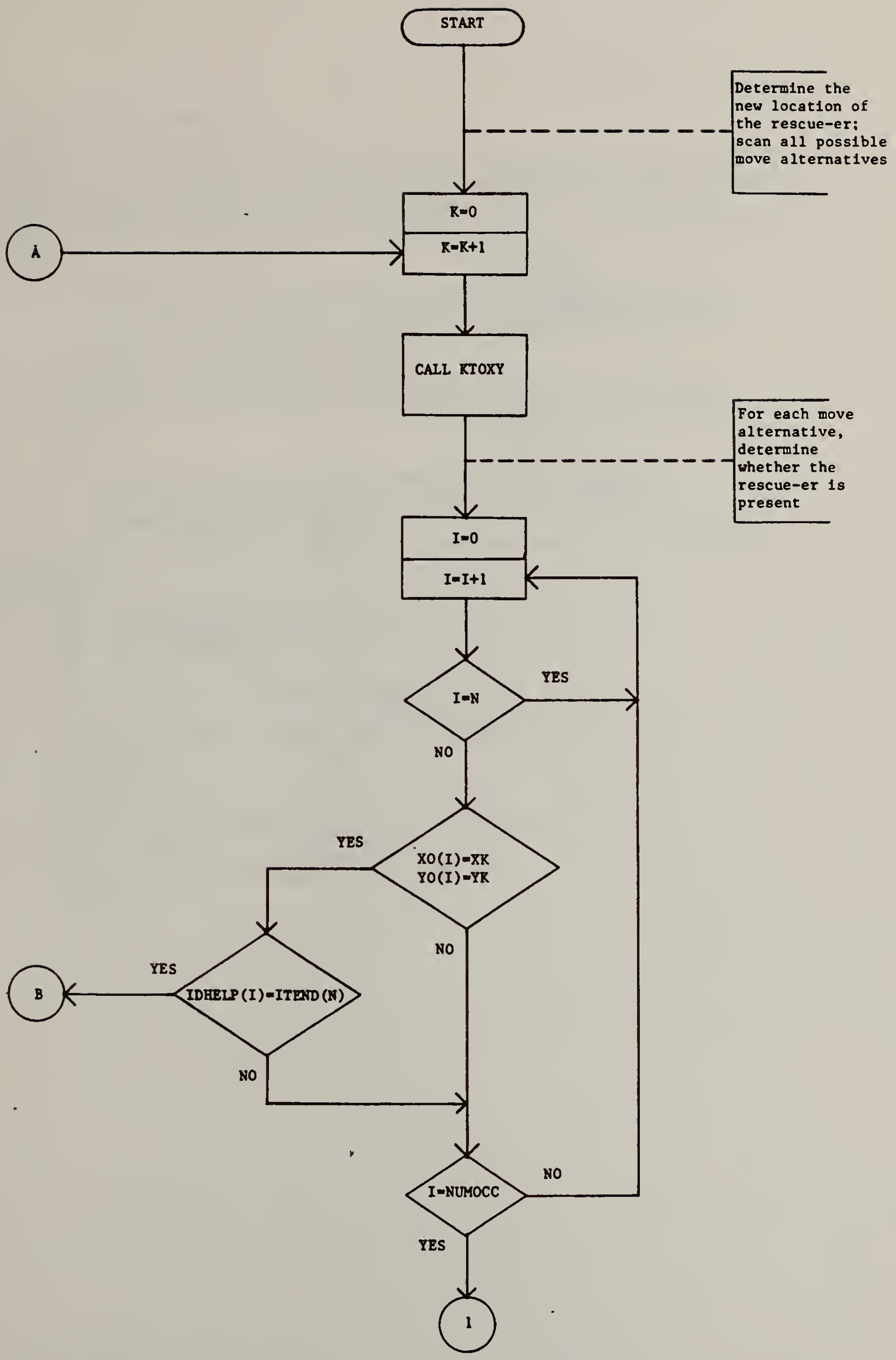


Figure 3.4 Flow Chart for Subroutine HELPR3

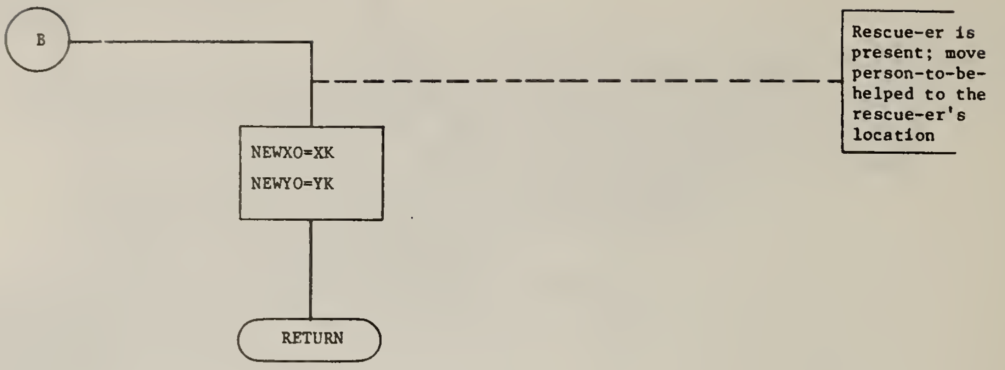
```
C
C
C
C
C  SUBROUTINE HELPR3
C
C  THIS ROUTINE RESTORES HELPER'S ORIGINAL EGRESS MOVEMENT OBJECTIVE, AFTER
C  THE HELPER HAS IN FACT REACHED THE PERSON TO BE HELPED.  IT ALSO SLOWS
C  THE HELPER'S OWN MOVEMENT.
C
      SUBROUTINE HELPR3 (ITIME,NTHIS,PHELP,NHELP,XO,YO,
     1  IXHELP,IYHELP,NSPACE,XLO,XHI,YLO,YHI,NE,
     2  IGOALX,IGOALY,IAGREE,IGX,IGY,IAG,IXX,IREACH,
     3  IHANDI)
      DIMENSION IAGREE(20),IHANDI(20)
C  RESTORE VALUE OF IAGREE, AND SLOW  MOVEMENT:
      IAGREE(NTHIS)=1
      IHANDI(NTHIS)=1
      RETURN
      END
```

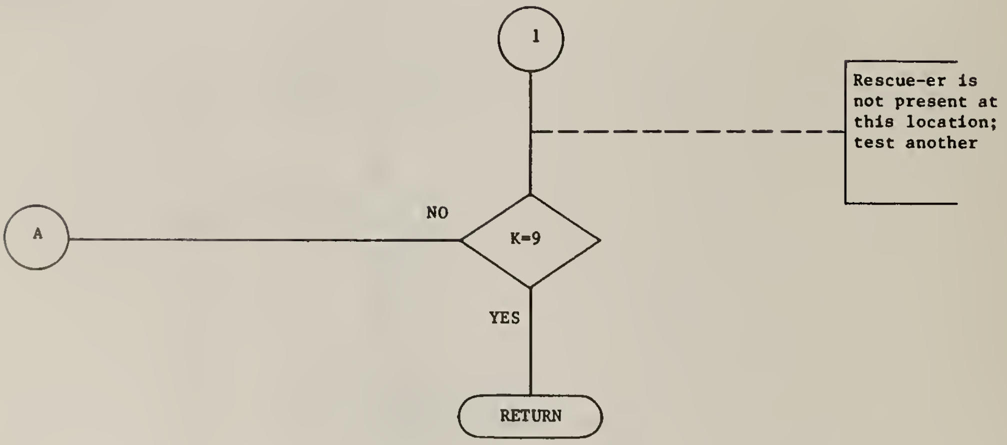
Table 3.4 Subroutine HELPR3: FORTRAN Listing



START
Determine the new location of the rescue-er; scan all possible move alternatives
K=0
K=K+1
A
CALL KTOXY
For each move alternative, determine whether the rescue-er is present
I=0
I=I+1
I=N
YES
NO
XO(I)=XK
YO(I)=YK
YES
NO
IDHELP(I)=ITEND(N)
YES
B
NO
I=NUMOCC
NO
YES
1

Figure 3.5 Flow Chart for Subroutine HELPED


B
Rescue-er is present; move person-to-be-helped to the rescue-er's location
NEWXO=XK
NEWYO=YK
RETURN


1
Rescue-er is not present at this location; test another
NO
A
K=9
YES
RETURN

```
C
C
C
C
C  SUBROUTINE HELPED
C
C  THIS PROGRAM CAUSES THE PERSON-TO-BE-HELPED (PTBH) TO REMAIN
C  WITH THE ASSIGNED HELPER FOR THE REMAINDER OF THE RUN, ONCE
C  THE HELPER FIRST REACHES THE PTBH.  AS THE HELPER MOVES, THE
C  PTBH SCANS THE FIELD AND MOVES DIRECTLY TO THE NEW LOCATION
C  OCCUPIED BY THE HELPER.
C
      SUBROUTINE HELPED (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1  XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,IRAND,
     2  P,MOVE,XK,YK,K,NUMOCC,IDHELP,ITEND,NEWXO,NEWYO)
      INTEGER XO(20),YO(20),XK,YK
      DIMENSION IDHELP(20),ITEND(20)
C  DETERMINE NEW LOCATION OF HELPER, AND MOVE PTBH TO THAT
C  LOCATION.  FIRST, SCAN ALL POSSIBLE MOVE ALTERNATIVES:
      K=0
10    K=K+1
      CALL KTOXY (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1  XO,YO,IBAR,TOTBAR;XT,YT,NAGREE,XE,YE,IAGREE,
     2  IRAND,P,MOVE,XK,YK,K)
C  FOR EACH ALTERNATIVE, DETERMINE WHETHER HELPER IS PRESENT:
      I=0
15    I=I+1
      IF (I.EQ.NTHIS) GO TO 15
      IF ((XO(I).EQ.XK).AND.(YO(I).EQ.YK)) GO TO 20
30    IF (I.EQ.NUMOCC) GO TO 25
      GO TO 15
20    IF (IDHELP(I).EQ.ITEND(NTHIS)) GO TO 35
      GO TO 30
C  HELPER IS PRESENT.  MOVE PTBH TO HELPER'S LOCATION:
35    NEWXO=XK
      NEWYO=YK
      RETURN
C  HELPER IS NOT PRESENT AT THIS ALTERNATIVE LOCATION
C  TRY ANOTHER:
25    IF (K.EQ.9) GO TO 40
      GO TO 10
40    RETURN
      END
```

Table 3.5 Subroutine HELPED: FORTRAN Listing

4.0 EXPANDED INPUT/OUTPUT AND UTILITY FEATURES

4.1 Subroutines INEXEC, INBLDG, and INOCC

To simulate a building fire using BFIRES, the user's input file must specify three categories of data. These are control data, environmental descriptors, and occupant descriptors. Control data include instructions about the number of replications desired for a given experiment, the total number of time frames each replication is to contain, the total number of occupants to be included in the simulation, and a seed number for random number generation[1]. Other control data specify output options desired, and maximum conditions permitted for spatial crowding and other functions. Finally, title information and user-supplied format statements input as control data. These data are read into the computer by Subroutine INEXEC (See Table 4.1a).

Environmental descriptors specify the floor plan under study, the location of the fire ignition point, and the fire product spread rate. Floor plan descriptions include locations of walls and doors in x-y space, and such additional information about doors as (a) whether they are initially open or closed, and (b) whether they are of the manual or automatic-closure type. The fire ignition point is also entered as an x-y spatial location. The rate of spread fire products was discussed earlier in Section 2.1. These data are input through Subroutine INBLDG (see Table 4.1b).

Occupant descriptors provide information about the initial locations of simulated occupants in x-y space, as well as data specifying various individual characteristics. Examples of such characteristics include occupants' (a) interruption limits (refer to Stahl, 1979, pp. 146-148), (b) mobility status (c) knowledge of safe exit location, (d) likelihood of opening closed doors and closing open doors just passed through, and (e) egress goals and subgoals. The various parameters which describe simulated occupants, and the way in which BFIRES uses this information is presented in the documentation of BFIRES/VERSION 1 (Stahl, 1979). Occupant descriptors are input by means of Subroutine INOCC (see Table 4.1c). Examples of control, environmental, and occupant data are provided in Table 4.1d, which illustrates a sample input file.

4.2 Subroutine PLOT, SMKSUM, and HLPSUM

The output capabilities of BFIRES I are enhanced in the new version through the introduction of three additional output options. Subroutine PLOT produces a two-dimensional graphic display of occupant and fire product locations for each successive time frame. Each individual display

[1] The Interdata 7/32 variation of BFIRES contains a random number generator which requires the user to input a five digit seed. The Univac 1108 version references an external random generator, and does not require a seed to be input.

contains x-y grid locations for all person-occupiable points. Empty locations are denoted by "0", points containing fire products are denoted by "1", and points occupied by simulated people (regardless of the number of persons simultaneously occupying the area denoted by a given point) point) are denoted by "8". In its current form, PLOT does not display physical aspects of the environment such as walls or doors. However, the user may prepare an appropriately scaled floor plan on transparent material, and overlay this plan on the computer-produced displays. By reviewing displays plotted for successive time frames, the user may observe incremental changes in occupant and fire product locations. In this way, each display is analogous to a single frame of motion picture film. The listing for subroutine PLOT is given in Table 4.2a. An example of PLOT output is shown in Figure 4.2a.

Subroutine SMKSUM prints tables describing the degree to which simulated building occupants inhabited a smoke-filled environment. For each occupant in the fire event the table lists (a) the time frames spent in the clear (smoke-free) environment, (b) the number of "low exposure" time frames, (c) the number of "high exposure" time frames, and (d) the total number of frames during which the was exposed to fire products. A frame of exposure is defined as any time frame during which an occupant is located at a smoke-filled x-y coordinate. The number of frames of exposure for each occupant is accounted for by Subroutine KTOXIC, as discussed in Section 2.3. The terms "low exposure" and "high exposure" refer to the total number of frames an occupant spends in a smoke-filled environment, as a function of the total length of the simulated event. These terms are computed in Subroutines ASSIGN and CMPUTE[1], and are detailed in Sections 4.3 and 5.2 of this report. The listing for Subroutine SMKSUM is shown in Table 4.2b, and an example of output from this option is provided in Figure 4.2b.

Finally, a tabular summary of rescue activities is provided by Subroutine HLPSUM. Tables produced by this option display pairs of preassigned rescuers and persons-to-be-helped (PTBH), and for each pair indicates (a) whether the rescuer ever actually reached the location of the PTBH, (b) the time frame during which the meeting occurred, (c) whether the pair ever attained safe egress from the threatened zone under study, and if so (d) the time frame during which escape occurred. The listing for Subroutine HLPSUM is shown in Table 4.2c. An example of tabular output from this option is given in Figure 4.2c.

4.3 Subroutines INITLZ, INITFR, and CMPUTE

Replication parameters are variables which are initalized once at the beginning of each replication of a simulation experiment. These include

[1] Occupants in the "backtrack" interruption mode (as defined in the BFIRES/ VERSION 1 documentation) are not processed through Subroutine ASSIGN. For these occupants, values of low and high exposure are computed by Subroutine CMPUTE.

various counters and flags, and other variables which are preset at some value. These are initialized through a call to Subroutine INITLZ. Time frame parameters must be reset at the onset of each successive time increment. Examples of such variables are occupant locators, which change value as simulated occupants move and alter their spatial locations. These parameters are initialized within Subroutine INITFR. See Tables 4.3a and 4.3b.

The period of time spent by occupants in an environment infiltrated by fire products is calculated within Subroutines ASSIGN and CMPUTE. Whenever an occupant is behaving under the "backtrack" interruption mode (as defined within BFIRES/VERSION 1 documentation, Stahl, 1979, pp. 149-152), processing will not be routed through ASSIGN. Hence, those occupants must be processed through CMPUTE, which duplicates ASSIGN's calculation of exposure periods. Within these routines, an occupant is considered to have experienced a "no exposure" time frame if during the frame: (a) the occupant inhabits a noninfiltrated coordinate point, or (b) he has inhabited an inflitrated area for a period shorter than one-third the total length of the simulated fire event. An occupant is said to have experienced a "low exposure" time frame if he has been in the infiltrated area for a period between one-third and one-half the total length of the event. When an occupant has spent more than half the total available frames within infiltrated areas, each frame above the half-way point is recorded as a "high exposure" time frame.

For example, consider a fire event that was run for 100 frames of simulated time. The first 33 (100/3) frames spent by occupants on the floor--whether in infiltrated or noninfiltrated areas--will be recorded as "no exposure" time frames. Any frames between 33 and 50 (100/2) spent in an infiltrated area will be recorded as "low exposure" time frames. Any frames beyond the 51st spent by occupants in infiltrated areas will be recorded as "high exposure" frames. When these data are summarized in tabular form (as produced by the SMKSUM output option described in Section 4.2), the user can quickly compare simulated occupants on the basis of relative exposure to fire products. Except for the function of Subroutine SBIAS[1] (see Section 2.4), BFIRES makes no inferences from exposure statistics compiled by CMPUTE and ASSIGN. It is for the program user to judge the effects of occupying infiltrated environments for various periods of time. Subroutine CMPUTE is listed in Table 4.3c.

[1] SBIAS biases occupants' movement decision making to favor noninfiltrated spatial locations. It is assumed that as time advances, any occupant who has not yet escaped will become more fully surrounded by fire products. Hence, such an occupant will find it continually more difficult to select effective moves. Eventually (when fully surrounded), decision making becomes a random process. In this manner, BFIRES simulates the decremental effects of inhabiting infiltrated areas for increasing periods of time.

```
C
C
C
C
C  SUBROUTINE INEXEC
C
C  THIS ROUTINE READS-IN EXECUTIVE AND CONTROL PARAMETERS
C
      SUBROUTINE INEXEC (IR,IP,NUMREP,TOTIME,NUMOCC,IRAND,
     1  EVLOPT,RREPRT,RREPT2,PIO,PI2,IALLOW,TITLE,
     2  PRINT1,PRINT2,PRINT3,PRINT4)
      DIMENSION TITLE(20)
      INTEGER TOTIME,EVLOPT,PRINT1,PRINT2,PRINT3,PRINT4,RREPRT,RREPT2
      PRINT1=0
      PRINT2=0
      PRINT3=0
      PRINT4=0
C  PRE-SET I/O PARAMETERS
      IR=5
      IP=6
C  READ-IN TITLE AND USER-SUPPLIED FORMAT STATEMENT
      READ (IR,103) TITLE
C  READ-IN NUMBER OF REPLICATIONS, LENGTH OF REPLICATIONS, INITIAL
C  NUMBER OF OCCUPANTS, AND RANDOM NUMBER SEED (SEED MUST BE READ,
C  ALTHOUGH IT IS ONLY USED BY APPLICATIONS ON 32-BIT PROCESSORS)
      READ (IR,100) NUMREP,TOTIME,NUMOCC,IRAND
C  READ-IN OPTIONS
      READ (IR,101) EVLOPT,RREPRT,RREPT2
C  READ-IN INTERRUPTION AND CROWDING PARAMETERS
      READ (IR,102) PIO,PI2,IALLOW
C
C  DETERMINE VALUES OF PRINT-OPTION PARAMETERS:
      I=RREPRT
      J=RREPT2
      IF ((I.EQ.0).AND.(J.EQ.3)) GO TO 200
      IF (((I.EQ.1).AND.(J.NE.1)).OR.((I.EQ.2).AND.((J.EQ.1)
     1  .OR.(J.EQ.2)))) PRINT1=1
      IF (((I.EQ.2).AND.((J.NE.1)).OR.((I.EQ.3).AND.(J.NE.0))
     1  .OR.((I.EQ.1).AND.(J.EQ.3)))) PRINT2=1
      IF (((I.EQ.3).AND.(J.NE.2)).OR.((I.EQ.1).AND.(J.GE.2)))
     1  PRINT3=1
      IF (((I.EQ.1).AND.(J.EQ.1)).OR.((I.GE.2).AND.(J.GE.2)))
     1  PRINT4=1
      GO TO 210
200   PRINT3=1
      PRINT4=1
210   CONTINUE
C
C  INEXEC DATA FORMAT
100   FORMAT (3(I3,1X),I5)
101   FORMAT (3(I2))
102   FORMAT (2(F4.2,1X),I2)
103   FORMAT (20A4)
C
      RETURN
      END
```

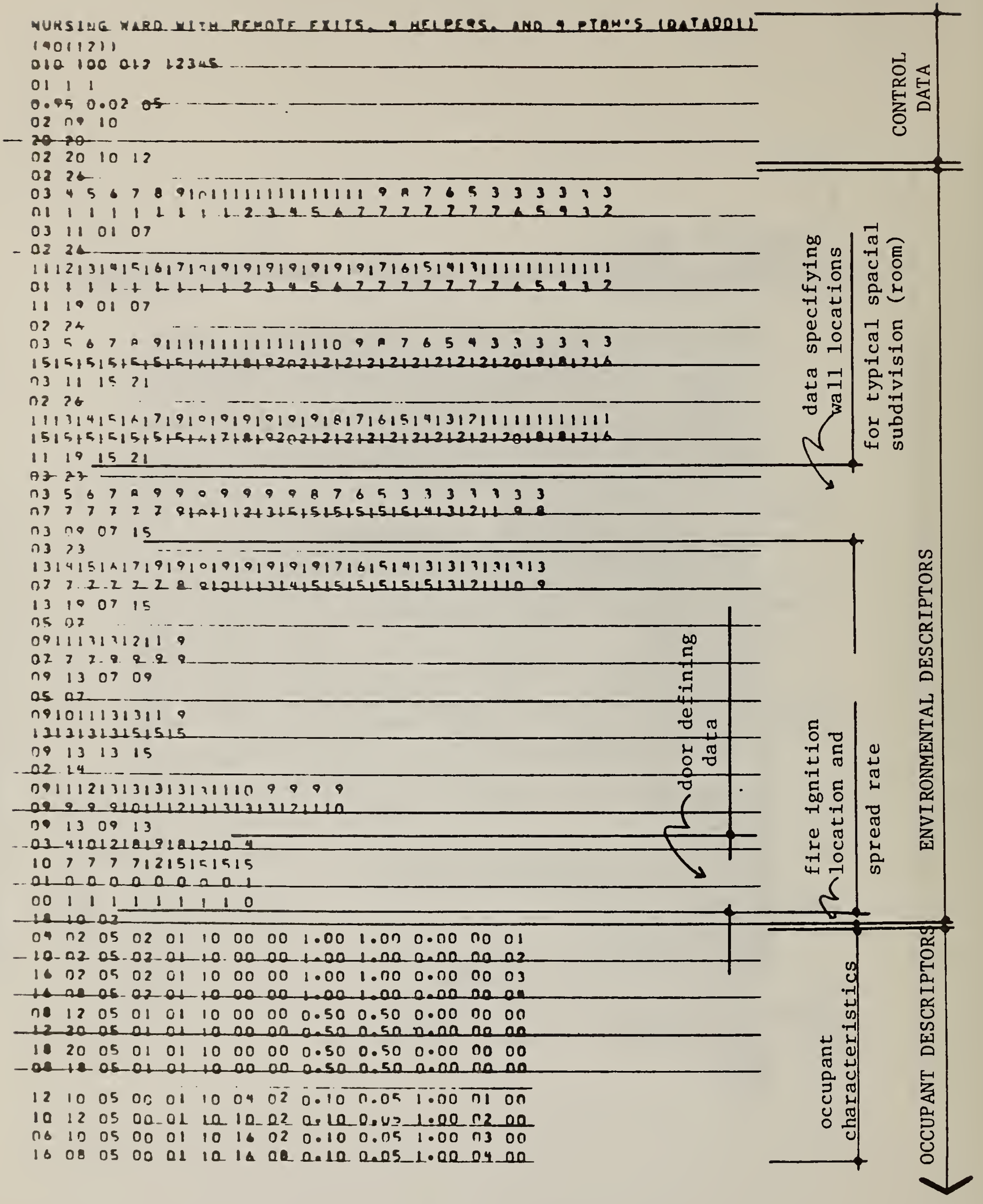
Table 4.1a Subroutine INEXEC: FORTRAN Listing

```
C
C
C
C
C
C   SUBROUTINE INBLDG
C
C   THIS ROUTINE READS-IN BUILDING FLOOR PLAN DESCRIPTORS, LOCATING
C   WALLS AND DOORS. IT ALSO READS-IN INITIAL FIRE LOCATION, AND THE SPREAD-RATE
C   PARAMETER
C
      SUBROUTINE INBLDG (NUMEXT,NSPACE,ND,XE,YE,NE,NPOINT,IBAR,
     1  TOTBAR,XLO,XHI,YLO,YHI,IDOOR,IR,XT,YT,ISRATE,
     2  MAXX,MAXY)
      DIMENSION NE(20),NPOINT(20),IBAR(20,75,2),
     1  IDOOR(30,4)
      INTEGER TOTBAR,XT,YT,XE(10),YE(10),XLO(20),XHI(20),
     1  YLO(20),YHI(20)
C   READ-IN NUMBER OF EXITS FROM THE FLOOR PLAN, NUMBER OF SPACES
C   ON THE FLOOR, AND NUMBER OF DOORS ON THE PLAN
      READ (IR,100) NUMEXT,NSPACE,ND
C   READ-IN X,Y FLOORPLAN DELIMITERS
      READ (IR,101) MAXX,MAXY
C   READ-IN LOCATIONS OF EXITS, WALL POINTS, AND INTERIOR DOORS
      READ (IR,101) (XE(I),I=1,NUMEXT),(YE(I),I=1,NUMEXT)
      DO 200 IS=1,NSPACE
      READ (IR,101) NE(IS),NPOINT(IS)
      TOTBAR=NPOINT(IS)
      READ (IR,102) (IBAR(IS,I,1),I=1,TOTBAR)
      READ (IR,102) (IBAR(IS,I,2),I=1,TOTBAR)
200   READ (IR,101) XLO(IS),XHI(IS),YLO(IS),YHI(IS)
      READ (IR,102) (IDOOR(I,1),I=1,ND)
      READ (IR,102) (IDOOR(I,2),I=1,ND)
      READ (IR,102) (IDOOR(I,3),I=1,ND)
      READ (IR,102) (IDOOR(I,4),I=1,ND)
C
C   READ-IN INITIAL FIRE LOCATION AND SPREAD-RATE PARAMETER
      READ (IR,100) XT,YT,ISRATE
C
C   INBDLG DATA FORMAT
C
100   FORMAT (3(I2,1X))
101   FORMAT (4(I2,1X))
102   FORMAT ()
C
      RETURN
      END
```

