NBS TECHNICAL NOTE **1176**

U.S. DEPARTMENT OF COMMERCE/National Bureau of Standards

Utility Programs for Generating the Hershey Character Fonts on Microcomputers and Laboratory Plotters

NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards[1] was established by an act of Congress on March 3, 1901. The Bureau's overall goal is to strengthen and advance the Nation's science and technology and facilitate their effective application for public benefit. To this end, the Bureau conducts research and provides: (1) a basis for the Nation's physical measurement system, (2) scientific and technological services for industry and government, (3) a technical basis for equity in trade, and (4) technical services to promote public safety. The Bureau's technical work is performed by the National Measurement Laboratory, the National Engineering Laboratory, and the Institute for Computer Sciences and Technology.

THE NATIONAL MEASUREMENT LABORATORY provides the national system of physical and chemical and materials measurement; coordinates the system with measurement systems of other nations and furnishes essential services leading to accurate and uniform physical and chemical measurement throughout the Nation's scientific community, industry, and commerce; conducts materials research leading to improved methods of measurement, standards, and data on the properties of materials needed by industry, commerce, educational institutions, and Government; provides advisory and research services to other Government agencies; develops, produces, and distributes Standard Reference Materials; and provides calibration services. The Laboratory consists of the following centers:

Absolute Physical Quantities[2] — Radiation Research — Chemical Physics — Analytical Chemistry — Materials Science

THE NATIONAL ENGINEERING LABORATORY provides technology and technical services to the public and private sectors to address national needs and to solve national problems; conducts research in engineering and applied science in support of these efforts; builds and maintains competence in the necessary disciplines required to carry out this research and technical service; develops engineering data and measurement capabilities; provides engineering measurement traceability services; develops test methods and proposes engineering standards and code changes; develops and proposes new engineering practices; and develops and improves mechanisms to transfer results of its research to the ultimate user. The Laboratory consists of the following centers:

Applied Mathematics — Electronics and Electrical Engineering[2] — Manufacturing Engineering — Building Technology — Fire Research — Chemical Engineering[2]

THE INSTITUTE FOR COMPUTER SCIENCES AND TECHNOLOGY conducts research and provides scientific and technical services to aid Federal agencies in the selection, acquisition, application, and use of computer technology to improve effectiveness and economy in Government operations in accordance with Public Law 89-306 (40 U.S.C. 759), relevant Executive Orders, and other directives; carries out this mission by managing the Federal Information Processing Standards Program, developing Federal ADP standards guidelines, and managing Federal participation in ADP voluntary standardization activities; provides scientific and technological advisory services and assistance to Federal agencies; and provides the technical foundation for computer-related policies of the Federal Government. The Institute consists of the following centers:

Programming Science and Technology — Computer Systems Engineering.

[1] Headquarters and Laboratories at Gaithersburg, MD, unless otherwise noted; mailing address Washington, DC 20234.
[2] Some divisions within the center are located at Boulder, CO 80303.

Utility Programs for Generating the Hershey Character Fonts on Microcomputers and Laboratory Plotters

C. E. Dick
J. Hilsenrath

National Measurement Laboratory
National Bureau of Standards
Washington, DC 20234

ABSTRACT

Two programs are described that allow for the storage and manipula-
tion of digitized fonts of graphic arts symbols and characters on
the Apple II computer system. These fonts are based on the work of
A. V. Hershey and provide the user with a repertory of digitized
characters suitable for use in the preparation of camera-ready
illustration in the laboratory environment. The programs described
contain routines for reading files of the Hershey coordinates,
storing them as text files, displaying them as individual charac-
ters or combined text on the high resolution display, and two
methods for editing them or creating special symbols and graphics.

Key words: Applesoft basic programs; camera-ready illustrations;
digitized graphic symbols; Hershey character fonts; microcomputers.

Table of Contents

List of Figures

Utility Programs for Generating the Hershey Character Fonts
on
Microcomputers and Laboratory Plotters

INTRODUCTION

With the advent of microcomputers able to drive inexpensive
x-y plotters, it is now possible for scientists to produce camera
ready illustrations in their own laboratory. Such drawings have,
in the past, required costly and time consuming drafting and hand
lettering by graphics artists.

The ingredients of such an in-house facility, in addition to
the microcomputer and plotter, are:

     a) a collection of mechanical drafting routines

     b) a repertory of graphic arts fonts in digitized form

     c) computer programs to store, retrieve, edit, and display
        them on video terminals, printers, and plotters.