Table 4.1b: Subroutine INBLDG: FORTRAN Listing

```
C
C
C
C
C  SUBROUTINE INOCC
C
C  THIS ROUTINE READS-IN OCCUPANT PARAMETERS, AND INITIALIZES OCCUPANT
C  LOCATOR VARIABLES.
C
      SUBROUTINE INOCC (IR,NUMOCC,XO,YO,INTLIM,IHANDI,KNOWAY,
     1  POPEN,PCLOSE,TSMOKE,PHELP,IXHELP,IYHELP,KXO,KYO,NE,
     2  INITXO,INITYO,NSPACE,NUMEXT,IGOALX,IGOALY,IDHELP,ITEND)
      DIMENSION INTLIM(20),IHANDI(20),KNOWAY(20),POPEN(20),
     1  PCLOSE(20),PHELP(20),IXHELP(20),IYHELP(20)
      DIMENSION IBYSTD(20),KXO(20),KYO(20),INITXO(20),INITYO(20)
      DIMENSION IGOALX(20,10,20),IGOALY(20,10,20)
      DIMENSION ITEND(20),IDHELP(20),NE(20),IGX(20,10),IGY(20,10)
      INTEGER XO(20),YO(20),TSMOKE(20)
      DO 100 N=1,NUMOCC
C  READ-IN OCCUPANT PARAMETERS
      READ (IR,200) XO(N),YO(N),INTLIM(N),IHANDI(N),
     1  KNOWAY(N),TSMOKE(N),IXHELP(N),IYHELP(N),
     2  POPEN(N),PCLOSE(N),PHELP(N),IDHELP(N),ITEND(N)
      IBYSTD(N)=0
C  INITIALIZE OCCUPANT LOCATOR VARIABLES
      KXO(N)=XO(N)
      KYO(N)=YO(N)
      INITXO(N)=XO(N)
      INITYO(N)=YO(N)
100   CONTINUE
C  READ-IN OCCUPANTS′ SPATIAL GOAL POINTS:
      DO 300 IS=1,NSPACE
      IF (NE(IS).NE.0) GO TO 110
      J=1
      IGX(IS,J)=0
      IGY(IS,J)=0
      GO TO 115
110   J=NE(IS)
      READ (IR,201)(IGX(IS,IEXIT),IEXIT=1,J)
      READ (IR,201)(IGY(IS,IEXIT),IEXIT=1,J)
115   CONTINUE
      DO 300 N=1,NUMOCC
      DO 300 IEXIT=1,J
      IGOALX(IS,IEXIT,N)=IGX(IS,IEXIT)
300   IGOALY(IS,IEXIT,N)=IGY(IS,IEXIT)
C
C  INOCC INPUT FORMAT
200   FORMAT (8(I2,1X),3(F4.2,1X),2(I2,1X))
201   FORMAT ()
      RETURN
      END
```

Table 4.1c Subroutine INOCC: FORTRAN Listing

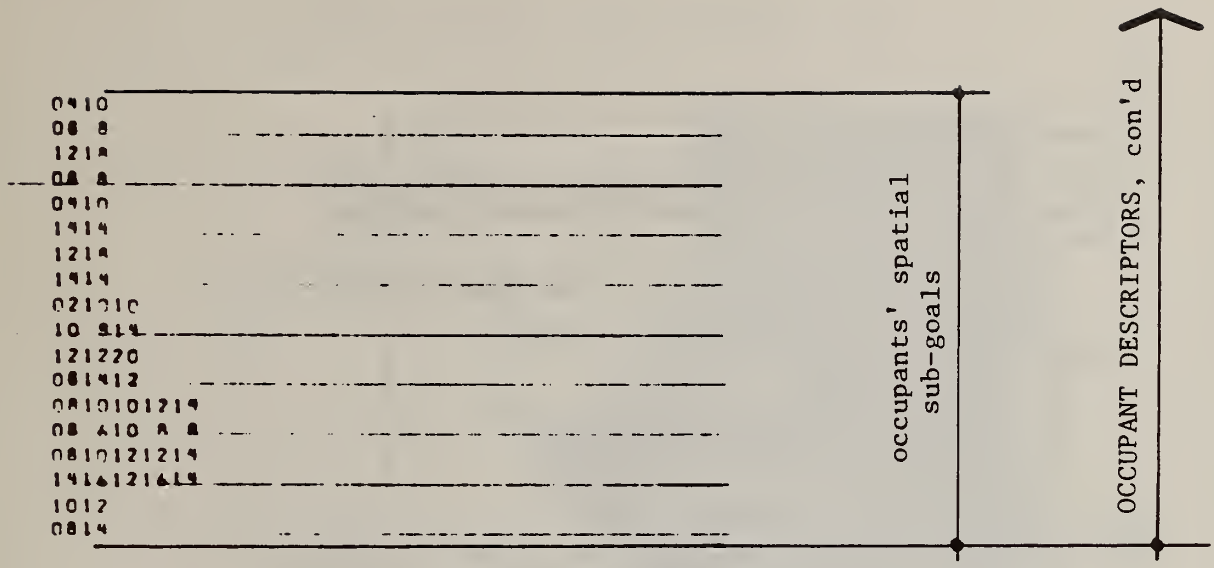


```
NURSING WARD WITH REMOTE EXITS, 4 HELPERS, AND 4 PTAM'S (DATA001)
(40(I2))
010 100 012 12345
01 1 1
0.95 0.02 05
02 09 10
20 20
02 20 10 12
02 26
03 4 5 6 7 8 91011111111111111111 9 8 7 6 5 3 3 3 3 3 3
01 1 1 1 1 1 1 1 1 2 3 4 5 6 7 7 7 7 7 7 7 6 5 4 3 2
03 11 01 07
02 26
111213141516171819191919191919171615141311111111111111
01 1 1 1 1 1 1 1 1 2 3 4 5 6 7 7 7 7 7 7 7 6 5 4 3 2
11 19 01 07
02 24
03 5 6 7 8 911111111111111110 9 8 7 6 5 4 3 3 3 3 3 3
151515151515151617181920212121212121212120191817 16
03 11 15 21
02 26
1113141516171910191919191918171615141317111111111111
1515151515151515161718192021212121212121212120191817 16
11 19 15 21
03 23
03 5 6 7 8 9 9 0 9 9 9 9 8 7 6 5 3 3 3 3 3 3 3
07 7 7 7 7 7 910111213151515151516141312 11 9 8
03 09 07 15
03 23
1314151617191910191919191917161514131313131313
07 7 7 7 7 7 8 91011131415151515151513121110 9
13 19 07 15
05 07
0911131312 11 9
07 7 7 9 9 9 9
09 13 07 09
05 07
0910111313 11 9
1313131313151515
09 13 13 15
02 14
09111213131313131311110 9 9 9 9
09 9 9 9101111213131313121110
09 13 09 13
03 4101218191812 10 4
10 7 7 7 71215151515
01 0 0 0 0 0 0 0 0 1
00 1 1 1 1 1 1 1 1 0
18 10 02
04 02 05 02 01 10 00 00 1.00 1.00 0.00 00 01
10 02 05 02 01 10 00 00 1.00 1.00 0.00 00 02
16 02 05 02 01 10 00 00 1.00 1.00 0.00 00 03
16 08 05 02 01 10 00 00 1.00 1.00 0.00 00 04
08 12 05 01 01 10 00 00 0.50 0.50 0.00 00 00
12 20 05 01 01 10 00 00 0.50 0.50 0.00 00 00
18 20 05 01 01 10 00 00 0.50 0.50 0.00 00 00
08 18 05 01 01 10 00 00 0.50 0.50 0.00 00 00
12 10 05 00 01 10 04 02 0.10 0.05 1.00 01 00
10 12 05 00 01 10 10 02 0.10 0.05 1.00 02 00
06 10 05 00 01 10 16 02 0.10 0.05 1.00 03 00
16 08 05 00 01 10 16 08 0.10 0.05 1.00 04 00
```

CONTROL DATA
data specifying wall locations
for typical spacial subdivision (room)
door defining data
fire ignition location and spread rate
ENVIRONMENTAL DESCRIPTORS
occupant characteristics
OCCUPANT DESCRIPTORS

con'd

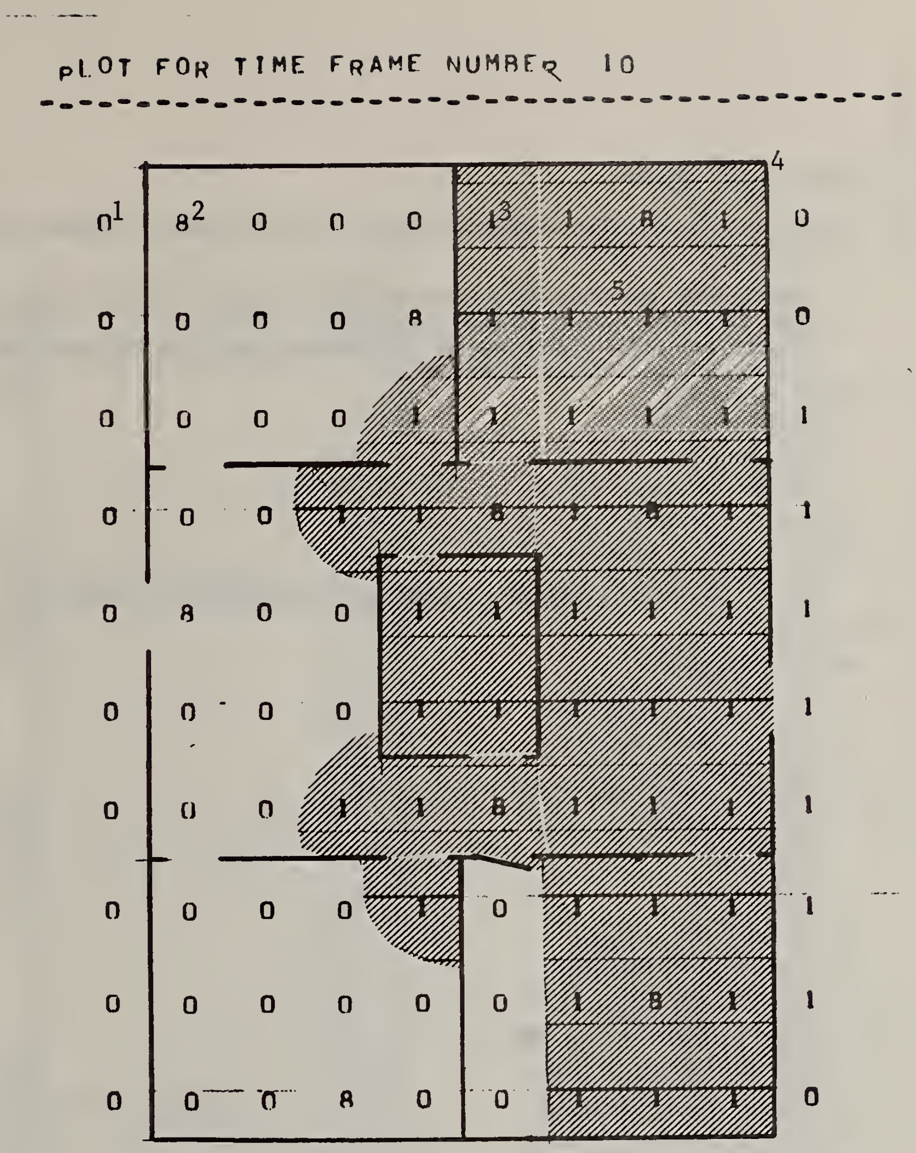
Table 4.1d Typical Input Data File, Indicating General File Organization and Structure



0910
08 8
1218
08 8
0910
1914
1218
1914
021010
10 814
121220
081912
0810101219
08 610 8 8
0810121219
1916121619
1012
0814
occupants' spatial sub-goals
OCCUPANT DESCRIPTORS, con'd

```
C
C
C
C
C  SUBROUTINE PLOT
C
C  THIS PROGRAM PLOTS SMOKE AND OCCUPANT LOCATIONS FOR A
C  SINGLE TIME FRAME.  ONLY PERSON-OCCUPIABLE LOCATIONS
C  ARE SHOWN.
C
      SUBROUTINE PLOT (XO,YO,NUMOCC,ITIME,MAXX,MAXY,ISMOKE)
      INTEGER XO(20),YO(20)
      DIMENSION ISMOKE(100,100),IPLOT(100,100)
C  BUILD THE PLOT MATRIX
      DO 100 J=2,MAXY,2
      DO 100 I=2,MAXX,2
      IF (ISMOKE(I,J).GT.0) GO TO 110
      IPLOT(I,J)=0
      GO TO 120
110   IPLOT(I,J)=1
120   CONTINUE
C  DETERMINE LOCATIONS OF OCCUPANTS WITHIN THE PLOT MATRIX
      DO 100 N=1,NUMOCC
      IF (XO(N).EQ.I.AND.YO(N).EQ.J) GO TO 130
      GO TO 100
130   IPLOT(I,J)=8
100   CONTINUE
C  PRINT THE PLOT MATRIX
      WRITE (6,500) ITIME
      DO 300 J=2,MAXY,2
      WRITE (6,501) (IPLOT(I,J),I=2,MAXX,2)
      WRITE (6,502)
300   CONTINUE
500   FORMAT ('1',1X,'PLOT FOR TIME FRAME NUMBER',I4,/,130('-'),///)
501   FORMAT (1X,60(2X,I2))
502   FORMAT (1X,/)
      RETURN
      END
```

Table 4.2a Subroutine PLOT: FORTRAN Listing



```
PLOT FOR TIME FRAME NUMBER  10
----------------------------------------------------

0   8   0   0   0   1   1   8   1   0

0   0   0   0   8   1   1   1   1   0

0   0   0   0   1   1   1   1   1   1

0   0   0   1   1   8   1   8   1   1

0   8   0   0   1   1   1   1   1   1

0   0   0   0   1   1   1   1   1   1

0   0   0   1   1   8   1   1   1   1

0   0   0   0   1   0   1   1   1   1

0   0   0   0   0   0   1   8   1   1

0   0   0   8   0   0   1   1   1   0
```

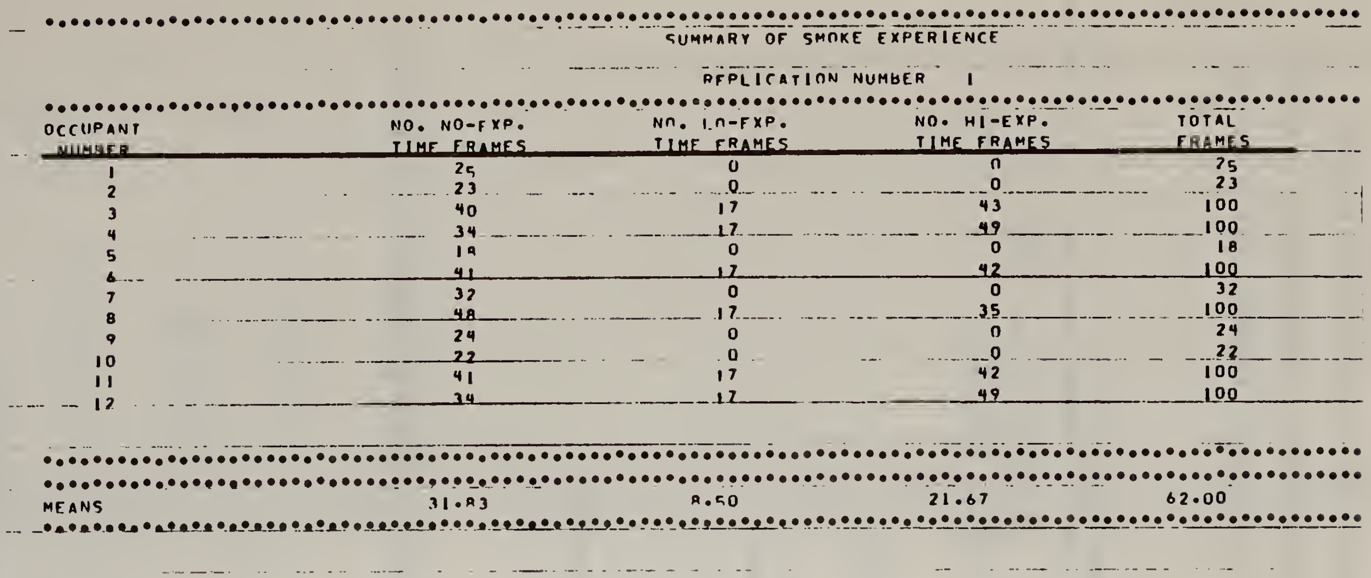
Notes:

(1) Denotes empty spatial location.
(2) Denotes point occupied by one or more simulated persons.
(3) Denotes point infiltrated by fire products.
(4) Floor plan elements (e.g., walls) drawn-in by user.
(5) Shading drawn-in by user, to denote extent of fire product infiltration.

Figure 4.2a Sample Output from the PLOT Option

```
C
C
C
C
C  SUBROUTINE SMKSUM
C
C  THIS ROUTINE PRINTS A TABLE SUMMARIZING OCCUPANTS' SMOKE EXPERIENCES FOR
C  A SINGLE REPLICATION.
C
      SUBROUTINE SMKSUM (III,NUMOCC,NOEXP,LOWEXP,IHIEXP,ITOTAL)
      REAL MEANNO,MEANLO,MEANHI,MEANTO
      DIMENSION NOEXP(20),LOWEXP(20),IHIEXP(20),ITOTAL(20),XNO(20),
     1  XLO(20),XHI(20),XTOTAL(20)
C  PRINT HEADINGS
      WRITE (6,100)
      WRITE (6,101)
      WRITE (6,102) III
      WRITE (6,101)
      WRITE (6,103)
C  PRINT SUMMARY TABLE
      DO 200 I=1,NUMOCC
      WRITE (6,105) I,NOEXP(I),LOWEXP(I),IHIEXP(I),ITOTAL(I)
200   CONTINUE
      WRITE (6,106)
C  COMPUTE MEANS
      DO 250 I=1,NUMOCC
      XNO(I)=NOEXP(I)
      XLO(I)=LOWEXP(I)
      XHI(I)=IHIEXP(I)
250   XTOTAL(I)=ITOTAL(I)
      TOTOCC=NUMOCC
      TOTNO=0.
      TOTLO=0.
      TOTHI=0.
      TOTOT=0.
      DO 300 I=1,NUMOCC
      TOTNO=TOTNO+XNO(I)
      TOTLO=TOTLO+XLO(I)
      TOTHI=TOTHI+XHI(I)
300   TOTOT=TOTOT+XTOTAL(I)
      MEANNO=TOTNO/TOTOCC
      MEANLO=TOTLO/TOTOCC
      MEANHI=TOTHI/TOTOCC
      MEANTO=TOTOT/TOTOCC
C  WRITE MEANS
      WRITE (6,101)
      WRITE (6,107) MEANNO,MEANLO,MEANHI,MEANTO
      WRITE (6,101)
C  FORMAT STATEMENTS
100   FORMAT ('1')
101   FORMAT (130('*'))
102   FORMAT (51X,'SUMMARY OF SMOKE EXPERIENCE',//,54X,'REPLICATION NUMB
     1ER',I4)
103   FORMAT (1X,'OCCUPANT',20X,'NO. NO-EXP.',10X,'NO. LO-EXP.',10X,
     1  'NO. HI-EXP.',10X,'TOTAL',/,2X,'NUMBER',21X,'TIME FRAMES',10X,
     2  'TIME FRAMES',10X,'TIME FRAMES',10X,'FRAMES')
104   FORMAT (130('*'),//)
105   FORMAT (5X,I2,26X,I3,18X,I3,18X,I3,16X,I3)
106   FORMAT (1X,//,130('*'))
107   FORMAT (1X,'MEANS',25X,F6.2,14X,F6.2,14X,F6.2,13X,F6.2)
      RETURN
      END
```

Table 4.2b Subroutine SMKSUM: FORTRAN Listing

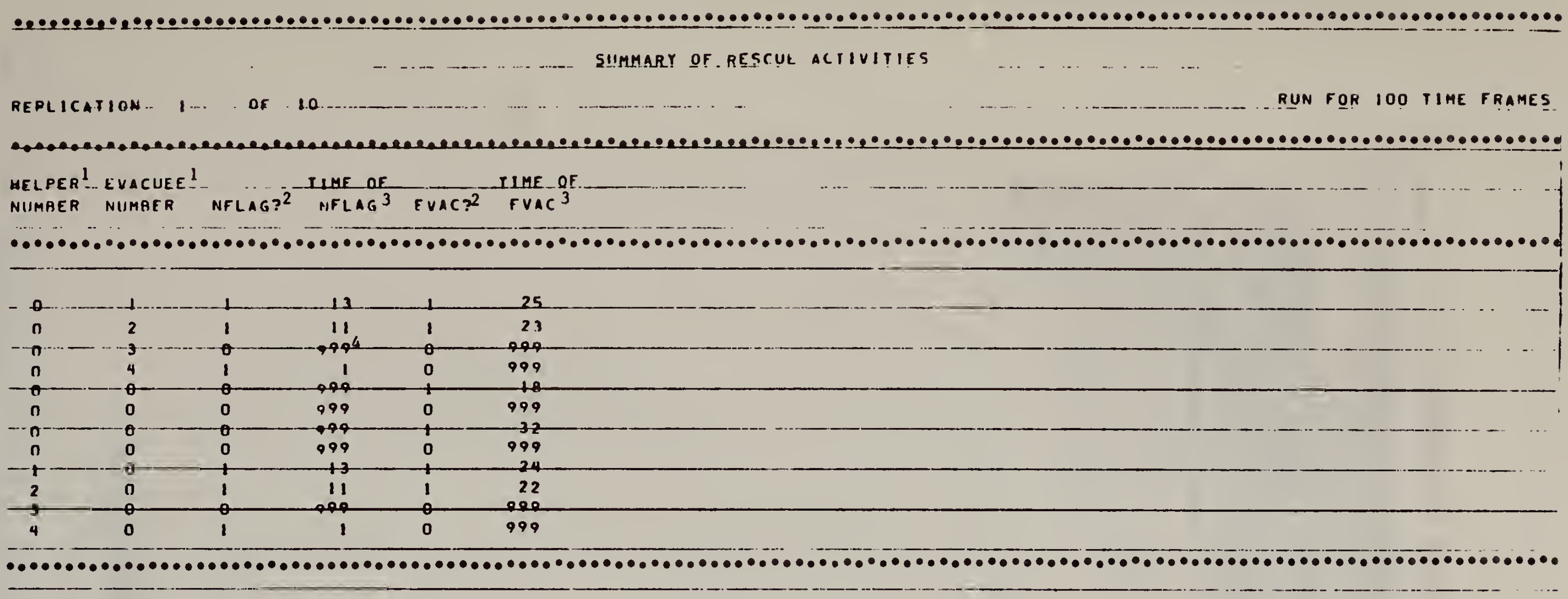


```
••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
                                                    SUMMARY OF SMOKE EXPERIENCE

                                                    REPLICATION NUMBER    1
••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
OCCUPANT                      NO. NO-EXP.          NO. LO-EXP.          NO. HI-EXP.          TOTAL
 NUMBER                       TIME FRAMES          TIME FRAMES          TIME FRAMES          FRAMES
      1                            25                   0                    0                 25
      2                            23                   0                    0                 23
      3                            40                  17                   43                100
      4                            34                  17                   49                100
      5                            18                   0                    0                 18
      6                            41                  17                   42                100
      7                            32                   0                    0                 32
      8                            48                  17                   35                100
      9                            24                   0                    0                 24
     10                            22                   0                    0                 22
     11                            41                  17                   42                100
     12                            34                  17                   49                100

••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
MEANS                           31.83                8.50                21.67                62.00
••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
```

Figure 4.2b Sample Output from the SMKSUM Option

```
C
C
C
C
C  SUBROUTINE HLPSUM
C
C  THIS ROUTINE GENERATES A SUMMARY TABLE FOR RESCUE ACTIVITIES.
C
      SUBROUTINE HLPSUM (IFLAG,NFLAG,NUMOCC,ISCORE,IREACH,III,NUMREP,
     1  TOTIME,IDHELP,ITEND,IP)
      DIMENSION IFLAG(20),NFLAG(20),IEVAC(20),ISCORE(20),IREACH(20),
     1  IDHELP(20),ITEND(20)
      INTEGER TOTIME
98    FORMAT ('1')
99    FORMAT (1X,130('*'),/)
100   FORMAT(1X,2(130('*')),/)
101   FORMAT(50X,'SUMMARY OF RESCUE ACTIVITIES',/)
102   FORMAT(1X,'REPLICATION',I4,5X,'OF',I4,80X,'RUN FOR',I4,1X,'TIME FR
     1AMES',/)
103   FORMAT(1X,'HELPER  EVACUEE',10X,'TIME OF',9X,'TIME OF')
104   FORMAT(1X,'NUMBER  NUMBER   NFLAG?   NFLAG   EVAC?   EVAC',/)
105   FORMAT(2X,I2,6X,I2,7X,I1,7X,I3,6X,I1,6X,I3)
106   FORMAT(' ')
C
      DO 200 N=1,NUMOCC
      IF (IFLAG(N).EQ.0) GO TO 210
      IEVAC(N)=1
      GO TO 220
210   IEVAC(N)=0
      ISCORE(N)=999
220   IF (NFLAG(N).EQ.1) GO TO 200
      IREACH(N)=999
200   CONTINUE
C
      WRITE (IP,98)
      WRITE (IP,100)
      WRITE (IP,101)
      WRITE (IP,102) III,NUMREP,TOTIME
      WRITE (IP,100)
      WRITE (IP,103)
      WRITE (IP,104)
      WRITE (IP,99)
      WRITE (IP,106)
      DO 300 N=1,NUMOCC
      WRITE (IP,105) IDHELP(N),ITEND(N),NFLAG(N),IREACH(N),IEVAC(N),
     1  ISCORE(N)
300   CONTINUE
      WRITE (IP,106)
      WRITE (IP,100)
      RETURN
      END
```

Table 4.2c Subroutine HLPSUM: FORTRAN Listing



```
                               SUMMARY OF RESCUE ACTIVITIES

REPLICATION  1   OF  10                                              RUN FOR 100 TIME FRAMES

HELPER[1] EVACUEE[1]           TIME OF           TIME OF
NUMBER    NUMBER     NFLAG?[2]  NFLAG[3]  EVAC?[2]  EVAC[3]

   0         1          1         13         1        25
   0         2          1         11         1        23
   0         3          0        999[4]      0       999
   0         4          1          1         0       999
   0         0          0        999         1        18
   0         0          0        999         0       999
   0         0          0        999         1        32
   0         0          0        999         0       999
   1         0          1         13         1        24
   2         0          1         11         1        22
   3         0          0        999         0       999
   4         0          1          1         0       999
```

Notes:

(1) Helper n and evacuee n constitute a single pair.
(2) Indicates whether the helper ever actually reaches his preassigned evacuee: 0=no; 1=yes.
(3) In time frame units.
(4) Denotes "not applicable", since a helper never reached an evacuee, or a pair never escaped during the period of the simulated event.

Figure 4.2c Sample Output from the HLPSUM Option

```
C
C
C
C
C  SUBROUTINE INITLZ
C
C  THIS ROUTINE INITIALIZES REPLICATION PARAMETERS
C
      SUBROUTINE INITLZ (NUMOCC,XO,YO,KXO,KYO,LOWEXP,NOEXP,
     1  IHIEXP,ITOTAL,ITOXIC,IFLAG,NUMSTP,ISCORE,IBACK,
     2  JTIME,INTR,INTNUM,ITYPE1,ITYPE2,IDPASS,MAXX,
     3  MAXY,XT,YT,ISMOKE,IREACH,NHELP,NFLAG,KNOWAY,IAGREE)
      DIMENSION KXO(20),KYO(20),NOEXP(20),LOWEXP(20),IHIEXP(20),
     1  ITOTAL(20),ITOXIC(20),IFLAG(20),NUMSTP(20),ISCORE
     2  (20),IBACK(20),INTR(20),INTNUM(20),ITYPE1(20),
     3  ITYPE2(20),IDPASS(20),ISMOKE(100,100),JTIME(10)
      DIMENSION IREACH(20),NHELP(20),NFLAG(20),KNOWAY(20)
      DIMENSION IAGREE(20)
      INTEGER XO(20),YO(20),XT,YT
C  INITIALIZE PARAMETERS
      DO 100 N=1,NUMOCC
      IAGREE(N)=KNOWAY(N)
      IREACH(N)=0
      NHELP(N)=0
      NFLAG(N)=0
      XO(N)=KXO(N)
      YO(N)=KYO(N)
      NOEXP(N)=0
      LOWEXP(N)=0
      IHIEXP(N)=0
      ITOTAL(N)=0
      ITOXIC(N)=0
      IFLAG(N)=0
      NUMSTP(N)=0
      ISCORE(N)=0
      IBACK(N)=0
      JTIME(N)=0
      INTR(N)=0
      INTNUM(N)=0
      ITYPE1(N)=0
      ITYPE2(N)=0
100   IDPASS(N)=0
C  INTIALIZE THE SMOKE MATRIC?X
      DO 200 J=2,MAXY,2
      DO 200 I=2,MAXX,2
200   ISMOKE(I,J)=0
      ISMOKE(XT,YT)=1
      RETURN
      END
```

Table 4.3a Subroutine INITLZ: FORTRAN Listing

```
C
C
C
C
C   SUBROUTINE INITFR
C
C   THIS PROGRAM INITIALIZES OCCUPANT LOCATORS AT THE START OF EACH
C   TIME-FRAME.
C
      SUBROUTINE INITFR (NTHIS,XO,YO,ITIME,XPRIOR,YPRIOR,
     1  IXTRCE,IYTRCE,XOB,YOB)
      DIMENSION IXTRCE(20,100),IYTRCE(20,100)
      INTEGER XO(20),YO(20),XPRIOR(20),YPRIOR(20),XOB(20,100),
     1  YOB(20,100)
      XPRIOR(NTHIS)=XO(NTHIS)
      YPRIOR(NTHIS)=YO(NTHIS)
      IXTRCE(NTHIS,ITIME)=XO(NTHIS)
      IYTRCE(NTHIS,ITIME)=YO(NTHIS)
      XOB(NTHIS,ITIME)=XO(NTHIS)
      YOB(NTHIS,ITIME)=YO(NTHIS)
      RETURN
      END
```

Table 4.3b Subroutine INITFR: FORTRAN Listing

```
C
C
C
C
C  SUBROUTINE CMPUTE
C
C  THIS ROUTINE COMPUTES VALUES OF NOEXP,LOWEXP, AND IHIEXP (THE SMOKE
C  EXPOSURE COUNTERES) FOR THOSE OCCUPANTS CURRENTLY IN THE TYPE-2
C  INTERRUPTION MODE.  ?THESE OCCUPANTS BI-PASS SUBROUTINE ASSIGN,
C  WHERE THE SMOKE EXPOSURE COUNTERS ARE NORMALLY COMPUTED.
C  SUBROUTINE CMPUTE IS BI-PASSED BY OCCUPANTS CURRENTLY UNINTERRUPTED,
C  AND BY OCCUPANTS CURRENTLY IN THE TYPE-1 INTERRUPTION MODE.
C  CMPUTE IS ALSO CALLED FOR THOSE OCCUPANTS WHO ARE IMMOBILE
C  (I.E., FOR WHOM IHANDI(N)=2).
C
      SUBROUTINE CMPUTE (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1  XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,IRAND,P,MOVE,
     2  XK,YK,K,L,IS,IGOALX,IGOALY,IENTER,X,IDOOR,POPEN,ND,
     3  MDOOR,PCLOSE,NOEXP,LOWEXP,IHIEXP,ITOTAL,ISMOKE,
     4  TSMOKE,ITOXIC,TOTIME)
      DIMENSION NOEXP(20),LOWEXP(20),IHIEXP(20),ITOXIC(20)
      DIMENSION ITOTAL(20)
      INTEGER TOTIME
      CALL KTOXIC (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,XO,YO,
     1  IBAR,X1,YT,NAGREE,XE,YE,IAGREE,IRAND,P,MOVE,XK,YK,
     2  K,L,IS,IGOALX,IGOALY,IENTER,X,IDOOR,POPEN,ND,MDOOR,
     3  PCLOSE,NOEXP,LOWEXP,IHIEXP,ITOTAL,ISMOKE,TSMOKE,
     4  ITOXIC)
      A=ITOXIC(NTHIS)
      B=FLOAT(TOTIME)/3.
      C=FLOAT(TOTIME)/2.
      IF (A.GT.B) GO TO 10
      NOEXP(NTHIS)=NOEXP(NTHIS)+1
      GO TO 12
10    IF (A.GT.C) GO TO 11
      LOWEXP(NTHIS)=LOWEXP(NTHIS)+1
      GO TO 12
11    IHIEXP(NTHIS)=IHIEXP(NTHIS)+1
12    ITOTAL(NTHIS)=ITOTAL(NTHIS)+1
      RETURN
      END
```

Table 4.3c Subroutine CMPUTE: FORTRAN Listing

5.0 MODIFICATIONS TO BFIRES/VERSION 1 ROUTINES

5.1 EXEC2

This section addresses the BFIRES executive, which has been modified to process the smoke and rescue modules, and to facilitate the expanded input, output and utility features of BFIRES II. Principal differences between EXEC1 and EXEC2 are: (1) the incorporation of special data-reading subroutines (INEXEC, INBLDG, and INOCC) into EXEC2, and (2) the delegation of parameter initialization tasks to subroutines (INITLZ and INITFR). Formerly, data input and parameter initialization were conducted within the body of the BFIRES executive. The use of subroutines for these functions reduces the length and complexity of EXEC2, and makes it an easier program for the user to follow. EXEC2 is described in detail in Figure 5.1 and Table 5.1.

5.2 Subroutine ASSIGN

Subroutine ASSIGN required modifications to enable it to function properly within BFIRES/VERSION 2. The new version of ASSIGN, referred to as ASSIGN/VERSION 2, includes calls to subroutines KTOXIC and SBIAS (discussed in specific sections of this report), the subroutines which process the effects of occupying a toxic envionment upon egress movement decision making. ASSIGN/VERSION 2 is detailed in Figure 5.2 and Table 5.2.

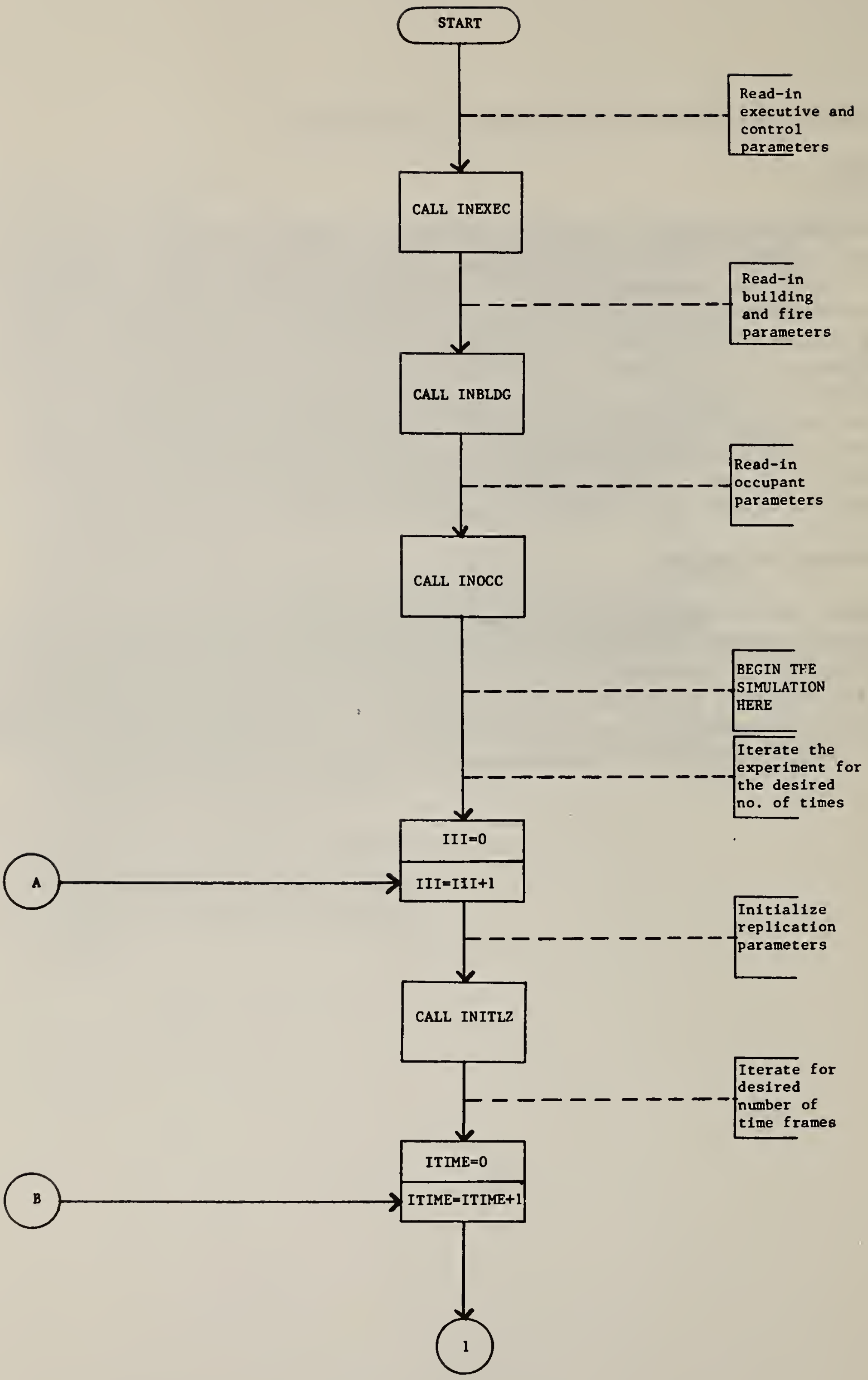
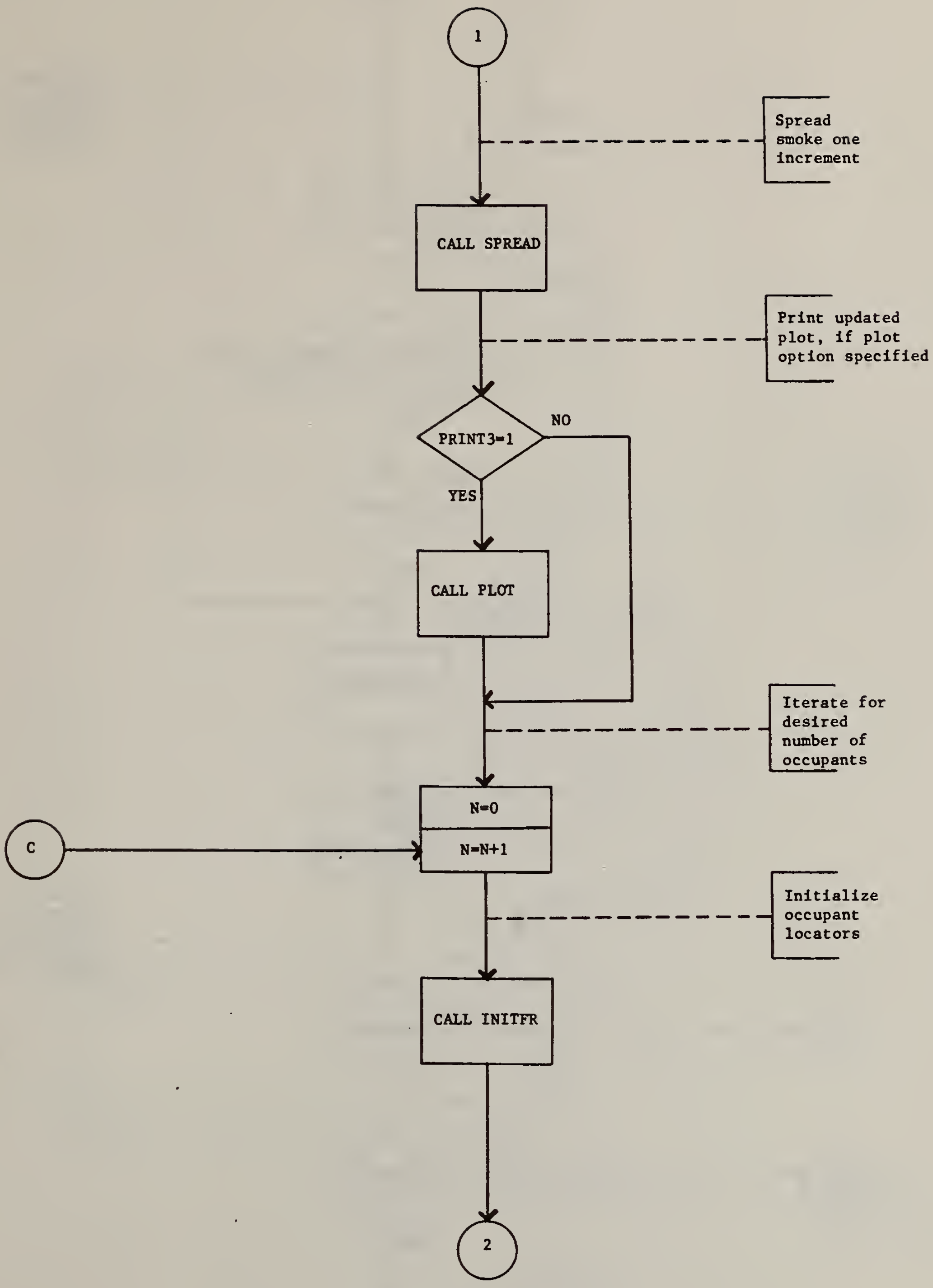
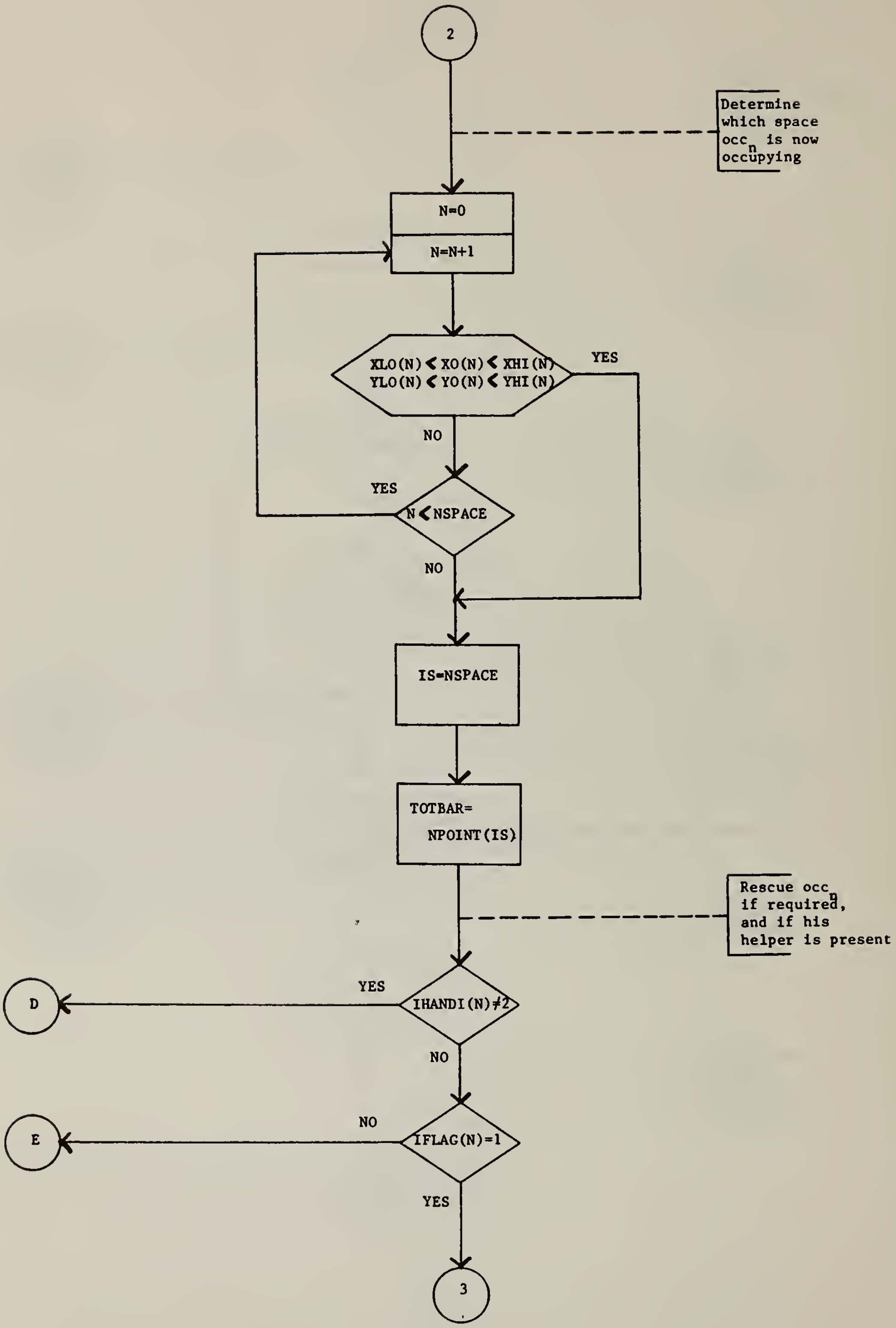
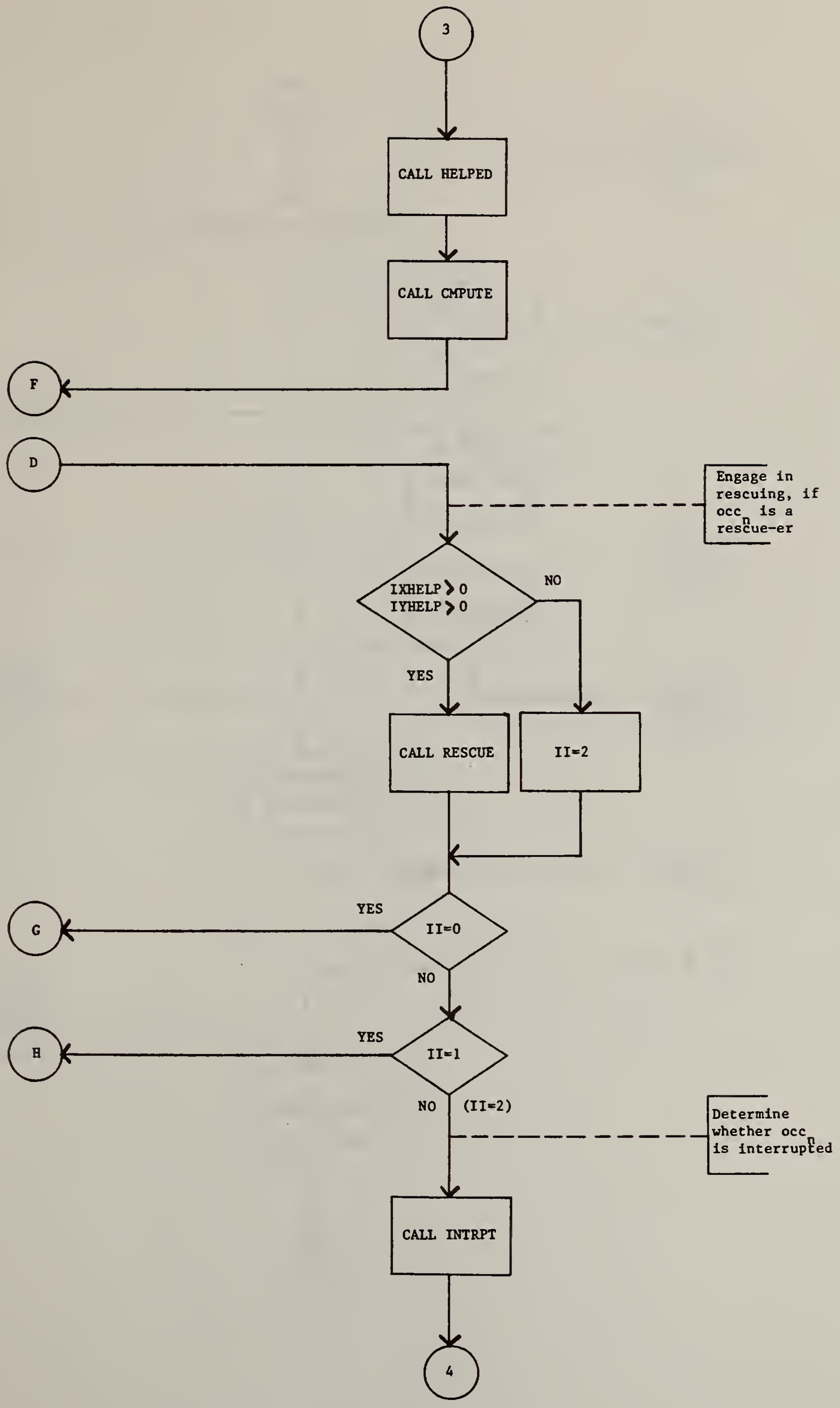
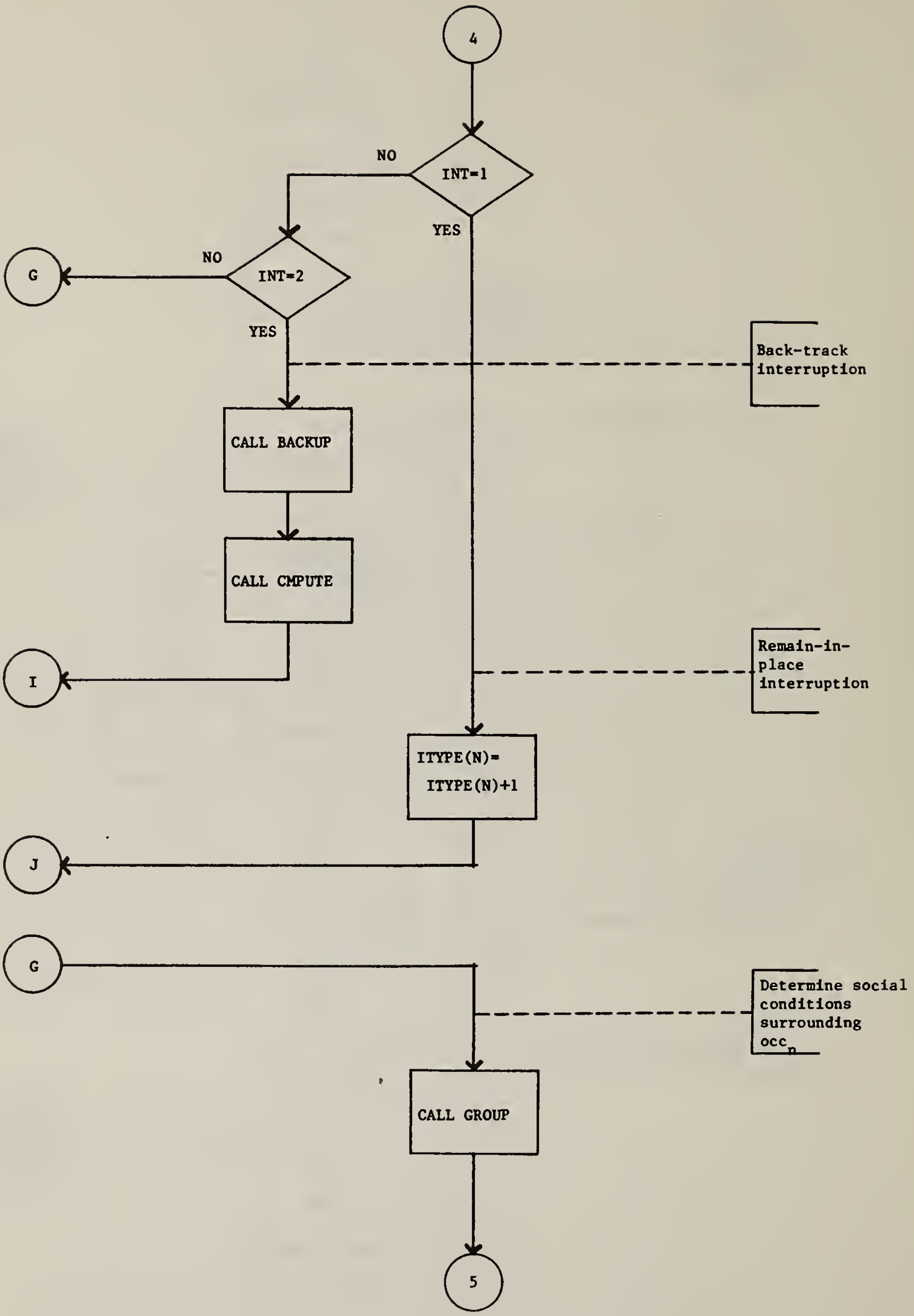

START
Read-in executive and control parameters
CALL INEXEC
Read-in building and fire parameters
CALL INBLDG
Read-in occupant parameters
CALL INOCC
BEGIN TFE SIMULATION HERE
Iterate the experiment for the desired no. of times
III=0
A
III=III+1
Initialize replication parameters
CALL INITLZ
Iterate for desired number of time frames
ITIME=0
B
ITIME=ITIME+1
1

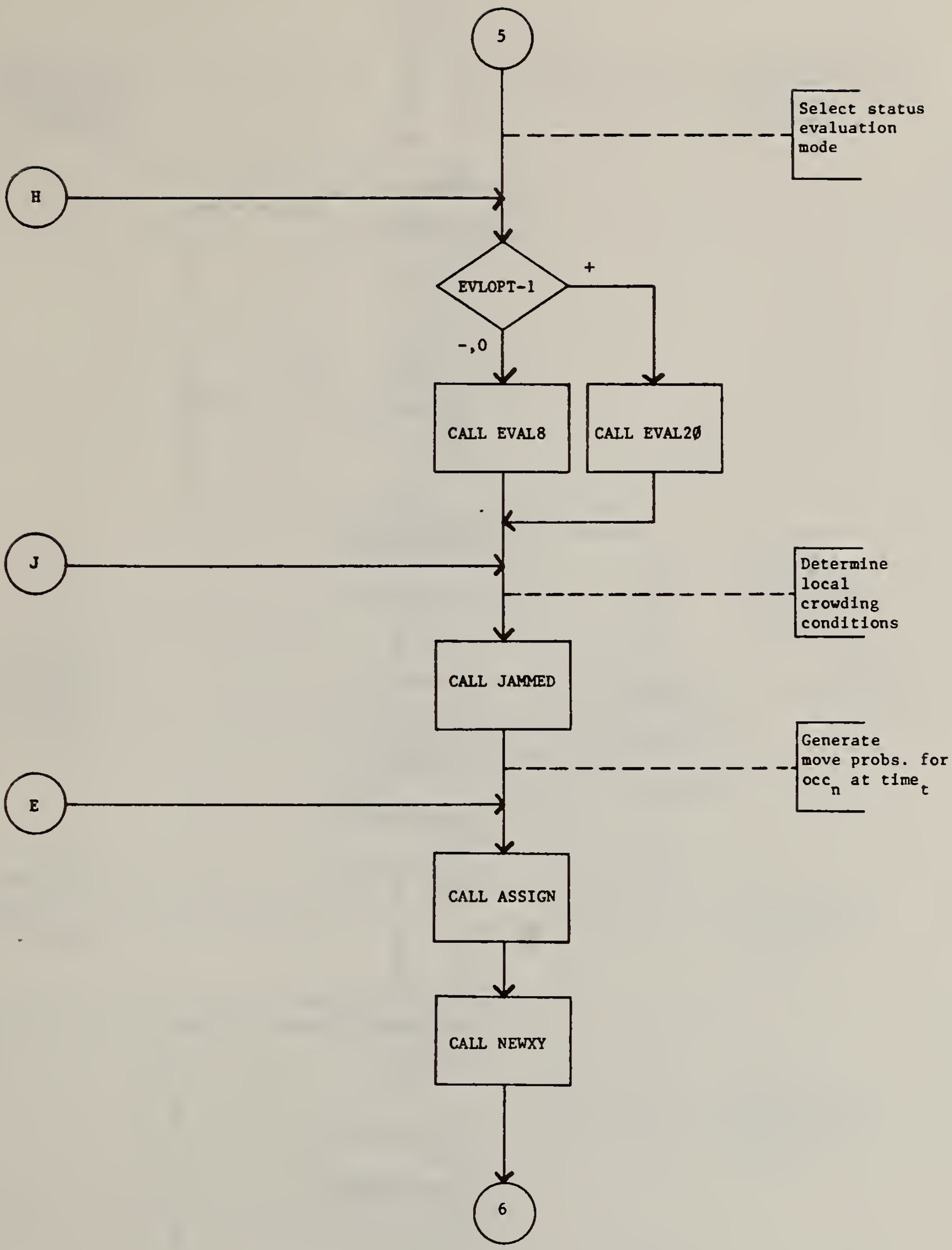
Figure 5.1 Flow Chart for EXEC2

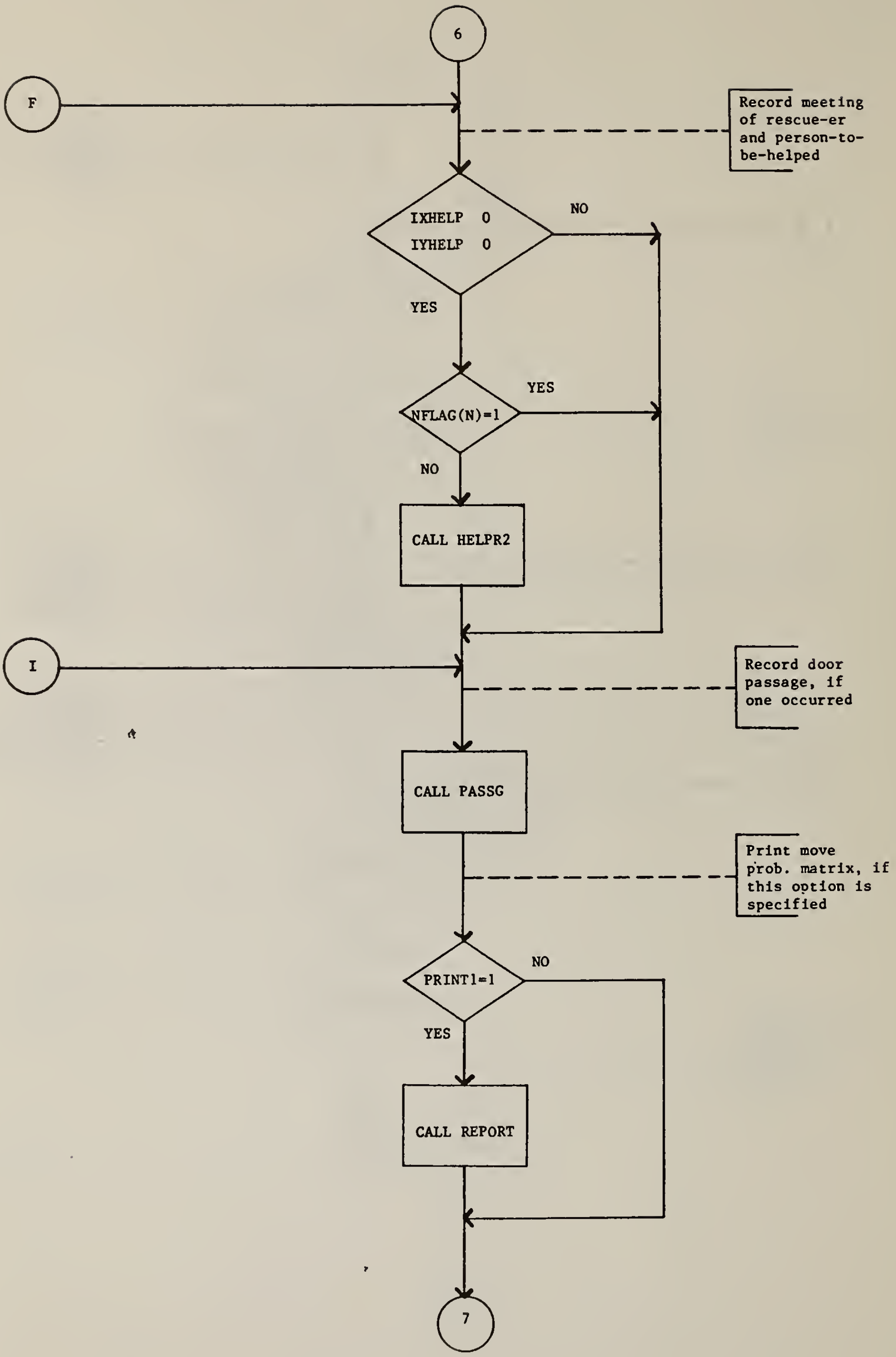

1
Spread smoke one increment
CALL SPREAD
Print updated plot, if plot option specified
PRINT3=1
NO
YES
CALL PLOT
Iterate for desired number of occupants
N=0
N=N+1
C
Initialize occupant locators
CALL INITFR
2

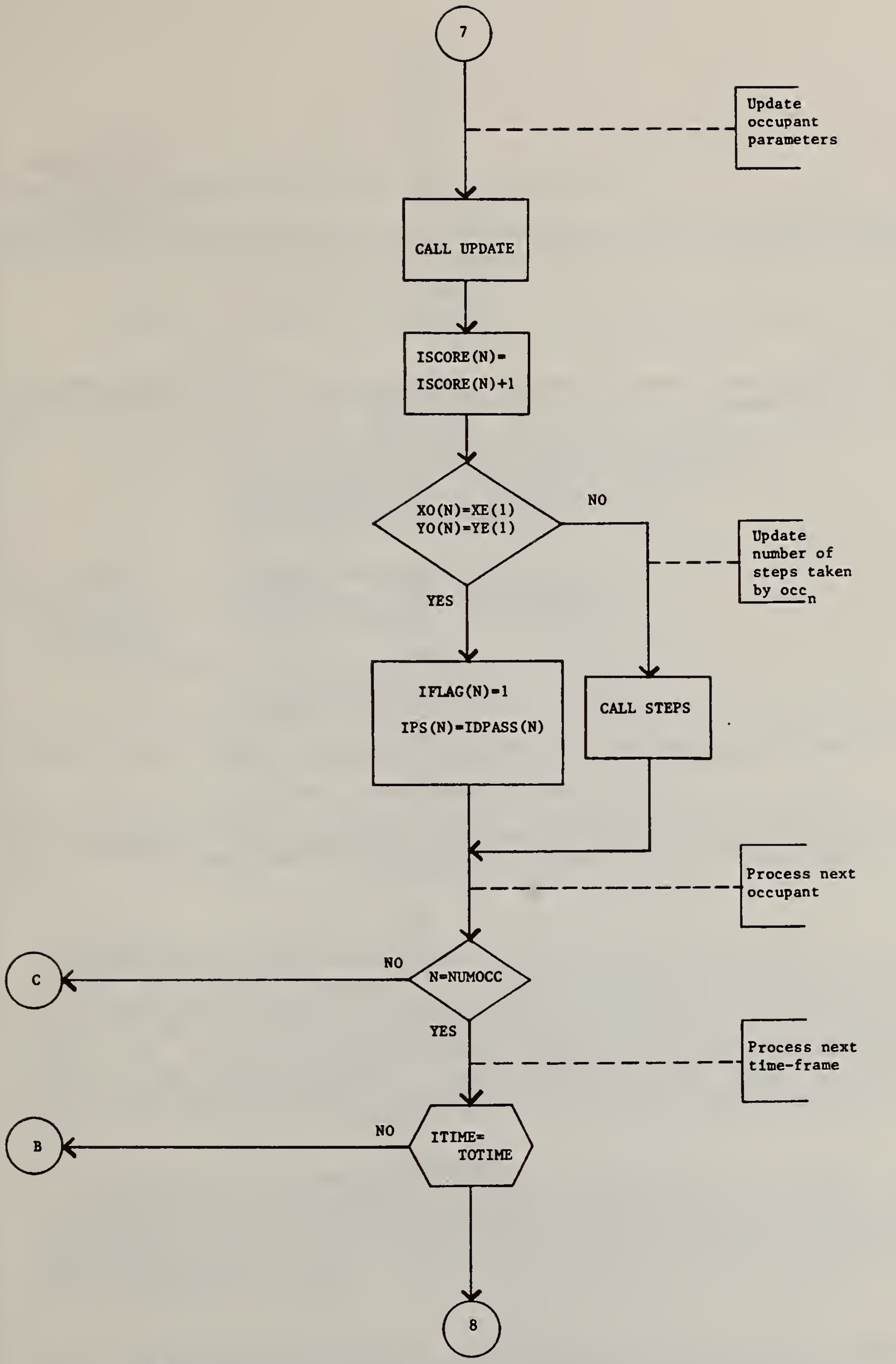

2
Determine which space occ_n is now occupying
N=0
N=N+1
XLO(N) < XO(N) < XHI(N)
YLO(N) < YO(N) < YHI(N)
YES
NO
YES
N < NSPACE
NO
IS=NSPACE
TOTBAR=
NPOINT(IS)
Rescue occ_n if required, and if his helper is present
YES
D
IHANDI(N)≠2
NO
NO
E
IFLAG(N)=1
YES
3

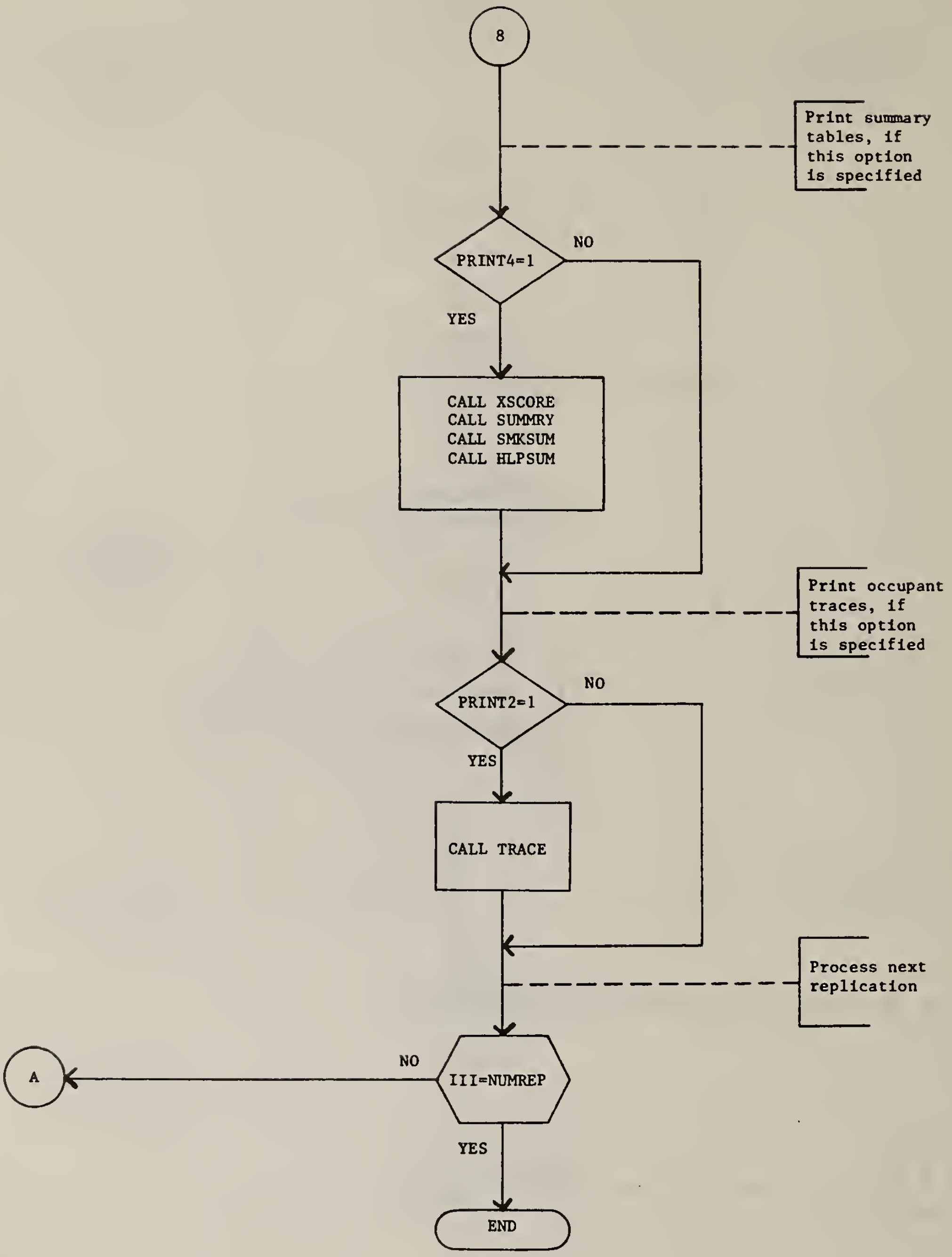

3
CALL HELPED
CALL CMPUTE
F
D
Engage in rescuing, if occ$_n$ is a rescue-er
IXHELP > 0
IYHELP > 0
NO
YES
CALL RESCUE
II=2
YES
G
II=0
NO
YES
H
II=1
NO
(II=2)
Determine whether occ$_n$ is interrupted
CALL INTRPT
4


4
INT=1
NO
YES
INT=2
NO
G
YES
Back-track interruption
CALL BACKUP
CALL CMPUTE
I
Remain-in-place interruption
ITYPE(N)= ITYPE(N)+1
J
G
Determine social conditions surrounding occ$_n$
CALL GROUP
5


5
Select status evaluation mode
H
EVLOPT-1
+
-,0
CALL EVAL8
CALL EVAL2Ø
J
Determine local crowding conditions
CALL JAMMED
Generate move probs. for occ$_n$ at time$_t$
E
CALL ASSIGN
CALL NEWXY
6


6
F
Record meeting of rescue-er and person-to-be-helped
IXHELP 0
IYHELP 0
NO
YES
NFLAG(N)=1
YES
NO
CALL HELPR2
I
Record door passage, if one occurred
CALL PASSG
Print move prob. matrix, if this option is specified
PRINT1=1
NO
YES
CALL REPORT
7


7
Update
occupant
parameters
CALL UPDATE
ISCORE(N)=
ISCORE(N)+1
XO(N)=XE(1)
YO(N)=YE(1)
NO
Update
number of
steps taken
by occ$_n$
YES
IFLAG(N)=1
IPS(N)=IDPASS(N)
CALL STEPS
Process next
occupant
C
NO
N=NUMOCC
YES
Process next
time-frame
B
NO
ITIME=
TOTIME
8


8
Print summary tables, if this option is specified
PRINT4=1
NO
YES
CALL XSCORE
CALL SUMMRY
CALL SMKSUM
CALL HLPSUM
Print occupant traces, if this option is specified
PRINT2=1
NO
YES
CALL TRACE
Process next replication
III=NUMREP
NO
A
YES
END

```
C
C
C
C*****  THIS VERSION IS FOR THE  NBS UNIVAC 1108                         *********
C
C********************************************************************************
C********************************************************************************
C*                                                                              *
C*                                                                              *
C*       BFIRES II -- HUMAN EGRESS BEHAVIOR DURING BUILDING FIRES / VERSION 2   *
C*                                                                              *
C*                                                                              *
C********************************************************************************
C********************************************************************************
C*                                                                              *
C*       WRITTEN BY FRED I. STAHL, PH.D. RESEARCH PSYCHOLOGIST                  *
C*                  ENVIRONMENTAL DESIGN RESEARCH DIVISION                      *
C*                  CENTER FOR BUILDING TECHNOLOGY,NEL                          *
C*                  NATIONAL BUREAU OF STANDARDS                                *
C*                  WASHINGTON, DC  20234                                       *
C*                                                                              *
C*       FOR        PROGRAM FOR DESIGN CONCEPTS                                 *
C*                  CENTER FOR FIRE RESEARCH,NEL                                *
C*                  NATIONAL BUREAU OF STANDARDS                                *
C*                                                                              *
C*       IN SUPPORT OF THE                                                      *
C*                                                                              *
C*                  NBS/HEW FIRE-LIFE SAFETY PROGRAM                            *
C*                                                                              *
C********************************************************************************
C*                                                                              *
C*                  REVISED:   DECEMBER, 1979                                   *
C*                                                                              *
C********************************************************************************
C********************************************************************************
C
C
C
C
C         THE BFIRES PROGRAM SIMULATES HUMAN EGRESS BEHAVIOR DURING BUILDING
C    FIRES.  BFIRES IS A DISCRETE TIME STOCHASTIC SIMULATION BASED ON A NON-
C    STATIONARY MARKOV MODEL OF THE BUILDING FIRE PROCESS.  ACCORDING TO THIS
C    MODEL, FIRES MAY BE UNDERSTOOD IN TERMS OF THREE INTERACTING COMPONENTS.
C    THESE ARE  (1) THE FIRE AND ITS BI-PRODUCTS,  (2) THE BUILDING ENCLOSURE,
C    AND  (3) THE HUMAN OCCUPANTS.  EACH POSSESSES UNIQUE CHARACTERISTICS,
C    AND THE BEHAVIOR OF EACH CONTRIBUTES TO THE OVERALL OUTCOME OF ANY FIRE
C    EVENT (I.E., HOW MANY PEOPLE ESCAPED, HOW MUCH TIME WAS REQUIRED FOR ESCAPE,
C    ETC.).
C
C         EXEC2 IS THE SECOND VERSION OF THE BFIRES EXECUTIVE ROUTINE.  THE
C    PURPOSE OF EXEC2 IS TO READ-IN ALL USER-SUPPLIED DATA, AND THEN TO RUN A
C    FIRE EVENT FOR A GIVEN PERIOD OF TIME.  IN ADDITION, EXEC2 PERMITS THE USER
C    TO CONDUCT A NUMBER (UP TO 20) OF REPLICATIONS OF A GIVEN FIRE EVENT, IN A
C    SINGLE COMPUTER RUN.  EXEC2 PERMITS THE SIMULATION OF TOXIC SMOKE EFFECTS,
C    AND OF RESCUE ACTIVITIES.
C
C
C         THE FOLLOWING DATA MUST BE PROVIDED BY THE USER.  THESE DESCRIBE THE
C    FIRE, THE BUILDING ENCLOSURE, AND THE OCCUPANTS FOR THE COMPUTER.
C    THESE DESCRIPTIONS DEFINE THE INITIAL STATE ,TIME-FRAME 1..  DATA ARE INPUT
```

Table 5.1 EXEC2: FORTRAN Listing

```
C   FOR THE FOLLOWING VARIABLES, WHICH ARE LISTED IN THE ORDER OF INPUT...
C
C     TITLE
C     NUMREP       TOTAL NUMBER OF REPLICATIONS DESIRED (MAX=20).
C     TOTIME       TOTAL NUMBER OF TIME FRAMES TO BE RUN (MAX=100).
C     NUMOCC       TOTAL NUMBER OF OCCUPANTS IN THE SIMULATION (MAX=20).
C     IRAND        ANY 5 DIGIT ODD NUMBER, WHEN RUNNING ON A 32-BIT COMPUTER
C                    (ONLY).  WHEN USING THE NBS UNIVAC 1108, A 5 DIGIT DUMMY
C                    NUMBER MUST BE INPUT.
C     EVLOPT       STATUS EVALUATION SELECTOR (1=EVAL8, 2=EVAL20).
C     RREPRT       OUTPUT OPTION SELECTOR (SEE TABLE BELOW).
C     RREPT2       OUTPUT OPTION SELECTOR (SEE TABLE BELOW).
C     PIO          PROBABILITY OF ENCOUNTERING NO INTERRUPTION,  DURING A GIVEN
C                    TIME FRAME.
C     PI2          PROBABILITY OF ENCOUNTERING A 'TYPE-2' INTERRUPTION DURING A
C                    GIVEN TIME FRAME (PI1 = PIO - PI2).
C     IALLOW       CROWDING FACTOR, I.E., THE MAX NUMBER OF PERSONS PERMITTED
C                    AT ANY PERSON-OCCUPIABLE LOCATION, AT ANY POINT IN TIME.
C     NUMEXT       NUMBER OF EXITS FROM THE FLOOR (MAX=2).
C     NSPACE       NUMBER OF SPATIAL SUBDIVISIONS ON THE FLOOR (MAX=20).
C     ND           NUMBER OF DOORS ON THE FLOOR (MAX=30).
C     MAXX,MAXY    HIGHEST X AND HIGHEST Y COORDS USED TO DEFINE THE FLOORPLAN.
C     XE,YE        X,Y COORDINATES OF EXITS FROM THE FLOOR.
C     NE           NUMBER OF EXITS FROM A PARTICULAR SPATIAL SUBDIVISION ,MAX=2)
C     NPOINT       NUMBER OF GRID POINTS COMPRISING WALLS WHICH ENCLOSE A
C                    PARTICULAR SPATIAL SUBDIVISION.
C     IBAR,IS,1,J) X COORD OF THE JTH POINT OF A WALL ENCLOSING SPACE IS.
C     IBAR(IS,2,J) Y COORD OF THE JTH POINT OF A WALL ENCLOSING SPACE IS.
C     XLO,XHI      RANGE OF X COORDS OF POINTS COMPRISING AN ENCLOSURE.
C     YLO,YHI      RANGE OF Y COORDS OF POINTS COMPRISING AN ENCLOSURE.
C     IDOOR(1,I)   X COORD OF THE LOCATION OF THE ITH DOOR.
C     IDOOR(2,I)   Y COORD OF THE LOCATION OF THE ITH DOOR.
C     IDOOR(3,I)   TYPE IDENTIFIER FOR ITH DOOR (0=MANUAL, 1=AUTO).
C     IDOOR(4,I)   INITIAL POSITION OF ITH DOOR (0=CLOSED, 1=OPEN).
C     XT,YT        X,Y COORDS OF INITIAL FIRE LOCATION.
C     ISRATE       FIRE DIFFUSION RATE FACTOR.
C     XO,YO        X,Y COORDS OF INITIAL OCCUPANT LOCATIONS.
C     INTLIM       OCCUPANTS' INTERRUPTION LIMIT.
C     IHANDI       OCCUPANT'S MOBILITY STATUE (0=UNIMPAIRED, 1=IMPAIRED BUT
C                    MOBILE, 2=IMMOBILE--MUST BE ASSISTED).
C     KNOWAY       OCCUPANTS' INITIAL KNOWLEDGE OF BEST EXIT.
C     TSMOKE       OCCUPANTS' SMOKE TOLERANCE FACTOR.
C     IXHELP       X COORD OF A PERSON-TO-BE-HELPED (PTBH) BY THIS OCCUPANT.
C     IYHELP       Y COORD OF A PERSON-TO-BE-HELPED (PTBH) BY THIS OCCUPANT.
C     POPEN        PROBABILITY THAT THE OCCUPANT WILL OPEN A CLOSED DOOR.
C     PCLOSE       PROBABILITY THAT THE OCCUPANT WILL CLOSE AN OPEN DOOR.
C     IDHELP       RESCUERS' I.D. NUMBER (0, IF OCCUPANT IS NOT DESIGNATED AS A
C                    RESCUER).
C     ITEND        RESCUEES' I.D. NUMBER (0, IF OCCUPANT IS NOT DESIGNATED AS A
C                    RESCUEE).
C     IGOALX       X COORD OF BEST EXIT FROM A SPATIAL SUBDIVISION (NOT FLOOR).
C     IGOALY       Y COORD OF THE BEST EXIT FROM A SPATIAL SUBDIVISION.
C
C   THESE DATA ARE ENTERED IN THE FOLLOWING SEQUENCE...
C
C*******************************************************************************
C
C::SYSTEM/OPERATIONS DATA INPUT::
C
C        TITLE   (ANY 80 CHARACTER COMMENT)
```

```
C             (20A4)
C        NUMREP,TOTIME,NUMOCC,IRAND
C             (3(I3,1X),I5)
C        EVLOPT,RREPRT,RREPT2
C             (3(I2))
C        PIO,PI2,IALLOW
C             (2(F4.2,1X),I2)
C-----------------------------------------------------------------------
C
C::BUILDING/FIRE DATA INPUT::
C
C        NUMEXT,NSPACE,ND
C             (3(I2,1X))
C        MAXX,MAXY
C             (2(I2,1X))
C        (XE(I),I=1,NUMEXT),(YE(I),I=1,NUMEXT)
C             (4(I2,1X))
C        FOR EACH SPACE IN THE FLOORPLAN, CONSTRUCT THE FOLLOWING MINIFILE:
C               NE,NPOINT
C                    (2(I2,1X))
C               IBAR(IS,I,1),I=1,TOTBAR
C                    (FREE)
C               IBAR(IS,I,2),I=1,TOTBAR
C                    (FREE)
C               XLO,XHI,YLO,YHI
C                    (4(I2,1X))
C        IDOOR(I,1),I=1,ND
C             (FREE)
C        IDOOR(I,2),I=1,ND
C             (FREE)
C        IDOOR(I,3),I=1,ND
C             (FREE)
C        IDOOR(I,4),I=1,ND
C             (FREE)
C        XT,YT,ISRATE
C             (3(I2,1X))
C-----------------------------------------------------------------------
C
C::OCCUPANT DATA INPUT::
C
C        FOR EACH OCCUPANT IN THE SIMULATION, CONSTRUCT THE FOLLOWING
C        MINIFILE:
C               XO,YO,INTLIM,IHANDI,KNOWAY,TSMOKE,IXHELP,IYHELP,POPEN,
C               PCLOSE,PHELP,IDHELP,ITEND
C                    (8(I2,1X),3(F4.2,1X),2(I2,1X))
C        FOR EACH SPACE (IS) ON THE FLOOR, CONSTRUCT THE FOLLOWING MINIFILE
C        DESCRIBING OCCUPANTS EGRESS GOALS
C               (IGOALX(IS,IEXIT),IEXIT=1,NE(IS))
C                    (FREE)
C               (IGOALY(IS,IEXIT),IEXIT=1,NE(IS))
C                    (FREE)
C
C********************************************************************************
C
C   SEVERAL OUTPUT OPTIONS ARE AVAILABLE TO THE USER.  THESE ARE:
C
C     (1) REPORT (STATE OF ALL OCCUPANTS, GIVEN FOR EACH TIME FRAME).
C     (2) TRACE  (LOCATION OF EACH OCC, AS THIS VARIES FROM FRAME TO
C                 FRAME).
C     (3) PLOT   (GRAPHIC DESCRIPTION OF SMOKE AND OCCUPANT LOCATIONS).
```

```
C     (4) SUMMRY (OCCUPANT EGRESS SUMMARY), AND
C         SMKSUM (OCCUPANT SMOKE EXPOSURE SUMMARY), AND
C         HLPSUM (RESCUE ACTIVITY SUMMARY).
C
C   THESE MAY BE CALLED INDIVIDUALLY OR COLLECTIVELY, IN ACCORDANCE WITH
C   THE USER'S SPECIFICATION OF INPUT VARIABLES RREPRT & RREPT2, AS
C   FOLLOWS:
C
C               OPTION             RREPRT   RREPT2
C               -----------------------------------
C               (1)                    1        0
C               (2)                    2        0
C               (3)                    3        0
C               (4)                    1        1
C               (1)(2)                 2        1
C               (2)(3)                 3        1
C               (1)(3)                 1        2
C               (1)(4)                 2        2
C               (2)(4)                 3        2
C               (3)(4)                 0        3
C               (1)(2)(3)              1        3
C               (1)(2)(4)              2        3
C               (1)(2)(3)(4)           3        3
C
C
C********************************************************************************
C
C
      DIMENSION NOEXP(20),LOWEXP(20),IHIEXP(20),ITOTAL(20),ITOXIC(20)
      DIMENSION ITYPE1(20),ITYPE2(20),IDPASS(20),IPS(20)
      DIMENSION IXTRCE(20,100),IYTRCE(20,100),IDHELP(20),ITEND(20)
      DIMENSION IBACK(20),JTIME(20),INITYO(20),INITXO(20)
      DIMENSION INTR(20),INTNUM(20),TITLE(20)
      DIMENSION IENTER(9),IAGREE(20),IREACH(20),NHELP(20)
      DIMENSION IBAR(20,75,2),IHANDI(20),KNOWAY(20),NFLAG(20)
      DIMENSION INTLIM (20),IBYSTD(20),NE(20),NPOINT(20)
      DIMENSION PTDIST(20),PEDIST(20),P(9)
      DIMENSION IGOALX(20,10,20),IGOALY(20,10,20),KXO(20),KYO(20)
      DIMENSION POPEN(20),PCLOSE(20),IDOOR(30,4),IDOPEN(30,100)
      DIMENSION IFLAG(20),ISCORE(20),NUMSTP(20),PM(9,100)
      DIMENSION PHELP(20),IXHELP(20),IYHELP(20),ISMOKE(100,100)
      INTEGER XPRIOR(20),YPRIOR(20),SCORE(20),TSMOKE(20)
      INTEGER XT,YT,XO(20),YO(20),XE(10),YE(10),TOTBAR,TOTIME
      INTEGER XLO(20),XHI(20),YLO(20),YHI(20),EVLOPT
      INTEGER XOB(20,100),YOB(20,100),RREPRT,RREPT2
      INTEGER PRINT1,PRINT2,PRINT3,PRINT4
C
C  DUMMY VARIABLE(S):
      DO 5001 N=1,20
5001  IBYSTD(N)=0
C
C  READ-IN EXECUTIVE AND CONTROL VARIABLES
      CALL INEXEC (IR,IP,NUMREP,TOTIME,NUMOCC,IRAND,EVLOPT,
     1  RREPRT,RREPT2,PIO,PI2,IALLOW,TITLE,
     2  PRINT1,PRINT2,PRINT3,PRINT4)
C  READ-IN BUILDING AND FIRE DESCRIPTORS
      CALL INBLDG (NUMEXT,NSPACE,ND,XE,YE,NE,NPOINT,IBAR,
     1  TOTBAR,XLO,XHI,YLO,YHI,IDOOR,IR,XT,YT,ISRATE,
     2  MAXX,MAXY)
C  READ-IN OCCUPANT PARAMETERS
```

```
      CALL INOCC (IR,NUMOCC,XO,YO,INTLIM,IHANDI,KNOWAY,
     1  POPEN,PCLOSE,TSMOKE,PHELP,IXHELP,IYHELP,KXO,KYO,NE,
     2  INITXO,INITYO,NSPACE,NUMEXT,IGOALX,IGOALY,IDHELP,ITEND)
C
C
C
C
C **             EXECUTE THE SIMULATION EXPERIMENT          **
C
C
C
C
C  ITERATE FOR DESIRED NUMBER OF REPLICATIONS
      DO 90 III=1,NUMREP
C  INITIALIZE REPLICATION VARIABLES
      CALL INITLZ (NUMOCC,XO,YO,KXO,KYO,LOWEXP,NOEXP,
     1 IHIEXP,ITOTAL,ITOXIC,IFLAG,NUMSTP,ISCORE,IBACK,
     2  JTIME,INTR,INTNUM,ITYPE1,ITYPE2,IDPASS,MAXX,
     3  MAXY,XT,YT,ISMOKE,IREACH,NHELP,NFLAG,KNOWAY,IAGREE)
C
C  ITERATE FOR DESIRED NUMBER OF TIME FRAMES
C
      DO 50 ITIME=1,TOTIME
C  SPREAD SMOKE AT RATE INPUT BY USER
      CALL SPREAD (ISMOKE,IBAR,IDOOR,NPOINT,MAXX,MAXY,
     1  NSPACE,ND,ISRATE,ITIME)
      CONTINUE
C  PRINT SUMMARY PLOT FOR CURRENT TIME FRAME
      IF (PRINT3.EQ.1) GO TO 100
      GO TO 110
100   CALL PLOT (XO,YO,NUMOCC,ITIME,MAXX,MAXY,ISMOKE)
110   CONTINUE
C
      DO 501 I=1,ND
501   IDOPEN(I,ITIME)=IDOOR(I,4)
C
C  ITERATE FOR DESIRED NUMBER OF OCCUPANTS
      DO 60 NTHIS=1,NUMOCC
      IF (IFLAG(NTHIS).EQ.1) GO TO 60
C  INITALIZE OCCUPANT LOCATORS
      CALL INITFR (NTHIS,XO,YO,ITIME,XPRIOR,YPRIOR,
     1  IXTRCE,IYTRCE,XOB,YOB)
C  DETERMINE WHICH SPACE THE OCCUPANT IS CURRENTLY OCCUPYING
      N=0
15    N=N+1
      IF (((XLO(N).LT.XO(NTHIS)).AND.
     1   (YLO(N).LT.YO(NTHIS))).AND.
     2   ((XHI(N).GT.XO(NTHIS)).AND.
     3   (YHI(N).GT.YO(NTHIS)))) GO TO 25
      GO TO 20
25    IS=N
      GO TO 26
20    IF (N.LT.NSPACE) GO TO 15
      IS=NSPACE
26    TOTBAR=NPOINT(IS)
C  RESCUE THIS OCCUPANT, IF THIS IS REQ'D, AND IF HIS ASSIGNED HELPER
C  IS PRESENT:
      IF (IHANDI(NTHIS).NE.2) GO TO 157
      IF (NFLAG(NTHIS).EQ.1) GO TO 155
      GO TO 158
```

```
155   CALL HELPED (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1  XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,IRAND,
     2  P,MOVE,XK,YK,K,NUMOCC,IDHELP,ITEND,NEWXO,NEWYO)
      CALL CMPUTE (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1  XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,IRAND,P,MOVE,
     2  XK,YK,K,L,IS,IGOALX,IGOALY,IENTER,X,IDOOR,POPEN,ND,
     3  MDOOR,PCLOSE,NOEXP,LOWEXP,IHIEXP,ITOTAL,ISMOKE,
     4  TSMOKE,ITOXIC,TOTIME)
      GO TO 156
157   CONTINUE
C  PROCESS RESCUE ACTIVITY IF OCC(N) IS A DESIGNATED RESCUE-ER
      IF ((IXHELP(NTHIS).GT.0).AND.(IYHELP(NTHIS).GT.0)) GO TO 150
      GO TO 151
150   CALL RESCUE (ITIME,NTHIS,PHELP,NHELP,XO,YO,IXHELP,IYHELP,
     1  NSPACE,XLO,XHI,YLO,YHI,NE,IGOALX,IGOALY,IAGREE,IGX,IGY,
     2  IAG,IXX,IREACH,IHANDI,II)
      GO TO 152
151   II=2
152   CONTINUE
      IF (II.EQ.0) GO TO 31
      IF (II.EQ.1) GO TO 65
C  DETERMINE WHETHER THE OCCUPANT IS INTERRUPTED
      CALL INTRPT (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1   XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,
     2   IRAND,P,MOVE,XK,YK,K,L,IS,IGOALX,IGOALY,IENTER,
     3   X,INTLIM,INTR,INTNUM,PI2,PIO)
      IF (INT.EQ.1) GO TO 27
      IF (INT.EQ.2) GO TO 30
      GO TO 31
C  REMAIN-IN-PLACE INTERRUPTION
27    ITYPE1(NTHIS)=ITYPE1(NTHIS)+1
      GO TO 70
C  BACK-TRACK INTERRUPTION
30    CONTINUE
      CALL BACKUP (IBACK,XO,YO,INITXO,INITYO,XOB,YOB,
     1   ITIME,NTHIS,NEWXO,NEWYO,INTR,JTIME)
      ITYPE2(NTHIS)=ITYPE2(NTHIS)+1
      IF (INTR(NTHIS).EQ.0) GO TO 31
      CALL CMPUTE (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1  XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,IRAND,P,MOVE,
     2  XK,YK,K,L,IS,IGOALX,IGOALY,IENTER,X,IDOOR,POPEN,ND,
     3  MDOOR,PCLOSE,NOEXP,LOWEXP,IHIEXP,ITOTAL,ISMOKE,
     4  TSMOKE,ITOXIC,TOTIME)
      GO TO 71
31    CONTINUE
C  DETERMINE SOCIAL CONDITIONS SURROUNDING THE OCCUPANT
      CALL GROUP (NTHIS,NUMOCC,IHANDI,KNOWAY,KOOCC,NHANDI,NKNOW,NAGREE,
     1   IAGREE)
      IF (NHANDI.GT.0) GO TO 65
      GO TO 67
65    CONTINUE
C  DETERMINE MODE OF STATUS EVALUATION
67    IF (EVLOPT-1) 68,68,69
68    CALL EVAL8(XO,YO,XT,YT,XE,YE,NTHIS,IAGREE,ITIME,IEVAL,
     1   PTDIST,TDIST,PEDIST,EDIST,IS,IGOALX,IGOALY)
      GO TO 70
69    CALL EVAL20 (MXTIME,MK,XO,YO,XE,YE,NTHIS,IAGREE,
     1   ITIME,C,IEVAL,TOTIME,XT,YT)
C  DETERMINE LOCAL CROWDING CONDITIONS SURROUNDING THE OCCUPANT
70    CALL JAMMED (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1   XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,IRAND,
     2   P,MOVE,XK,YK,K,IALLOW,NUMOCC,IENTER)
C  GENERATE MOVE PROBABILITIES
158    CALL ASSIGN (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1   XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,
     2   IRAND,P,MOVE,XK,YK,K,L,IS,IGOALX,IGOALY,IENTER,
     3   X,IDOOR,POPEN,ND,MDOOR,PCLOSE,NOEXP,LOWEXP,IHIEXP,
     4   ITOTAL,ISMOKE,TSMOKE,ITOXIC,TOTIME,NHELP,PM)
      CALL NEWXY (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1   XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,
     2   IRAND,P,MOVE,XK,YK,K,NEWXO,NEWYO)
156   CONTINUE
C  RECORD MEETING OF RESCUE-ER WITH PTBH
      IF ((IXHELP(NTHIS).GT.0).AND.(IYHELP(NTHIS).GT.0)) GO TO 153
      GO TO 71
153   IF (NFLAG(NTHIS).EQ.1) GO TO 71
      CALL HELPR2 (IREACH,NTHIS,XO,YO,IXHELP,IYHELP,ITIME,NFLAG,
     1  NUMOCC,INITXO,INITYO)
C  RECORD DOOR-PASSAGE, IF ONE OCCURRED
71    CALL PASSG (IDPASS,IDOOR,XO,YO,NTHIS,ND,NEWXO,NEWYO)
      IF (PRINT1.EQ.1) GO TO 72
      GO TO 61
C  PRINT MOVE PROBABILITY MATRIX, IF THIS OPTION IS SELECTED
72    CONTINUE
      CALL REPORT (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1  XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,
     2  IRAND,P,MOVE,XK,YK,K,NUMEXT,NUMOCC,TOTIME,INTLIM,
     3  KNOWAY,PTDIST,TDIST,PEDIST,EDIST,NEWXO,NEWYO,
     4   EVLOPT,IDOOR,IDOPEN,ND,INTR)
C  UPDATE OCCUPANT LOCATORS
61    CALL UPDATE (XO,YO,NTHIS,NEWXO,NEWYO)
      ISCORE(NTHIS)=ISCORE(NTHIS)+1
      IF ((XO(NTHIS).EQ.XE(1)).AND.
     1   (YO(NTHIS).EQ.YE(1))) GO TO 62
      GO TO 66
62    IFLAG(NTHIS)=1
      IPS(NTHIS)=IDPASS(NTHIS)
      GO TO 60
C  UPDATE NUMBER OF STEPS TRAVERSED BY THE OCCUPANT
66    CALL STEPS (XPRIOR,YPRIOR,XO,YO,NUMSTP,NTHIS)
60    CONTINUE
50    CONTINUE
C  PRINT SUMMARY TABLE, IF THIS OPTION IS SELECTED
      IF (PRINT4.EQ.1) GO TO 63
      GO TO 64
C  COMPUTE ESCAPE SCORES FOR ALL OCCUPANTS IN THE RUN
63    CALL XSCORE (TOTIME,ISCORE,NUMOCC,SCORE)
      CALL SUMMRY (INITXO,INITYO,INTLIM,LBYSTD,IHANDI,KNOWAY,POPEN,
     1   PCLOSE,SCORE,NUMSTP,IPS,III,NUMREP,NUMOCC,TOTIME,XO,YO,XE,YE,
     2   TITLE,IXHELP,IYHELP,TSMOKE,PHELP)
      CALL SMKSUM (III,NUMOCC,NOEXP,LOWEXP,IHIEXP,ITOTAL)
      CALL HLPSUM (IFLAG,NFLAG,NUMOCC,ISCORE,IREACH,III,NUMREP,
     1  TOTIME,IDHELP,ITEND,IP)
64    CONTINUE
C  PRINT TRACES, IF THIS OPTION IS SELECTED:
      IF (PRINT2.EQ.1) GO TO 120
      GO TO 90
120   CALL TRACE (IXTRCE,IYTRCE,NTHIS,ITIME,NUMOCC,TOTIME)
90    CONTINUE
      END
```

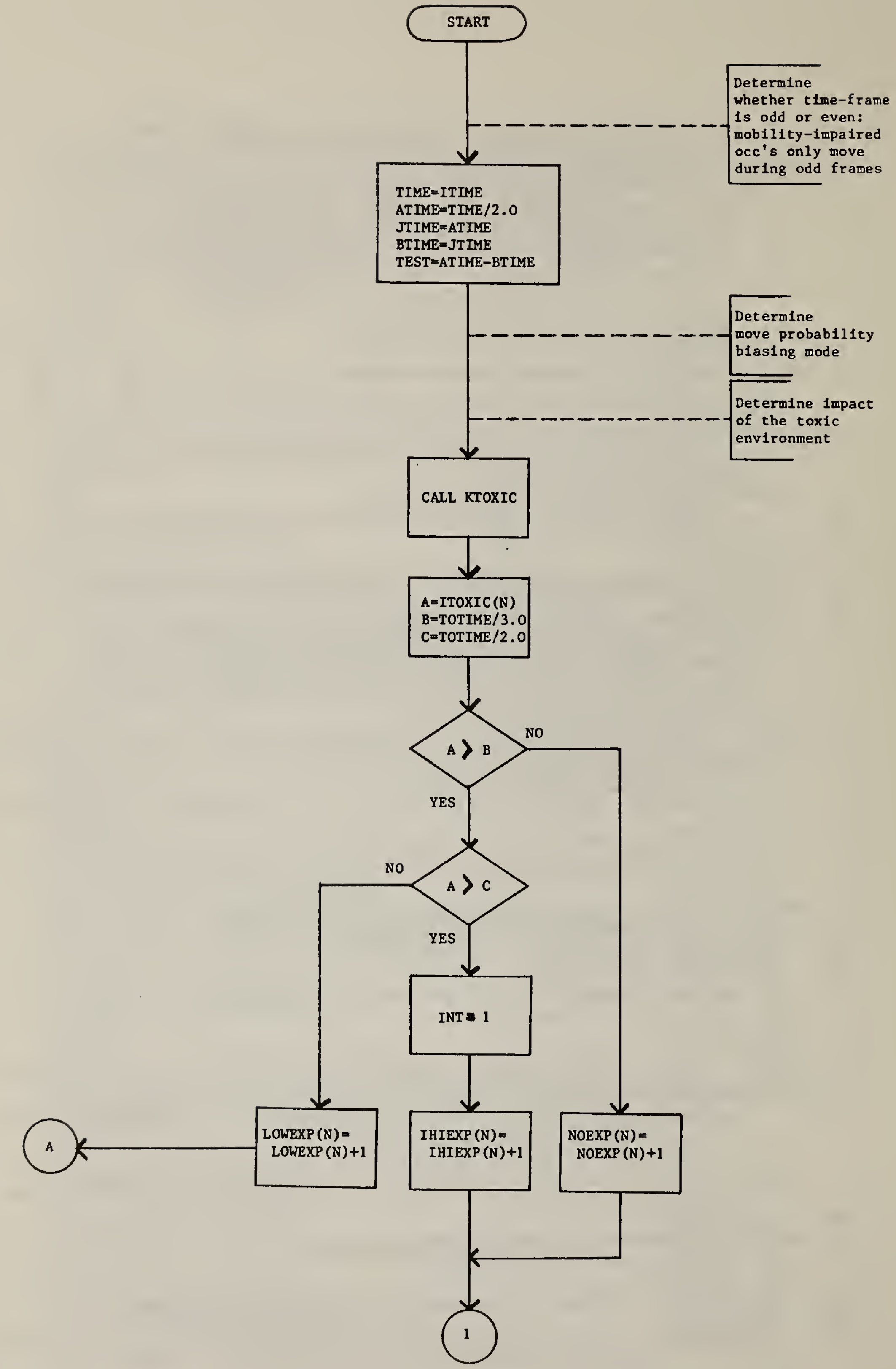
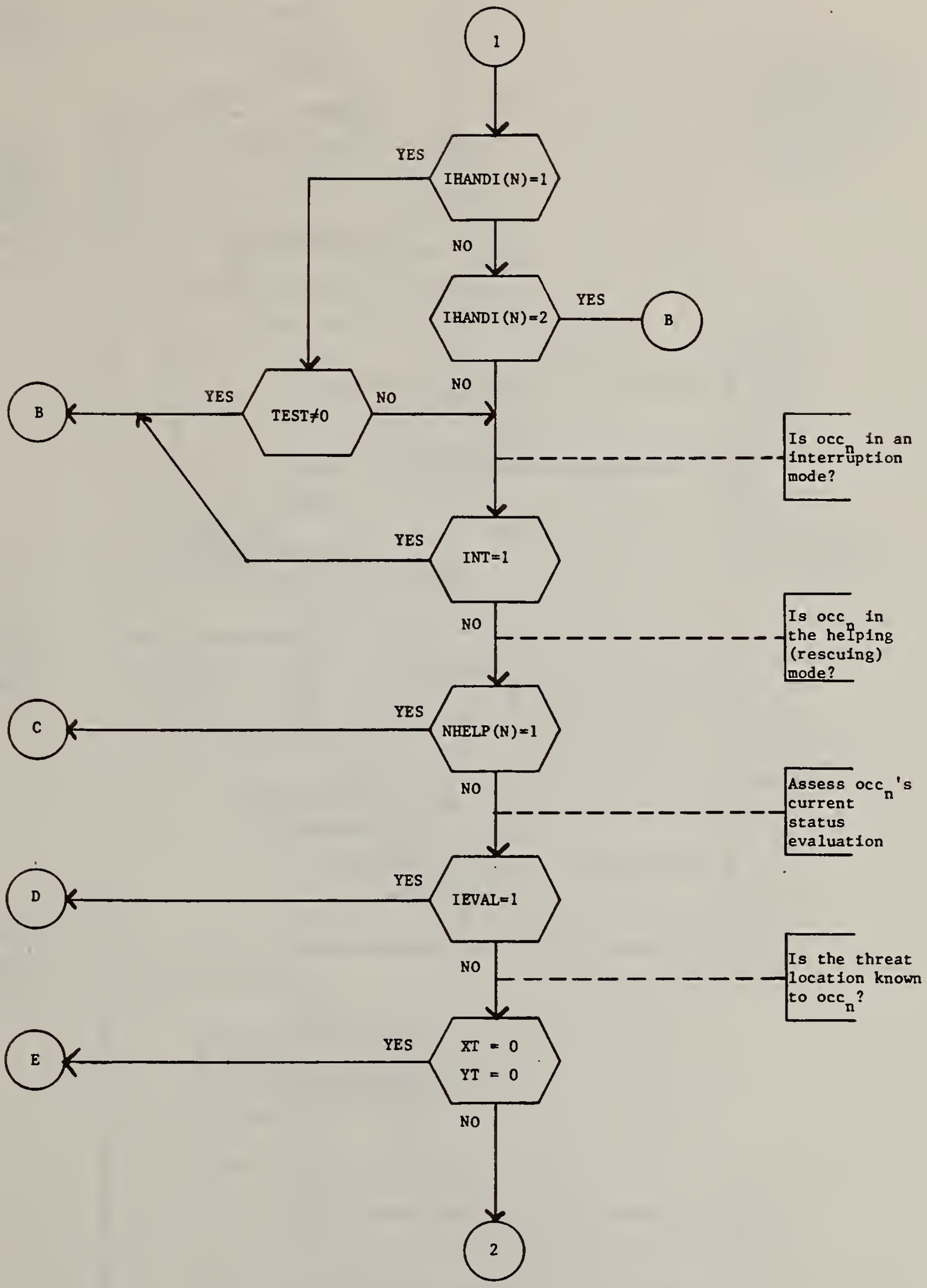
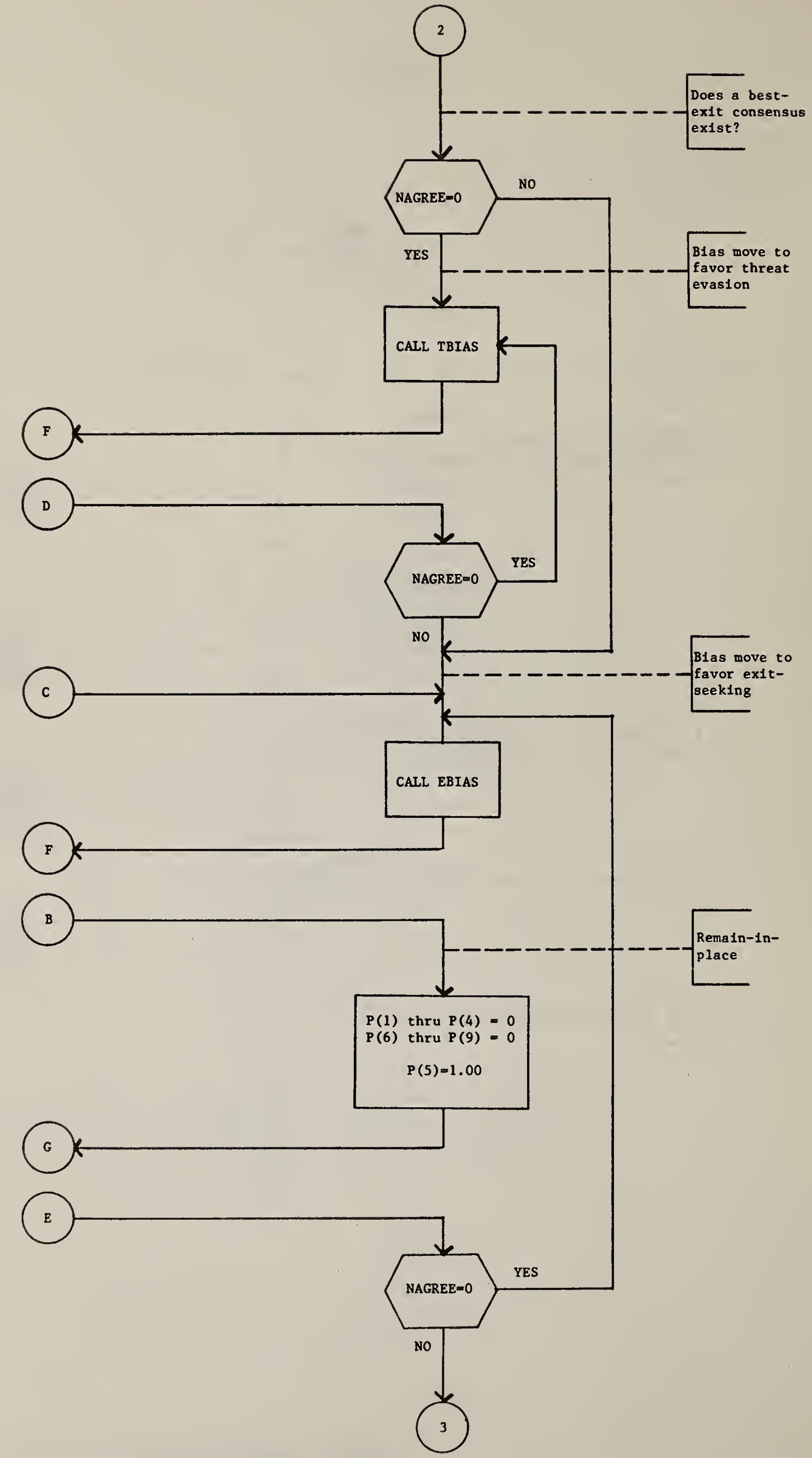
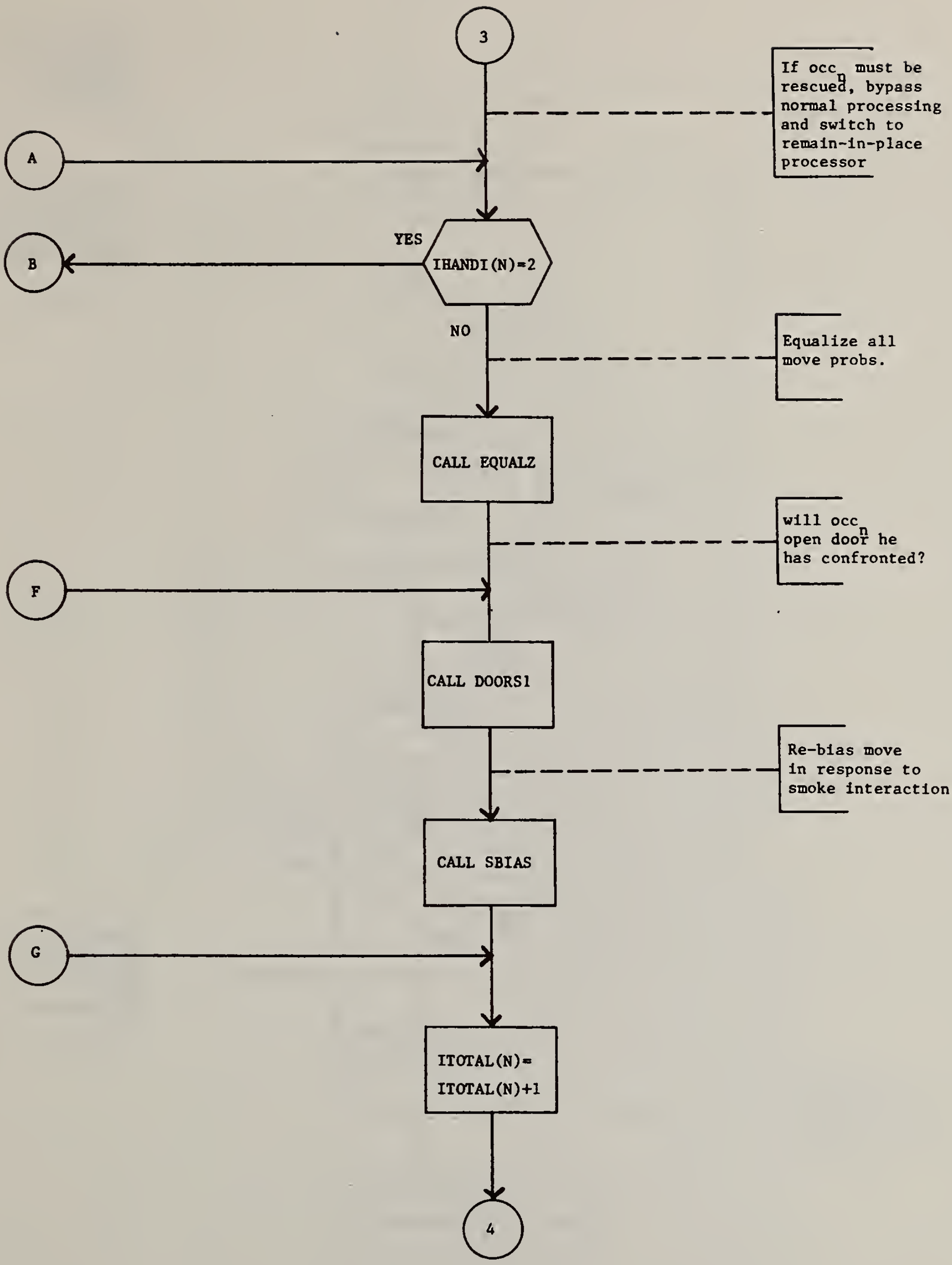
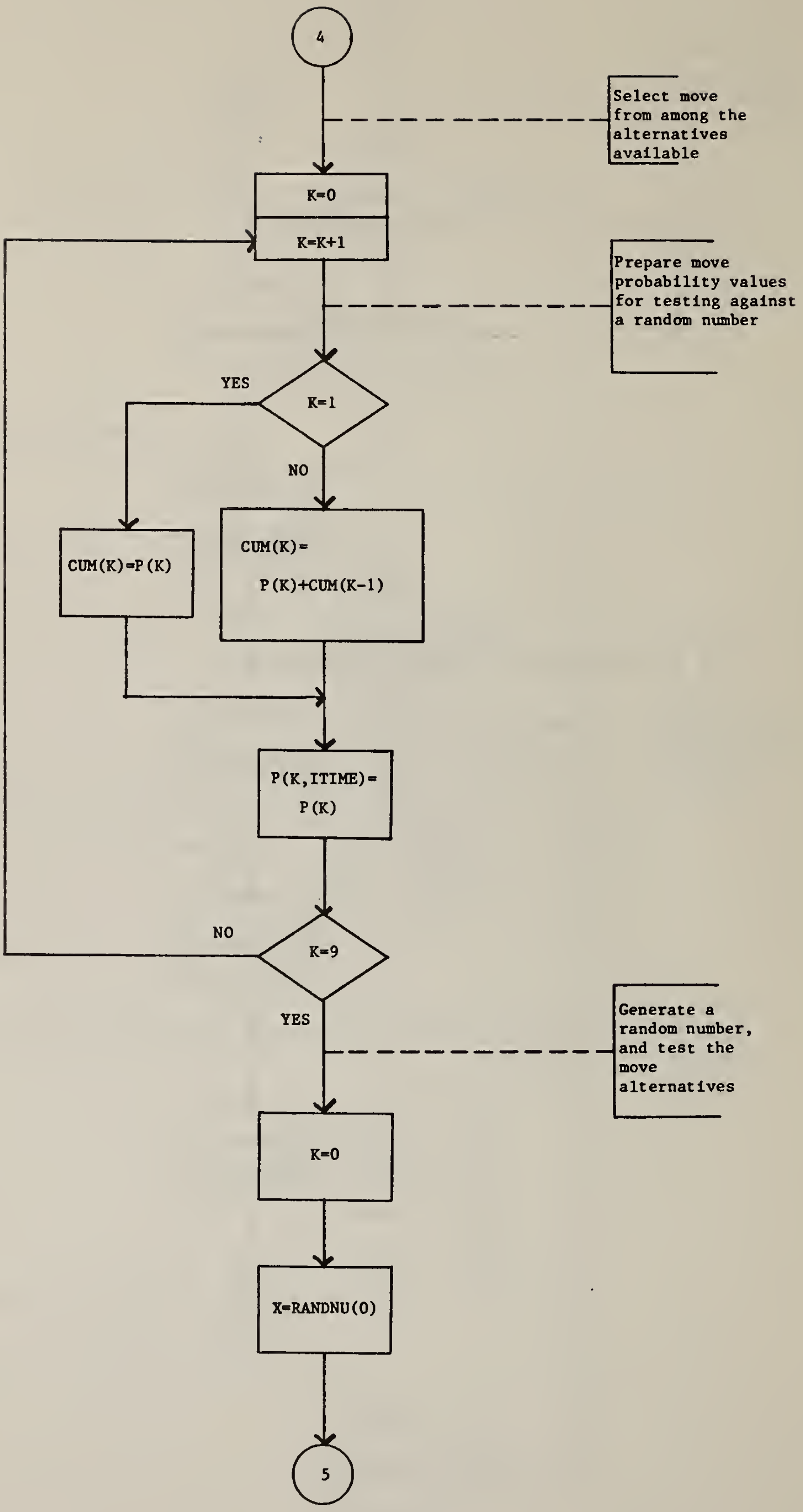


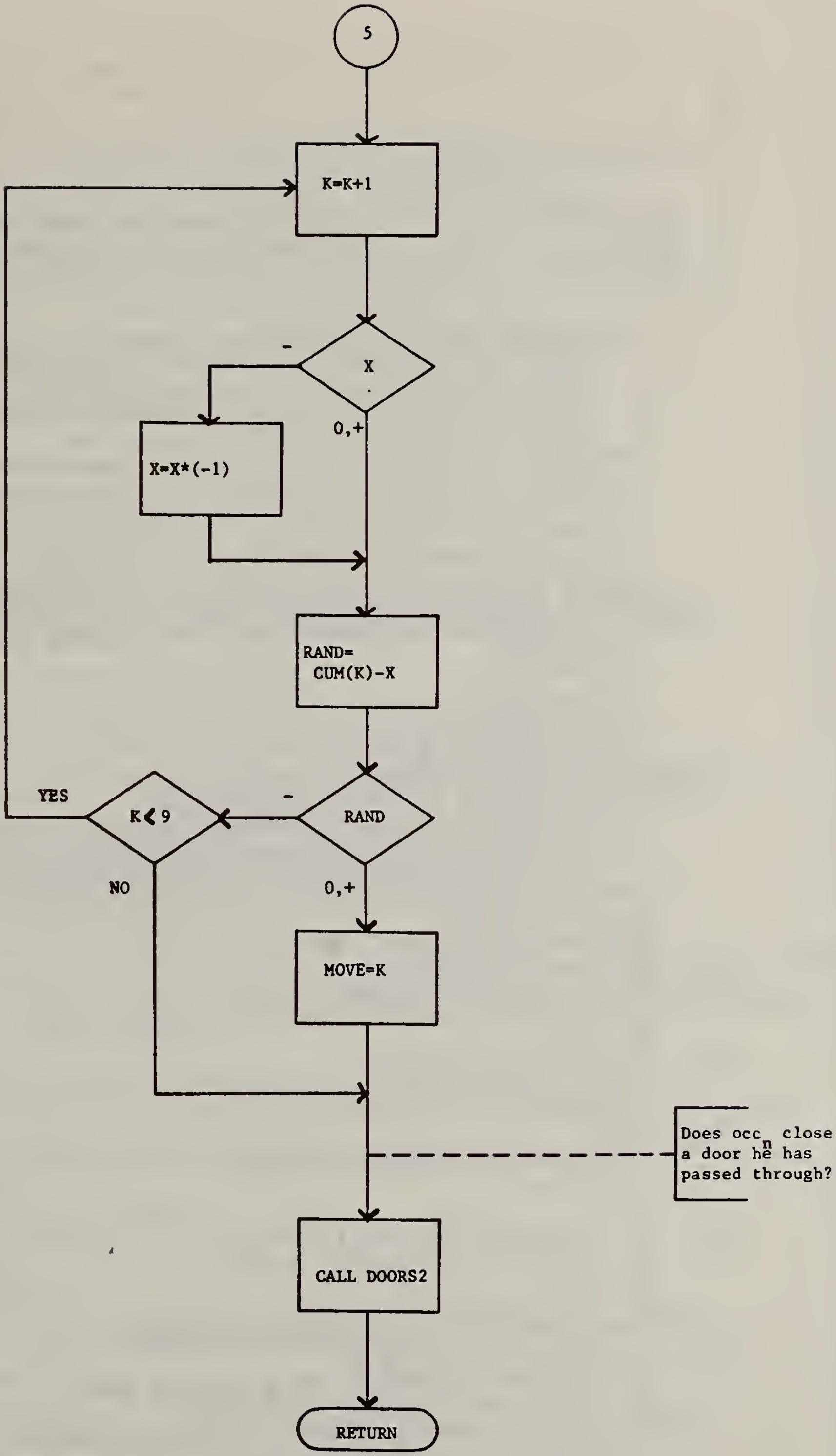
Figure 5.2 Flow Chart for Subroutine ASSIGN


1
IHANDI(N)=1
YES
NO
IHANDI(N)=2
YES
B
NO
TEST≠0
YES
NO
B
Is occ$_n$ in an interruption mode?
INT=1
YES
NO
Is occ$_n$ in the helping (rescuing) mode?
NHELP(N)=1
YES
C
NO
Assess occ$_n$'s current status evaluation
IEVAL=1
YES
D
NO
Is the threat location known to occ$_n$?
XT = 0
YT = 0
YES
E
NO
2


2
Does a best-exit consensus exist?
NAGREE=0
NO
YES
Bias move to favor threat evasion
CALL TBIAS
F
D
NAGREE=0
YES
NO
C
Bias move to favor exit-seeking
CALL EBIAS
F
B
Remain-in-place
P(1) thru P(4) = 0
P(6) thru P(9) = 0
P(5)=1.00
G
E
NAGREE=0
YES
NO
3


3
If occ$_n$ must be rescued, bypass normal processing and switch to remain-in-place processor
A
YES
B
IHANDI(N)=2
NO
Equalize all move probs.
CALL EQUALZ
will occ$_n$ open door he has confronted?
F
CALL DOORS1
Re-bias move in response to smoke interaction
CALL SBIAS
G
ITOTAL(N)=
ITOTAL(N)+1
4


4
Select move from among the alternatives available
K=0
K=K+1
Prepare move probability values for testing against a random number
YES
K=1
NO
CUM(K)=P(K)
CUM(K)=
P(K)+CUM(K-1)
P(K,ITIME)=
P(K)
NO
K=9
YES
Generate a random number, and test the move alternatives
K=0
X=RANDNU(0)
5


5
K=K+1
X
-
0,+
X=X*(-1)
RAND=
CUM(K)-X
RAND
-
K< 9
YES
NO
0,+
MOVE=K
Does occ$_n$ close
a door he has
passed through?
CALL DOORS2
RETURN

```
C
C
C
C
C   SUBROUTINE ASSIGN
C
C   THE PURPOSE OF ASSIGN IS TO CONSIDER ALL FACTORS COMPRISING THE
C   OCCUPANT'S CURRENT PERCEPTION OF HIS SITUATION, AND THEN TO
C   SWITCH CONTROL FOR DECISION MAKING TO THE APPROPRIATE BIASING
C   ROUTINE.
C
      SUBROUTINE ASSIGN (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1   XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,
     2   IRAND,P,MOVE,XK,YK,K,L,IS,IGOALX,IGOALY,IENTER,
     3   X,IDOOR,POPEN,ND,MDOOR,PCLOSE,NOEXP,LOWEXP,IHIEXP,ITOTAL,
     4   ISMOKE,TSMOKE,ITOXIC,TOTIME,NHELP,PM)
      DIMENSION IENTER(9),IBAR(20,75,2),IDOOR(30,4),POPEN(20),
     1  PCLOSE(20),IAGREE(20),IGOALX(20,10,20),IGOALY(20,10,20)
      DIMENSION NOEXP(20),LOWEXP(20),IHIEXP(20),ITOTAL(20),
     1  ITOXIC(20),ISMOKE(100,100),NHELP(20),PM(9,100)
      DIMENSION IHANDI(20),IBYSTD(20),P(9),CUM(9)
      INTEGER XO(20),YO(20),XE(10),YE(10),XT,YT,TSMOKE(20),TOTIME
C   DETERMINE WHETHER ITIME, THE CURRENT TIME FRAME, IS ODD OR EVEN...
C   MOBILITY-IMPAIRED OCCUPANTS ONLY MOVE DURING ODD TIME FRAMES
      TIME=ITIME
      ATIME=TIME/2.
      JTIME=ATIME
      BTIME=JTIME
      TEST=ATIME-BTIME
C   DETERMINE MOVE PROBABILITY BIASING MODE
C   DETERMINE IMPACT OF TOXIC ENVIRONMENT
      CALL KTOXIC (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1  XO,YO,IBAR,XT,YT,NAGREE,XE,YE,IAGREE,IRAND,P,MOVE,XK,
     2  YK,K,L,IS,IGOALX,IGOALY,IENTER,X,IDOOR,POPEN,ND,MDOOR,
     3  PCLOSE,NOEXP,LOWEXP,IHIEXP,ITOTAL,ISMOKE,TSMOKE,ITOXIC)
      A=ITOXIC(NTHIS)
      B=FLOAT(TOTIME)/3.
      C=FLOAT(TOTIME)/2.
      IF (A.GT.B) GO TO 300
      NOEXP(NTHIS)=NOEXP(NTHIS)+1
      GO TO 303
300   IF (A.GT.C)GO TO 301
      LOWEXP(NTHIS)=LOWEXP(NTHIS)+1
      GO TO 302
301   INT=1
      IHIEXP(NTHIS)=IHIEXP(NTHIS)+1
      GO TO 303
C
303   IF (IHANDI(NTHIS).EQ.1) GO TO 1
      IF (IHANDI(NTHIS).EQ.2) GO TO 2
      GO TO 11
1     IF (TEST.NE.0.) GO TO 2
C    IS THE OCCUPANT IN AN INTERRUPTION MODE?
11    IF (INT.EQ.1) GO TO 2
C    IS THE OCCUPANT CURRENTLY IN A HELPING MODE?
      IF (NHELP(NTHIS).EQ.1) GO TO 504
C    ASSESS THE OCCUPANT'S CURRENT STATUS EVALUATION
      IF (IEVAL.EQ.1) GO TO 4
C    IS THE THREAT LOCATION KNOW TO THE OCCUPANT?
      IF ((XT.EQ.0).AND.(YT.EQ.0)) GO TO 5
```

Table 5.2 Subroutine ASSIGN: FORTRAN Listing

```
C  DOES A BEST-EXIT CONSENSUS EXIST (IF SO, NAGREE.NE.0)?
         IF (NAGREE.EQ.0) GO TO 404
         GO TO 504
C  BIAS MOVE PROBABILITIES TO FAVOR THREAT EVASION
404      CALL TBIAS (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
        1  XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,
        2  IRAND,P,MOVE,XK,YK,K,L,IS,IGOALX,IGOALY,IENTER)
         GO TO 6
4        IF (NAGREE.EQ.0) GO TO 404
C  BIAS MOVE PROBABILITIES TO FAVOR EXIT-SEEKING
504      CALL EBIAS (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
        1  XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,
        2  IRAND,P,MOVE,XK,YK,K,L,IS,IGOALX,IGOALY,IENTER)
         GO TO 6
C  REMAIN-IN-PLACE OPTION ENFORCED
2        P(5)=1.0
         DO 50 K=1,4
50       P(K)=0.0
         DO 51 K=6,9
51       P(K)=0.0
         GO TO 601
5        IF (NAGREE.NE.0) GO TO 504
C  IF OCC(N) IS IMMOBILE AND MUST BE RESCUED, BYPASS STANDARD
C  PROCESSING SEQUENCE, AND SWITCH CONTROL TO 'REMAIN-IN-PLACE'
C  PROCESSOR:
302      IF (IHANDI(NTHIS).EQ.2) GO TO 2
C  EQUALIZE ALL MOVE PROBABILITIES:
         CALL EQUALZ (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
        ~~1  XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,~~
        2  IRAND,P,MOVE,XK,YK,K,L,IS,IGOALX,IGOALY,IENTER)
C  DETERMINE WHETHER THE OCCUPANT WILL OPEN A DOOR HE HAS CONFRONTED
6        CALL DOORS1 (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
        1   XO,YO,IBAR,TOTBAR,XT,YT, NAGREE,XE,YE,IAGREE,
        2   IRAND,P,MOVE,XK,YK,K,L,IS,IGOALX,IGOALY,IENTER,
        3   X,IDOOR,POPEN,ND,MDOOR,PCLOSE,NOEXP,LOWEXP,IHIEXP,
        4   ITOTAL,ISMOKE,TSMOKE,IX)
C  BIAS MOVE PROBABILITIES IN RESPONSE TO SMOKE INTERACTION
         CALL SBIAS(ITIME,NTHIS,IHANDI,INIT,IBYSTD,IEVAL,XO,YO,IBAR,TOTBAR,
        1  XT,YT,NAGREE,XE,YE,IAGREE,IRAND,P,MOVE,XK,YK,K,L,IS,IGOALX,
        2  IGOALY,IENTER,X,IDOOR,POPEN,ND,MDOOR,PCLOSE,NOEXP,LOWEXP,IHIEXP,
        3  ITOTAL,ISMOKE,TSMOKE)
601      CONTINUE
         ITOTAL(NTHIS)=ITOTAL(NTHIS)+1
C  THE OCCUPANT SELECTS A MOVE FROM AMONG THE ALTERNATIVES AVAILABLE
         DO 100 K=1,9
C  PREPARE ALTERNATIVE MOVE PROBABILITY VALUES FOR TESTING AGAINST A
C  RANDOM NUMBER
         IF (K.EQ.1) GO TO 200
         CUM(K)=P(K)+CUM(K-1)
         GO TO 100
200      CUM(K)=P(K)
100      PM(K,ITIME)=P(K)
C  GENERATE A RANDOM NUMBER, X, AND TEST THE MOVE ALTERNATIVES
         K=0
         X=RANDNU(0)
7        K=K+1
         IF (X) 71,72,72
71       X=X*(-1)
72       RAND=CUM(K)-X
         IF (RAND) 52,53,53
```

```
53    MOVE=K
      GO TO 8
52    IF (K.LT.9) GO TO 7
C  DETERMINE WHETHER THE OCCUPANT CLOSES A DOOR JUST PASSED THROUGH
C  (IF IN FACT HE HAS JUST DONE SO!)
8     CALL DOORS2(ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1   XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,
     2   IRAND,P,MOVE,XK,YK,K,L,ISIGOALX,IGOALY,IENTER,
     3   X,IDOOR,POPEN,ND,MDOOR,PCLOSE,NOEXP,LOWEXP,IHIEXP,
     4   ITOTAL,ISMOKE,TSMOKE,IX)
12    RETURN
      END
```

6.0 SUMMARY

Recent modifications and enhancements of the BFIRES computer program are discussed and are presented as companion material to an earlier documentation of BFIRES/VERSION 1. Several shortcomings of BFIRES I are addressed. Chief among these are BFIRES I's inability to simulate either rescue activities or direct interactions between occupants and the toxic environment. The report documents new subroutines designed to mitigate these shortcomings. The routines are clustered into two modules new to BFIRES/VERSION 2: a "smoke" module and a "rescue" module. Additional improvements to the original BFIRES program involve clearer data file management and input facilities, and expanded output capabilities. Finally, BFIRES I routines requiring modification for operation within the BFIRES II environment are discussed.

7.0 REFERENCES

1. Archea, J. The Evacuation of Non-Ambulatory Patients from Hospital and Nursing Home Fires: A Framework for a Model. Washington, D.C.: U.S. Dept. of Commerce, National Bureau of Standards, NBSIR 79-1906, 1979.

2. Berl, W.G. and Halpin, B.M. Human Fatalities from Unwanted Fires. Laurel, Md.: The Johns Hopkins University Applied Physics Laboratory, APL/JHU FFP TR37, December 1978.

3. Phillips, A.W. The Physiological and Psychological Effects of Fires in High-rise Buildings. Factory Mutual Record, May-June 1973, 8-10.

4. Stahl, F.I. Final Report on the BFIRES/VERSION 1 Computer Simulation of Emergency Egress Behavior During Fires: Calibration and Analysis. Washington, D.C.: U.S. Dept. of Commerce, National Bureau of Standards, NBSIR 79-1713, 1979.

APPENDIX A. BFIRES LISTING

This appendix contains FORTRAN listings of all BFIRES subroutines not specifically covered by this report. Documentary material for there subroutines are provided in the earlier report on BFIRES/VERSION I (Stahl, 1979).

```
      SUBROUTINE AGREE (NTHIS,NUMOCC,IHANDI,KNOWAY,KOOCC,
     1  NHANDI,NKNOW,NAGREE,IAGREE)
      DIMENSION KNOWAY(20),IAGREE(20)
      KONE=0
      KTWO=0
      KZERO=0
      DO 50 I=1,NUMOCC
      IF (KNOWAY(I).EQ.1) GO TO 51
      IF (KNOWAY(I).EQ.2) GO TO 52
      IF (KNOWAY(I).EQ.0) GO TO 53
      GO TO 50
   51 KONE=KONE+1
      GO TO 50
   52 KTWO=KTWO+1
      GO TO 50
   53 KZERO=KZERO+1
   50 CONTINUE
      ONE=KONE
      TWO=KTWO
      ZERO=KZERO
      SUM=ONE+TWO+ZERO
      PSUM=.60*SUM
      IF ((ONE.GE.PSUM).OR.(TWO.GE.PSUM)) GO TO 1
      NAGREE=0
      IAGREE(NTHIS)=0
      GO TO 2
    1 NAGREE=1
      IF (ONE.GE.PSUM) GO TO 3
      IAGREE(NTHIS)=2
      GO TO 2
    3 IAGREE(NTHIS)=1
    2 RETURN
      END
```

```
      SUBROUTINE OTHERS (NTHIS,NUMOCC,IHANDI,KNOWAY,KOOCC,NHANDI,
     1   NKNOW)
      DIMENSION IHANDI(20),KNOWAY(20)
      NKNOW=0
      NHANDI=0
      IF (NUMOCC.GT.1) GO TO 1
      KOOCC=0
      GO TO 999
    1 KOOCC=1
      DO 50 I=1,NUMOCC
      IF (I.EQ.NTHIS) GO TO 50
      IF (IHANDI(I).EQ.1) GO TO 51
      GO TO 50
   51 NHANDI=NHANDI+1
   50 CONTINUE
      DO 60 I=1,NUMOCC
      IF (I.EQ.NTHIS) GO TO 60
      IF (KNOWAY(I).GT.0) GO TO 61
      GO TO 60
   61 NKNOW=NKNOW+1
   60 CONTINUE
  999 RETURN
      END
```

```
      SUBROUTINE GROUP (NTHIS,NUMOCC,IHANDI,KNOWAY,KOOCC,NHANDI,
     1   NKNOW,NAGREE,IAGREE)
      DIMENSION KNOWAY(20),IHANDI(20),IAGREE(20)
      CALL OTHERS (NTHIS,NUMOCC,IHANDI,KNOWAY,KOOCC,NHANDI,NKNOW)
      IF (KNOWAY(NTHIS).GT.0) GO TO 1
      IF (KOOCC.EQ.0) GO TO 999
      GO TO 2
    1 IF (KOOCC.EQ.0) GO TO 999
      IF (NKNOW.EQ.0) GO TO 999
    2 CALL AGREE (NTHIS,NUMOCC,IHANDI,KNOWAY,KOOCC,NHANDI,
     1  NKNOW,NAGREE,IAGREE)
  999  RETURN
      END
      SUBROUTINE EQUALZ(ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1  XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,
     2  IRAND,P,MOVE,XK,YK,K,L,IS,IGOALX,IGOALY,IENTER)
C
C: ^EQUALZ^ (EQUALIZE) CAUSES THE PROBABILITY VALUES OF THE
C: VARIOUS POSSIBLE MOVES TO BE SET EQUAL TO EACH OTHER, RESULTING IN
C: NO BIASING EFFECT:
C
      INTEGER XO(20),YO(20),XK,YK
      DIMENSION IMPOSS (9), P (9)
      NUMPOS=0
      DO 1 K=1,9
      CALL KPOSS (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1  XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,
     2  IRAND,P,MOVE,XK,YK,K,L,IS,IGOALX,IGOALY,IENTER)
      IF (L.EQ.1) GO TO 2
      GO TO 3
2     NUMPOS=NUMPOS+1
      IMPOSS(K)=0
      GO TO 1
3     IMPOSS(K)=1
1     CONTINUE
      DO 4 K=1,9
      IF (IMPOSS(K).EQ.1) GO TO 5
      P(K)=1.0/FLOAT(NUMPOS)
      GO TO 4
5     P(K)=0.0
4     CONTINUE
      RETURN
      END
```

```
      SUBROUTINE  TBIAS (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1  XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,
     1  IRAND,P,MOVE,XK,YK,K,L,IS,IGOALX,IGOALY,IENTER)
C
C: ^TBIAS^ (THREAT-BIAS) CAUSES THE PROBABILITY VALUES OF THE
C: VARIOUS MOVES TO BE ADJUSTED SO AS TO BIAS TOWARD THREAT-
C: REDUCTION (I.E., BIASING TOWARD INCREASING THE
C: DISTANCE BETWEEN OCC(NTHIS) AND THE THREAT POINT):
C
      INTEGER XO(20),YO(20),XT,YT,XK,YK
      DIMENSION M(9),DIST(9),P(9)
      TOTDST=0.
C
C: FOR EACH POSSIBLE MOVE, COMPUTE DISTANCE TO THREAT POINT:
C
      DO 10 K=1,9
      CALL KPOSS (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1  XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,
     2  IRAND,P,MOVE,XK,YK,K,L,IS,IGOALX,IGOALY,IENTER)
      IF (L.EQ.1) GO TO 1
      M(K)=1
      GO TO 10
1     M(K)=0
      DIST(K)=SQRT(FLOAT((XT-XK)**2+(YT-YK)**2))
      TOTDST=TOTDST+DIST(K)
10    CONTINUE
C
C: FOR EACH POSSIBLE MOVE, COMPUTE THE MOVE-PROB., P(K):
C
      DO 15 K=1,9
      IF (M(K).EQ.0) GO TO 2
      P(K)=0.
      GO TO 15
2     P(K)=DIST(K)/TOTDST
15    CONTINUE
      RETURN
      END
      SUBROUTINE EBIAS(ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1 XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,
     2 IRAND,P,MOVE,XK,YK,K,L,IS,IGOALX,IGOALY,IENTER)
C
C: THE DISTANCE BETWEEN OCC(NTHIS) AND THE AGREED:UPON EXIT:
C
      INTEGER XO(20),YO(20),XE(10),YE(10),XK,YK
      DIMENSION M(9),DIST(9),A(9),P(9),IAGREE(20)
      DIMENSION IGOALX(20,10,20),IGOALY(20,10,20)
      TOTDST=0.
      SUMA=0.
      ITEMP=IAGREE(NTHIS)
C
C: FOR EACH POSSIBLE MOVE, COMPUTE THE DISTANCE TO THE
C: AGREED-UPON EXIT:
C
      DO 10 K=1,9
      CALL KPOSS (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1 XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,
     2 IRAND,P,MOVE,XK,YK,K,L,IS,IGOALX,IGOALY,IENTER)
      IF (L.EQ.1) GO TO 1
      M(K)=1
      GO TO 10
1     M(K)=0
      DIST(K)=SQRT(FLOAT((IGOALX(IS,ITEMP,NTHIS)-XK)**2+
     1   (IGOALY(IS,ITEMP,NTHIS)-YK)**2))
      TOTDST=TOTDST+DIST(K)
10    CONTINUE
C
C: FOR EACH POSSIBLE MOVE, COMPUTE MOVE-PROB. VALUES, P(K).
C: IF DIST(K)=0.0, MOVE K IS SELECTED.  IF DIST(K)=0.0 FOR
C: MORE THAN ONE MOVE ALTERNATIVE, THEN THE MOVE-PROBS. FOR
C: THESE ARE EQUALIZED:
C
      DO 15 K=1,9
      IF (M(K).EQ.1) GO TO 15
      IF (DIST(K).EQ.0.) GO TO 15
      A(K)=TOTDST/DIST(K)
      SUMA=SUMA+A(K)
15    CONTINUE
C
      K=0
2     K=K+1
      IF (M(K).EQ.1) GO TO 3
      IF (DIST(K).EQ.0.) GO TO 5
      P(K)=A(K)/SUMA
      GO TO 4
3     P(K)=0.
4     IF (K.LT.9) GO TO 2
      RETURN
5     ZERO=0.
      DO 20 K=1,9
      IF(M(K).EQ.1) GO TO 20
      IF (DIST (K).EQ.0.) GO TO 6
      GO TO 20
6     ZERO=ZERO+1.
20    CONTINUE
      DO 25 K=1,9
      IF (M(K).EQ.1) GO TO 707
      IF (DIST(K).EQ.0.) GO TO 7
707   P(K)=0.0
      GO TO 25
7     P(K)=1./ZERO
25    CONTINUE
      RETURN
      END
      SUBROUTINE KPOSS (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1  XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,IRAND,
     2  P,MOVE,XK,YK,K,L,IS,IGOALX,IGOALY,IENTER)
      DIMENSION IBAR (20,75,2),IENTER(9)
      INTEGER XO(20),YO(20),XK,YK,TOTBAR
      CALL KTOXY (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1  XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,IRAND,
     2  P,MOVE,XK,YK,K)
      ICROSS=(XO(NTHIS)+XK)/2
      JCROSS=(YO(NTHIS)+YK)/2
      I=0
      J=1
 1    I=I+1
      IF (I.GT.TOTBAR) GO TO 5
      IF (IBAR(IS,I,J).EQ.ICROSS) GO TO 2
      GO TO 1
 2    J=J+1
      IF (IBAR(IS,I,J).EQ.JCROSS) GO TO 4
      J=J-1
      GO TO 1
 4     CONTINUE
      L=0
      GO TO 6
 5    L=1
      IF (IENTER(K).EQ.0) GO TO 4
 6    CONTINUE
      RETURN
      END
```

```
      SUBROUTINE JAMMED (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1   XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,IRAND,
     2   P,MOVE,XK,YK,K,IALLOW,NUMOCC,IENTER)
      INTEGER XO(20),YO(20),XK,YK,XE(10),YE(10)
      DIMENSION JAM(9),IENTER(9)
      DO 100 K=1,9
      CALL KTOXY (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1   XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,IRAND,
     2   P,MOVE,XK,YK,K)
      JAM(K)=0
      DO 200 N=1,NUMOCC
      IF (N.EQ.NTHIS) GO TO 200
      IF ((XO(N).EQ.XK).AND.(YO(N).EQ.YK)) GO TO 1
      GO TO 200
 1    JAM(K)=JAM(K)+1
 200  CONTINUE
      IF ((XK.EQ.XE(1)).AND.(YK.EQ.YE(1))) GO TO 2
      IF (JAM(K).LT.IALLOW) GO TO 2
      GO TO 3
 2    IENTER(K)=1
      GO TO 100
 3    IENTER(K)=0
 100  CONTINUE
      RETURN
      END
      SUBROUTINE KTOXY(ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1  XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,IRAND,
     2  P,MOVE,XK,YK,K)
      INTEGER XO(20),YO(20),XK,YK
      GO TO (1,2,3,4,5,6,7,8,9),K
 1    XK=XO(NTHIS)-2
      YK=YO(NTHIS)-2
      GO TO 10
 2    XK=XO(NTHIS)-2
      YK=YO(NTHIS)
      GO TO 10
 3    XK=XO(NTHIS)-2
      YK=YO(NTHIS)+2
      GO TO 10
 4    XK=XO(NTHIS)
      YK=YO(NTHIS)-2
      GO TO 10
 5    XK=XO(NTHIS)
      YK=YO(NTHIS)
      GO TO 10
 6    XK=XO(NTHIS)
      YK=YO(NTHIS)+2
      GO TO 10
 7    XK=XO(NTHIS)+2
      YK=YO(NTHIS)-2
      GO TO 10
 8    XK=XO(NTHIS)+2
      YK=YO(NTHIS)
      GO TO 10
 9    XK=XO(NTHIS)+2
      YK=YO(NTHIS)+2
 10   RETURN
      END
```

```
      SUBROUTINE DOORS1 (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1   XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,
     2   IRAND,P,MOVE,XK,YK,K,L,IS,IGOALX,IGOALY,IENTER,
     3   X,IDOOR,POPEN,ND,MDOOR,PCLOSE,NOEXP,LOWEXP,IHIEXP,
     4   ITOTAL,ISMOKE,TSMOKE,IX)
      DIMENSION P(9),IDOOR(30,4),POPEN(20)
      INTEGER XO(20),YO(20),XK,YK,TOTBAR
C
C: DETERMINE WHETHER OCC(I) ENCOUNTERS A DOOR:
C
      K=0
 5    K=K+1
      CALL KTOXY (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1   XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,
     2   IRAND,P,MOVE,XK,YK,K)
      ICROSS=(XO(NTHIS)+XK)/2
      JCROSS=(YO(NTHIS)+YK)/2
      I=0
      J=0
      J=1
 3    I=I+1
      IF (I.GT.ND) GO TO 1
      IF (IDOOR(I,J).EQ.ICROSS) GO TO 2
      GO TO 3
 2    J=J+1
      IF (IDOOR(I,J).EQ.JCROSS) GO TO 4
      J=J-1
      GO TO 3
 1    MDOOR=0
      IF (K.EQ.9) GO TO 999
      GO TO 5
      C
      C: DOOR IS ENCOUNTERED:
      C
      4     IX=I
            MDOOR=K
      C
      C: IF DOOR IS ALREADY OPEN, RETURN:
      C
            IF (IDOOR(I,4).EQ.1) GO TO 999
            X=RANDNU(0)
            IF (X) 55,56,56
      55    X=X*(-1)
      56    IF (X.LT.POPEN(NTHIS)) GO TO 6
      C
      C: OCC(I) OPENS CLOSED DOOR:
      C
            IDOOR(I,4)=1
            RETURN
      C
      C: OCC(I) LEAVES DOOR CLOSED:
      C
      6     P(MDOOR)=0.0
      C
      C: OCC(I) CHOSES TO LEAVE DOOR CLOSED:
      C: REDISTRIBUTE MOVE PROBABILITIES:
      C
            K=0
            SUM=0.0
            NPOSS=0
      7     K=K+1
            IF (K.GT.9) GO TO 9
            CALL KPOSS (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
           1   XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,
           2   IRAND,P,MOVE,XK,YK,K,L,IS,IGOALX,IGOALY,IENTER)
            IF (L.EQ.1) GO TO 8
            GO TO 7
      8     NPOSS=NPOSS+1
            SUM=SUM+P(K)
            GO TO 7
      9     DIFF=1.0-SUM
            SHARE=DIFF/FLOAT(NPOSS)
            DO 25 K=1,9
            CALL KPOSS (ITIME,NTHIS,IHANDI,INT,IBYSTD, IEVAL,
           1   XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,
           2   IRAND,P,MOVE,XK,YK,K,L,IS,IGOALX,IGOALY,IENTER)
            IF (L.EQ.1) GO TO 10
            GO TO 25
      10    P(K)=P(K)+SHARE
      25    CONTINUE
      999   RETURN
            END
```

```
            SUBROUTINE DOORS2 (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
           1   XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,
           2   IRAND,P,MOVE,XK,YK,K,L,IS,IGOALX,IGOALY,IENTER,
           3   X,IDOOR,POPEN,ND,MDOOR,PCLOSE,NOEXP,LOWEXP,IHIEXP,
           4   ITOTAL,ISMOKE,TSMOKE,IX)
            DIMENSION IDOOR(30,4),PCLOSE(20)
      C
      C: DETERMINE WHETHER OCC(I) CLOSES (A MANUAL) DOOR BEHIND HIM:
      C
            IF (MDOOR.EQ.0) RETURN
            IF (MOVE.EQ.MDOOR) GO TO 1
            RETURN
      1     IF (IDOOR(IX,3).EQ.0) GO TO 2
            RETURN
      2     X=RANDNU(0)
            IF (X) 55,56,56
      55    X=X*(-1)
      56    IF (X.LT.PCLOSE(NTHIS)) RETURN
            IDOOR(IX,4)=0
            RETURN
            END
      SUBROUTINE EVAL8 (XO,YO,XT,YT,XE,YE,NTHIS,
     1   IAGREE,ITIME,IEVAL,PTDIST,TDIST,PEDIST,
     1   EDIST,IS,IGOALX,IGOALY)
      INTEGER XO(20),YO(20),XE(10),YE(10),XT,YT
      DIMENSION PTDIST(20),IGOALX(20,10,20),IGOALY(20,10,20)
      DIMENSION PEDIST(20),IAGREE(20)
      ITEMP=IAGREE(NTHIS)
      IF ((XT.GT.0).AND.(YT.GT.0)) GO TO 1
      IF (ITEMP.GT.0) GO TO 2
      GO TO 6
1     TDIST=SQRT(FLOAT((XO(NTHIS)-XT)**2+
     1   (YO(NTHIS)-YT)**2))
      IF (ITIME.GT.1) GO TO 50
      PTDIST(NTHIS)=TDIST
50    TCHANG=TDIST-PTDIST(NTHIS)
      IF (ITEMP.GT.0) GO TO 2
      IF (TCHANG.GE.0.) GO TO 5
      GO TO 6
2     EDIST=SQRT(FLOAT((XO(NTHIS)-IGOALX(IS,ITEMP,NTHIS))**2
     1   +(YO(NTHIS)-IGOALY(IS,ITEMP,NTHIS))**2))
      IF (ITIME.GT.1) GO TO 55
      PEDIST(NTHIS)=EDIST
55    ECHANG=EDIST-PEDIST(NTHIS)
      IF ((XT.GT.0).AND.(YT.GT.0)) GO TO 3
      IF (ECHANG.LE.0.) GO TO 5
      GO TO 6
3     IF ((TCHANG.GE.0.).AND.(ECHANG.LE.0.)) GO TO 5
      GO TO 6
5     IEVAL=1
      GO TO 7
6     IEVAL=0
7     PTDIST(NTHIS)=TDIST
      PEDIST(NTHIS)=EDIST
      RETURN
      END
```

```
      SUBROUTINE EVAL20 (MXTIME,MK,XO,YO,XE,YE,NTHIS,IAGREE,
     1   ITIME,C,IEVAL,TOTIME,XT,YT)
      INTEGER XO(20),YO(20),XE(10),YE(10),XT,YT,TOTIME
      DIMENSION IAGREE(20)
      ITEMP=IAGREE(NTHIS)
      IF (ITEMP.EQ.0) GO TO 5
      TEST=SQRT(FLOAT((XO(NTHIS)-XE(ITEMP))**2+
     1   (YO(NTHIS)-YE(ITEMP))**2))
5     CONTINUE
      IF ((XT.GT.0).AND.(YT.GT.0)) GO TO 10
      IF (ITEMP.GT.0) GO TO 20
      GO TO 50
10    IF (ITEMP.GT.0) GO TO 20
      GO TO 30
C: ESCAPE-PENALTY:
20    IF (ITIME.LE.MXTIME) GO TO 21
      TIME=ITIME
      TTIME=TOTIME
      TDIST=TTIME-TIME
      GO TO 22
21    TDIST=TTIME-(FLOAT(MXTIME))
22    IF (TEST.LE.TDIST) GO TO 51
      GO TO 50
C: THREAT-EVASION PENALTY:
30    QTEST=SQRT(FLOAT((XO(NTHIS)-XT)**2+
     1   (YO(NTHIS)-YT)**2))
      IF (ITIME.LE.MXTIME) GO TO 31
      TIME=ITIME
      TDIST=TIME
      GO TO 32
31    TDIST=0.0
32    IF (QTEST.GE.TDIST) GO TO 51
50    IEVAL=0
      RETURN
51    IEVAL=1
      RETURN
      END
      SUBROUTINE INTRPT (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1   XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,
     2   IRAND,P,MOVE,XK,YK,K,L,IS,IGOALX,IGOALY,IENTER,
     3   X,INTLIM,INTR,INTNUM,PI2,PIO)
      DIMENSION INTNUM(20),INTLIM(20)
      DIMENSION INTR(20)
      IF (INTR(NTHIS).EQ.1) GO TO 1
      IF (INTNUM(NTHIS).LE.INTLIM(NTHIS)) GO TO 2
      INT=0
      RETURN
    1 INT=2
      RETURN
    2 X=RANDNU(0)
      IF (X) 5,6,6
    5 X=X*(-1)
    6 CONTINUE
      IF (X.LT.PI2) GO TO 3
      IF (X.LT.PIO) GO TO 4
      INT=1
      INTNUM(NTHIS)=INTNUM(NTHIS)+1
      RETURN
    3 INTNUM(NTHIS)=INTNUM(NTHIS)+1
      INTR(NTHIS)=1
      INT=2
      RETURN
    4 INT=0
      RETURN
      END
```

```
      SUBROUTINE BACKUP (IBACK,XO,YO,INITXO,INITYO,XOB,YOB,
     1   ITIME,NTHIS,NEWXO,NEWYO,INTR,JTIME)
      DIMENSION IBACK(20),INITXO(20),
     1   INITYO(20),JTIME(20),INTR(20)
      INTEGER XOB(20,100),YOB(20,100),XO(20),YO(20)
      IF (IBACK(NTHIS).EQ.0) GO TO 1
      IF ((XO(NTHIS).EQ.INITXO(NTHIS)).AND.
     1   (YO(NTHIS).EQ.INITYO(NTHIS))) GO TO 3
    1 IBACK(NTHIS)=IBACK(NTHIS)+1
      IF (IBACK(NTHIS).EQ.1) GO TO 2
      JTIME(NTHIS)=JTIME(NTHIS)-1
      GO TO 4
    2 IF (ITIME.GT.1) GO TO 21
      JTIME(NTHIS)=1
      GO TO 4
   21  JTIME(NTHIS)=ITIME-1
    4 KTIME=JTIME(NTHIS)
      NEWXO=XOB(NTHIS,KTIME)
      NEWYO=YOB(NTHIS,KTIME)
      RETURN
    3 INTR(NTHIS)=0
      IBACK(NTHIS)=0
      RETURN
      END
      SUBROUTINE NEWXY (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1   XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,
     2   IRAND,P,MOVE,XK,YK,K,NEWXO,NEWYO)
      INTEGER XK,YK
      CALL KTOXY (ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1   XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,
     2   IRAND,P,MOVE,XK,YK,K)
      NEWXO=XK
      NEWYO=YK
      RETURN
      END
```

```
      SUBROUTINE UPDATE (XO,YO,NTHIS,NEWXO,NEWYO)
      INTEGER XO(20),YO(20)
      XO(NTHIS)=NEWXO
      YO(NTHIS)=NEWYO
      RETURN
      END
```

```
C
C
C
C
C
C    SUBROUTINE XSCORE
C
C    THE PURPOSE OF XSCORE IS TO COMPUTE EACH OCCUPANT'S ESCAPE SCORE FOR THE
C    ENTIRE EVENT, I.E., SCORE=(TOTAL TIME AVAILABLE - CURRENT TIME FRAME)
C    DIVIDED BY THE TOTAL TIME AVAILABLE
C
      SUBROUTINE XSCORE (TOTIME,ISCORE,NUMOCC,SCORE)
      DIMENSION ISCORE(20),SCORE(20)
      INTEGER TOTIME
      SUMTIM=TOTIME
      DO 10 NTHIS=1,NUMOCC
      E=ISCORE(NTHIS)
C    COMPUTE THE OCCUPANT'S ESCAPE SCORE FOR THE ENTIRE EVENT
10    SCORE(NTHIS)=(SUMTIM-E)/SUMTIM
      RETURN
      END
```

```
C
C
C
C
C
C    SUBROUTINE STEPS
C
C    THE PURPOSE OF STEPS IS TO KEEP TRACK OF THE TOTAL NUMBER OF SPATIAL
C    RELOCATIONS ('STEPS') ACTUALLY MADE BY EACH OCCUPANT DURING THE FIRE EVENT
C    ...A STEP IS DEFINED AS A SPATIAL RELOCATION, AND THEREFORE, REMAINING-IN-
C    PLACE IS NOT RECORDED AS A STEP
C
      SUBROUTINE STEPS (XPRIOR,YPRIOR,XO,YO,NUMSTP,NTHIS)
      INTEGER XPRIOR(20),YPRIOR(20),XO(20),YO(20)
      DIMENSION NUMSTP(20)
C    RECORD WHETHER THE OCCUPANT MOVED DURING THE TIME FRAME JUST ENDED, THAT IS,
C    WHETHER (XPRIOR.NE.XO).AND.(YPRIOR.NE.YO)
      IF ((XPRIOR(NTHIS).EQ.XO(NTHIS)).AND.(YPRIOR(NTHIS).EQ.YO(NTHIS)))
     1  GO TO 999
C    UPDATE THE TOTAL NUMBER OF STEPS TAKEN BY THE OCCUPANT DURING THE EVENT
      NUMSTP(NTHIS)=NUMSTP(NTHIS)+1
999   RETURN
      END
```

```
C
C
C
C
C
C   SUBROUTINE PASSG
C
C   THE PURPOSE OF PASSG IS TO KEEP TRACK OF DOOR-PASSAGE BEHAVIOR EXHIBITED BY
C   OCCUPANTS
C
      SUBROUTINE PASSG (IDPASS,IDOOR,XO,YO,NTHIS,
     1   ND,NEWXO,NEWYO)
      DIMENSION IDPASS(20),IDOOR(30,4)
      INTEGER XO(20),YO(20)
C   DETERMINE WHETHER THE OCCUPANT PASSED THROUGH A DOORWAY DURING THE TIME-
C   FRAME JUST ENDED
      IXPASS=(NEWXO+XO(NTHIS))/2
      IYPASS=(NEWYO+YO(NTHIS))/2
      DO 10 I=1,ND
      IF ((IDOOR(I,1).EQ.IXPASS).AND.
     1   (IDOOR(I,2).EQ.IYPASS)) GO TO 5
      GO TO 10
C   UPDATE THE TOTAL NUMBER OF DOOR PASSAGES MADE BY THE OCCUPANT DURING THE
C   FIRE EVENT
5     IDPASS(NTHIS)=IDPASS(NTHIS)+1
10    CONTINUE
      RETURN
      END
```

```
C
C
C
C
C
C   SUBROUTINE REPORT
C
C   THE PURPOSE OF REPORT IS TO PRINT-OUT A SUMMARY TABLE FOR EACH TIME-FRAME
C   OF A GIVEN REPLICATION...EACH TABLE REPORTS RESULTS OF THE DECISION
C   PROCESS FOR EACH OCCUPANT IN THE TIME FRAME, PARTICULARLY MOVE PROBABILITY
C   VALUES ASSIGNED TO EACH AVAILABLE MOVE ALTERNATIVE AT THAT POINT IN TIME
C   ...X,Y COORDINATES ARE ALSO SHOWN FOR BOTH TIMES T AND T+1, FOR EACH
C   OCCUPANT IN THE TIME FRAME
C
      SUBROUTINE REPORT(ITIME,NTHIS,IHANDI,INT,IBYSTD,IEVAL,
     1 XO,YO,IBAR,TOTBAR,XT,YT,NAGREE,XE,YE,IAGREE,
     2 IRAND,P,MOVE,XK,YK,K,NUMEXT,NUMOCC,TOTIME,INTLIM,
     3 KNOWAY,PTDIST,TDIST,PEDIST,EDIST,NEWXO,NEWYO,
     4  EVLOPT,IDOOR,IDOPEN,ND,INTR)
      DIMENSION IBAR (20,75,2),INTLIM(20),IHANDI
     1  (20),KNOWAY(20),IBYSTD(20),P(9),INTR(20)
      DIMENSION IDOOR(30,4),IDOPEN(30,100),IAGREE(20)
      INTEGER XE(10),YE(10),XO(20),YO(20),XT,YT,TOTIME,
     1  XK,YK,EVLOPT
C
      IF ((ITIME.GT.1).OR.(NTHIS.GT.1)) GO TO 1
C
C: ECHO:CHECK INPUT PARAMETERS*
C:  (1) ENVIRONMENTAL:
C
      WRITE (6,100)
      WRITE (6,101) XT,YT
      WRITE (6,102) NUMEXT
      WRITE (6,104)
      WRITE (6,105) (XE(I),I=1,NUMEXT)
      WRITE (6,106) (YE(I),I=1,NUMEXT)
C
C:  (2) SYSTEM:
C
      WRITE (6,109) NUMOCC
      WRITE (6,110) TOTIME
      WRITE (6,111) IRAND
C
C:  (3) OCCUPANT:
C
      WRITE (6,112)
      WRITE (6,113) (INTLIM(I),I=1,NUMOCC)
      WRITE (6,115) (IHANDI(I),I=1,NUMOCC)
      WRITE (6,116) (KNOWAY(I),I=1,NUMOCC)
      WRITE (6,117)
C
1     CONTINUE
C
      IF (NTHIS.NE.1) GO TO 2
C
C: IF NTHIS=1, PRINT CURRENT TIME MARKER AND COLUMN HEADINGS:
C
      WRITE (6,118)
      WRITE (6,119) ITIME
      WRITE (6,118)
      WRITE (6,120)
      WRITE (6,118)
C
C
2     IF (INTR(NTHIS).EQ.1) GO TO 22
C
C: WRITE OUTPUT MATRIX:
      IF (EVLOPT-1) 20,20,21
 20   CONTINUE
      WRITE (6,121) NTHIS,XO(NTHIS),YO(NTHIS),INT,IBYSTD(NTHIS),IAGREE
     1 (NTHIS),TDIST,EDIST,(P(K),K=1,9),NEWXO,NEWYO
      GO TO 23
```

```
21    WRITE (6,123) NTHIS,XO(NTHIS),YO(NTHIS),INT,IBYSTD(NTHIS),
     1   IAGREE(NTHIS),(P(K),K=1,9),NEWXO,NEWYO
      GO TO 23
22    WRITE (6,127) NTHIS,XO(NTHIS),YO(NTHIS),INT,
     1   NEWXO,NEWYO
23    CONTINUE
      IF (NTHIS.EQ.NUMOCC) GO TO 3
      GO TO 4
3     WRITE (6,118)
4     IF ((ITIME.EQ.TOTIME).AND.(NTHIS.EQ.NUMOCC)) GO TO 5
      GO TO 6
5     WRITE (6,118)
      WRITE (6,124)
      WRITE (6,118)
      WRITE (6,125)
      WRITE (6,118)
      DO 30 I=1,ND
      WRITE (6,126) I,IDOOR(I,1),IDOOR(I,2),IDOOR(I,3),
     1   (IDOPEN(I,ITM),ITM=1,TOTIME)
30    CONTINUE
      WRITE (6,118)
      WRITE (6,122)
      WRITE (6,118)
      GO TO 6
C
C: OUTPUT FORMATING:
C
100   FORMAT (1X,120('*'),//,55X,'ECHO-CHECK INPUT PARAMETERS',//,120
     1 ('*'),//,1X,'(1) ENVIRONMENTAL:',/)
101   FORMAT (24X,'THREATENED EXIT: X= ',I2,4X,'Y=',I2)
102   FORMAT (24X,'NUMBER OF EXITS:  = ',I2)
103   FORMAT (24X,'NO. OF BARRIER PTS=',I3,/)
104   FORMAT (24X,'COORDINATES OF EXITS:  1  2  3  4  5  6  7  8  9 10',
     1 /)
105   FORMAT (43X,'X: ',10(I2,1X))
106   FORMAT (43X,'Y: ',10(I2,1X),/)
107   FORMAT (1X,'BARRIER-POINT MATRIX:',/)
108 . FORMAT (2X,'X:',38(I2,1X),/,2X,'Y:',38(I2,1X),/)
109   FORMAT (1X,'(2) SYSTEM',/,24X,'NUMBER OF OCCUPANTS IN THE SPACE ='
     1 ,I3)
110   FORMAT (24X,'TOTAL NO. OF TIME INCREMENTS =',I3)
111   FORMAT   (24X,'RANDOM NUMBER STARTER         =',I3,/)
112   FORMAT (1X,'(3) OCCUPANT:',/,12X,'PARAMETER',5X,'OCC NO  1  2  3
     1 4  5  6  7  8  9 10 11 12 13 1 4 15 16 17 18 19 20',/)
113   FORMAT (12X,'INTLIM',15X,20(I2,1X))
115   FORMAT (12X,'IHANDI',15X,20(I2,1X))
116   FORMAT (12X,'KNOWAY',15X,20(I2,1X),/)
117   FORMAT (2(1X,120('*'),/))
118   FORMAT (1X,120('*'),/)
119   FORMAT (1X,'TIME = ',I3,/)
120   FORMAT (6X,'PRIOR',18X,'EXIT',89X,'NEW',/,1X,'OCC',2X,'LOCAT',17X,
     1'AGREED',87X,'LOCAT',/,1X,'NUM',2X,'XO YO  INT    IBYSTD   UPON
     2  PTDIST TDIST   PEDIST EDIST   P(1)  P(2)  P(3)  P(4)  P(5)  P(6)
     3 P(7)  P(8)  P(9)',6X,'XO YO',/)
121   FORMAT (1X,I2,3X,I2,1X,I2,4X,I1,6X,I1,6X,I2,3X,2(7X,F6.3,2X),
     1 9(F5.3,1X),4X,I2,1X,I2)
122   FORMAT (50X,'END OF SIMULATION',/)
123   FORMAT (1X,I2,3X,I2,1X,I2,4X,I1,6X,I1,6X,
     1   I2,33X,9(F5.3,1X),4X,I2,1X,I2)
124   FORMAT (50X,'DOOR STATUS SUMMARY',/)
125   FORMAT (1X,'DOOR',4X,'X  Y',5X,'TYPE',5X,'T= 1  2  3  4  5
     1  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
     2  24 25 26 27 28 29 30',/)
126   FORMAT (2X,I2,5X,I2,1X,I2,7X,I1,8X,30(I2,1X))
127   FORMAT (1X,I2,3X,I2,1X,I2,4X,I1,106X,I2,1X,I2)
6     RETURN
      END
```

```
C
C
C
C
C
C   SUBROUTINE SUMMRY
C
C   THE PURPOSE OF SUMMRY IS TO PRINT-OUT A SUMMARY TABLE FOR EACH REPLICATION
C   RUN...OUTPUT VALUES ARE 'GRAND MEANS' TAKEN ACROSS OCCUPANTS AND TIME-
C   FRAMES, AND REPORT AGGREGATED SCORES AND FINAL STATUS AT THE END OF THE
C   LAST TIME FRAME FOR EACH REPLICATION
C
      SUBROUTINE SUMMRY (INITXO,INITYO,INTLIM,LBYSTD,IHANDI,KNOWAY,
     1   POPEN,PCLOSE,SCORE,NUMSTP,IPS,III,NUMREP,NUMOCC,TOTIME,
     2   XO,YO,XE,YE,TITLE,IXHELP,IYHELP,TSMOKE,PHELP)
      DIMENSION IXHELP(20),IYHELP(20),PHELP(20)
      DIMENSION INITXO(20),INITYO(20),INTLIM(20),LBYSTD(20),IHANDI(20),
     1  KNOWAY(20),POPEN(20),PCLOSE(20),SCORE(20),NUMSTP(20),IPS(20)
      DIMENSION NUMBER(20),SMEAN(20),DMEAN(20),DIST(20),DEINIT(20),
     1   DIFF(20),DFMEAN(20),TITLE(20)
      REAL NMEAN(20),IMEAN(20)
      INTEGER TOTIME,XO(20),YO(20),XE(10),YE(10),TSMOKE(20)
      DIFTOT=0.0
      DTOTAL=0.0
      STOTAL=0.0
      NTOTAL=0
      ITOTAL=0
      NSUM=NUMOCC*NUMREP
      SUM=NSUM
      AN=NUMOCC
      WRITE (6,100)
      WRITE (6,134) TITLE
      WRITE (6,119)
      WRITE (6,102)
      WRITE (6,103)
      WRITE (6,119)
      WRITE (6,104) III,NUMREP,TOTIME
      WRITE (6,102)
      WRITE (6,119)
      WRITE (6,105)
      WRITE (6,119)
      WRITE (6,106)
      DO 200 N=1,20
      NUMBER(N)=N
200   CONTINUE
      WRITE (6,107) (NUMBER(N),N=1,20)
      WRITE (6,101)
      DO 55 K=1,NUMOCC
55    DEINIT(K)=SQRT(((INITXO(K)-XE(1))**2)+((INITYO(K)-YE(1))**2))
      WRITE (6,131) (DEINIT(K),K=1,NUMOCC)
      WRITE (6,108) (INITXO(K),K=1,NUMOCC)
      WRITE (6,109) (INITYO(K),K=1,NUMOCC)
      WRITE (6,135)(IXHELP(K),K=1,NUMOCC)
      WRITE (6,136)(IYHELP(K),K=1,NUMOCC)
      WRITE (6,110) (INTLIM(K),K=1,NUMOCC)
      WRITE (6,111) (LBYSTD(K),K=1,NUMOCC)
      WRITE (6,112) (IHANDI(K),K=1,NUMOCC)
      WRITE (6,113) (KNOWAY(K),K=1,NUMOCC)
      WRITE (6,137)(TSMOKE(K),K=1,NUMOCC)
      WRITE (6,114) (POPEN(K),K=1,NUMOCC)
      WRITE (6,115) (PCLOSE(K),K=1,NUMOCC)
      WRITE (6,138)(PHELP(K),K=1,NUMOCC)
      WRITE (6,102)
      WRITE (6,119)
      WRITE (6,116)
      WRITE (6,119)
      WRITE (6,106)
      WRITE (6,107) (NUMBER(N),N=1,20)
      WRITE (6,101)
C..COMPUTE REPLICATION MEANS...
      DO 20 K=1,NUMOCC
      DIST(K)=SQRT(((XE(1)-XO(K)) **2)+((YE(1)-YO(K))**2))
```

```
         DTOTAL=DTOTAL+DIST(K)
         STOTAL=STOTAL+SCORE(K)
         NTOTAL=NTOTAL+NUMSTP(K)
 20      ITOTAL=ITOTAL+IPS(K)
         SMEAN(III)=STOTAL/AN
         NMEAN(III)=FLOAT(NTOTAL)/AN
         IMEAN(III)=FLOAT(ITOTAL)/AN
         DMEAN(III)=DTOTAL/AN
C
         DO 201 K=1,NUMOCC
         DIFF(K)=DEINIT(K)-DIST(K)
         DIFTOT=DIFTOT+DIFF(K)
 201     NUMSTP(K)=NUMSTP(K)+1
         WRITE (6,127) (XO(K),K=1,NUMOCC)
         WRITE (6,128) (YO(K),K=1,NUMOCC)
         WRITE (6,129) (DIST(K),K=1,NUMOCC)
         WRITE (6,132) (DIFF(K),K=1,NUMOCC)
         WRITE (6,117) (SCORE(K),K=1,NUMOCC)
         WRITE (6,118) (NUMSTP(K),K=1,NUMOCC)
         WRITE (6,120) (IPS(K),K=1,NUMOCC)
C. COMPUTE NESCAP...
         NESCAP=NUMOCC
         DO 300 L=1,NUMOCC
         IF (SCORE(L).LE.0.001) GO TO 301
         GO TO 300
 301     NESCAP=NESCAP-1
 300     CONTINUE
         WRITE (6,126) NESCAP
         WRITE (6,102)
         WRITE (6,122) NUMOCC
         WRITE (6,130) DMEAN(III)
         DFMEAN(III)=DIFTOT/AN
         WRITE (6,133) DFMEAN(III)
         WRITE (6,123) SMEAN(III)
         WRITE (6,124) NMEAN(III)
         WRITE (6,125) IMEAN(III)
         WRITE (6,101)
 100     FORMAT ('1')
 101     FORMAT (130('*'))
 102     FORMAT (130('*'),/,130('*'))
 103     FORMAT (57X,'SIMULATION SUMMARY')
 104     FORMAT (3X,'REPLICATION',I5,'   OF',I5,70X,'RUN FOR',I4,
        1   ' TIME FRAMES')
 105     FORMAT (57X,'INITIAL CONDITIONS')
 106     FORMAT (63X,'OCCUPANT NUMBER')
 107     FORMAT (11X,20(I2,3X))
 108     FORMAT (3X,'INITXO',1X,29(I3,2X),I3)
 109     FORMAT (3X,'INITYO',1X,29(I3,2X),I3)
 110     FORMAT (3X,'INTLIM',1X,29(I3,2X),I3)
 111     FORMAT (3X,'LBYSTD',1X,29(I3,2X),I3)
 112     FORMAT (3X,'IHANDI',1X,29(I3,2X),I3)
 113     FORMAT (3X,'KNOWAY',1X,29(I3,2X),I3)
 114     FORMAT (3X,'POPEN ',1X,29(F3.2,2X),F3.2)
 115     FORMAT (3X,'PCLOSE',1X,29(F3.2,2X),F3.2)
 116     FORMAT (58X,'OUTCOMES')
 117     FORMAT (3X,'SCORE ',1X,20(F4.2,1X))
 118     FORMAT (3X,'STEPS ',1X,20(I3,2X))
 120     FORMAT (3X,'PASSES',1X,20(I3,2X))
 119     FORMAT (' ')
 121     FORMAT (57X,'END OF REPLICATION')
 122     FORMAT (9X,'MEANS COMPUTED ACROSS',I3,' OCCUPANTS...')
 123     FORMAT (20X,'MEAN SCORE = ',F4.2)
 124     FORMAT (20X,'MEAN STEPS = ',F6.2)
 125     FORMAT (20X,'MEAN PASSES =',F6.2)
 126     FORMAT (3X,'NUMBER ESCAPED',5X,I3)
 127     FORMAT (3X,'FIN X ',1X,20(I3,2X))
 128     FORMAT (3X,'FIN Y ',1X,20(I3,2X))
 129     FORMAT (3X,'DIST  ',1X,20(F4.1,1X))
 130     FORMAT (20X,'MEAN DIST  = ',F4.1)
 131     FORMAT (3X,'INIT D',1X,20(F4.1,1X))
 132     FORMAT (3X,'DIFF D',   20(F5.1))
 133  FORMAT (20X,'MEAN DIFF D =',F4.1)
 134  FORMAT (1X,'RUN TITLE/  ',20A4)
 135  FORMAT (3X,'IXHELP',1X,29(I3,2X),I3)
 136  FORMAT (3X,'IYHELP',1X,29(I3,2X),I3)
 137  FORMAT (3X,'TSMOKE',1X,29(I3,2X),I3)
 138  FORMAT (3X,'PHELP',2X,29(F3.2,2X)F3.2)
      RETURN
      END
```

```
C
C
C
C
C
C     SUBROUTINE TRACE
C
C     THE PURPOSE OF TRACE IS TO PRINT-OUT A SUMMARY TABLE FOR EACH OCCUPANT IN
C     A GIVEN REPLICATION...EACH SUCH TABLE TRACES THE SPATIAL DISPLACEMENT
C     (CHANGE IN X,Y COORDINATES) OF EACH OCCUPANT ACROSS THE ENTIRE SIMULATED
C     EVENT
C
        SUBROUTINE TRACE (IXTRCE,IYTRCE,NTHIS,ITIME,NUMOCC,
       1   TOTIME)
        DIMENSION IXTRCE(20,100),IYTRCE(20,100)
        INTEGER TOTIME
        WRITE (6,1)
        DO 25 I=1,NUMOCC
        WRITE (6,2) I
        DO 24 J=1,TOTIME
        WRITE (6,3) J,IXTRCE(I,J),IYTRCE(I,J)
   24   CONTINUE
        WRITE (6,4)
   25   CONTINUE
        WRITE (6,5)
    1   FORMAT (1X,120('*'),//,48X,
       1   'OCCUPANT MOVEMENT TRACES',//,1X,120('*'))
    2   FORMAT (1X,120('*'),//,1X,'OCCUPANT NUMBER:',I4,/,
       1   10X,'TIME',/,10X,'FRAME',10X,'X',5X,'Y',/,
       2   1X,120('*'))
    3   FORMAT (12X,I2,10X,I2,4X,I2)
    4   FORMAT (1X,120('*'))
    5   FORMAT (1X,120('*'),//,55X,'END OF TRACES',//,
       1   1X,120('*'))
        RETURN
        END
```

```
C
C
C
C
C  SUBROUTINE PLOT
C
C  THIS PROGRAM PLOTS SMOKE AND OCCUPANT LOCATIONS FOR A
C  SINGLE TIME FRAME.  ONLY PERSON-OCCUPIABLE LOCATIONS
C  ARE SHOWN.
C
      SUBROUTINE PLOT (XO,YO,NUMOCC,ITIME,MAXX,MAXY,ISMOKE)
      INTEGER XO(20),YO(20)
      DIMENSION ISMOKE(100,100),IPLOT(100,100)
C  BUILD THE PLOT MATRIX
      DO 100 J=2,MAXY,2
      DO 100 I=2,MAXX,2
      IF (ISMOKE(I,J).GT.0) GO TO 110
      IPLOT(I,J)=0
      GO TO 120
110   IPLOT(I,J)=1
120   CONTINUE
C  DETERMINE LOCATIONS OF OCCUPANTS WITHIN THE PLOT MATRIX
      DO 100 N=1,NUMOCC
      IF (XO(N).EQ.I.AND.YO(N).EQ.J) GO TO 130
      GO TO 100
130   IPLOT(I,J)=8
100   CONTINUE
C  PRINT THE PLOT MATRIX
      WRITE (6,500) ITIME
      DO 300 J=2,MAXY,2
      WRITE (6,501) (IPLOT(I,J),I=2,MAXX,2)
      WRITE (6,502)
300   CONTINUE
500   FORMAT ('1',1X,'PLOT FOR TIME FRAME NUMBER',I4,/,130('-'),///)
501   FORMAT (1X,60(2X,I2))
502   FORMAT (1X,/)
      RETURN
      END
```

FEDERAL INFORMATION PROCESSING STANDARD SOFTWARE SUMMARY

01. Summary date

Yr.	Mo.	Day
80	02	20

02. Summary prepared by (Name and Phone)

03. Summary action
New □ Replacement ☒ Deletion □
Previous Internal Software ID

04. Software date

Yr.	Mo.	Day
79	12	01

05. Software title
Computer Simulation of Human Behavior in Fires: Version 2

06. Short title
BFIRESII

07. Internal Software ID

08. Software type
□ Automated Data System
☒ Computer Program
□ Subroutine/Module

09. Processing mode
□ Interactive
☒ Batch
□ Combination

10. Application area
General
□ Computer Systems Support/Utility
☒ Scientific/Engineering
□ Bibliographic/Textual
□ Management/Business
□ Process Control
□ Other

Specific
Computer simulation of human behavior.

11. Submitting organization and address
Environmental Design Research Division
Center for Building Technology
National Engineering Laboratory
National Bureau of Standards
Washington, DC 20234

12. Technical contact(s) and phone
Dr. Fred I. Stahl
(301) 921-2627

13. Narrative

BFIRESII aids the prediction of escape times and escape routes of building occupants during fire emergencies. The program executes a discrete time stochastic simulation of human movement within bounded physical spaces, in response to specified fire conditions. Unlike its predecessor (BFIRES), BFIRESII ("BFIRES-two") permits the simulation of behavioral consequences of occupying a smoke-filled environment, as well as rescue activities. Input/output is via punched cards and line printer, or via terminal. in batch mode.

14. Keywords
Building fires; computer-assisted building design; fire research; fire safety; human performance simulation.

15. Computer manuf'r and model
UNIVAC 1108

16. Computer operating system
EXEC 8 Ver. 33R3A

17. Programing language(s)
FORTRAN V

18. Number of source program statements

19. Computer memory requirements
63,500 36-bit words

20. Tape drives

21. Disk/Drum units

22. Terminals

23. Other operational requirements

24. Software availability
Available □ Limited ☒ In-house only □

Dr. Fred I. Stahl (301) 921-2627
Research Psychologist
BR A355
National Bureau of Standards
Washington, DC 20234

25. Documentation availability
Available ☒ Inadequate □ In-house only □

same as item 24

26. FOR SUBMITTING ORGANIZATION USE

NBS-114A (REV. 9-78)

U.S. DEPT. OF COMM.
BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NO.
NBSIR 80-1982

2. Gov't. Accession No.

3. Recipient's Accession No.

4. TITLE AND SUBTITLE
BFIRES/Version 2: Documentation of Program Modifications

5. Publication Date
MARCH 1980

6. Performing Organization Code

7. AUTHOR(S)
Fred I. Stahl

8. Performing Organ. Report No.

9. PERFORMING ORGANIZATION NAME AND ADDRESS

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, DC 20234

10. Project/Task/Work Unit No.
7431117

11. Contract/Grant No.

12. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS *(Street, City, State, ZIP)*
same as item 9

13. Type of Report & Period Covered

14. Sponsoring Agency Code

15. SUPPLEMENTARY NOTES
Companion document to NBSIR 79-1713

☒ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

16. ABSTRACT *(A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here.)*

Several shortcomings of BFIRES/Version 1 are discussed. Chief among these are the program's inability to simulate rescue activities during fire events, and to simulate direct interactions between occupant behavior and toxic qualities of smoke filled environments. This report documents a revised program, BFIRES/Version 2, which contains new subroutines developed to mitigate these problems. These subroutines are grouped into two modules: (1) a "smoke" module designed to simulate the experience of inhabiting a smoke filled environment, and (2) a "rescue" module intended to permit the rescue of physically immobile occupants. Additional improvements incorporated into BFIRES/Version 2 include more efficient file management and data input facilities, and expanded output capabilities.

17. KEY WORDS *(six to twelve entries; alphabetical order; capitalize only the first letter of the first key word unless a proper name; separated by semicolons)*
Architectural research; building fires; computer-aided design; environmental psychology; fire research; fire safety; human behavior in fires; modeling technique; programming; simulation of human behavior.

18. AVAILABILITY ☒ Unlimited

□ For Official Distribution. Do Not Release to NTIS

□ Order From Sup. of Doc., U.S. Government Printing Office, Washington, DC 20402, SD Stock No. SN003-003-

☒ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

19. SECURITY CLASS (THIS REPORT)
UNCLASSIFIED

20. SECURITY CLASS (THIS PAGE)
UNCLASSIFIED

21. NO. OF PRINTED PAGES
110

22. Price
$9.00