Mechanical drafting and data handling routines have been
under development for some time and are the subject of a
forthcoming report. For the character digitization, we have been
able to draw on the pioneering and definitive work of Dr. Allen V.
Hershey as subsequently presented by Wolcott and Hilsenrath. The
third item on the above list, the software required to digitize
and display the characters, is the subject of this report. This
report describes two programs for the Apple II computer[*] operating
under the disk operating system (DOS 3.3). The first program,
which is treated in detail, contains routines for reading files of
Hershey coordinates and storing them in a more compact format as
text files, displaying individual characters or complete fonts on
the hires (high resolution) screen, and two methods for editing or

[*] In this report certain commercial products are referenced by
    name. These references are for informational purposes only
    and do not imply that they are necessarily the best
    available for the purpose or do not imply endorsement by NBS.

    C.E. Dick and J.Hilsenrath

making new characters. This program is the main subject of this report. The second program, which is treated as an appendix to this paper, details utilities for combining individual Hershey characters on the hires screen for the creation of logos for use in other programs. It is meant to be illustrative of some of the possible applications of the Hershey fonts.

THE ORIGIN AND DIFFUSION OF THE HERSHEY FONTS

In the mid sixties, Dr. Allen V. Hershey undertook a project at the Naval Weapons Laboratory in Dahlgren, Virginia, to automate the preparation of his highly mathematical reports and those of others involving complex chemical structures and electrical diagrams. That work produced a comprehensive suite of FORTRAN typographic programs; a repertory of occidental alphabets and oriental ideograms, cartographic symbols, etc. The typographic system has been fully documented in a series of NWL reports (1-5). Early applications of the system are to be found in Hershey's own scientific publications (6-10) in which the system was honed.

More than 120 copies of the original system have been distributed (largely in card decks) to universities, industries, and government laboratories. Applications range from computer assisted typesetting to computer driven engraving machines with the output characters appearing on all types of printers and plotters (11). Although Hershey's characters have been incorporated in many commercial graphics packages, unfortunately, they do not all acknowledge the source. NBS has used the system for well over ten years, and in 1976 undertook the publication, in book form (12), of the occidental fonts, and the distribution on magnetic tape (13), of the coordinates for both the occidental and oriental characters. This was followed in 1978 by the publication of a subset incorporated in a Technical Note by N.W. Wolcott entitled "FORTRAN IV Enhanced Character Graphics" (14). Since then, the economic advantage of in-laboratory production of camera ready illustrations for research papers has led to the

incorporation of the Hershey character fonts as user defined characters under the Hewlett-Packard Graphics Language (15), and in a system for use on the Apple II computer system. It is this latter application which is the subject of this report.

Extensive use of the Hershey characters was made in 1981 by Claude A. Kagan who succeeded in adapting them for a variety of microcomputer systems, including the Apple II operating under CPM (16). A more recent application of the Hershey character set has been made by Daniel Macero and associates (17) who have modified a line printer to provide a resolution of 100 dots-per-inch horizontally and 72 dots-per-inch vertically.

THE CURRENT PROGRAMS

The present report details a technique for using the Hershey characters with the Apple II computer system operating under the Apple Disk Operating System, DOS 3.3 (18). Although the High Resolution Character Generator Routine supplied as part of the Applesoft Toolkit (19) provides for the creation and retrieval of characters and special symbols, these characters are limited to a 7x8 dot matrix format and are designed to be used on the high resolution graphics screen.

Consequently, we present in this report two programs that will allow the storage of the Hershey characters in a compact format as text files (CANDY APPLE) and their subsequent use as input to the hires graphics screen (CANDY WRAPPER). The first of these programs is described in some detail since it provides the means of storing the digitized character coordinates as well as a technique for creation of special characters and symbols. The second program is included as an appendix to illustrate a particular use of the Hershey characters. A more extensive program describing their use for digital plotting is currently being documented.

Although some effort has been expended to faithfully reproduce the character fonts tabulated by Wolcott and Hilsenrath,

we have made some reasonably liberal changes in these fonts to provide for our particular application.

THE HERSHEY CHARACTERS

As tabulated by Wolcott and Hilsenrath (12-13), the Hershey characters consist of a series of numbers which indicate the end points of straight line segments which are used to define alphanumeric and special symbols. Each succeeding number pair indicates the coordinates of the next dot to be connected in a connect the dots scheme. For example, the vectors that correspond to the capital A illustrated in Figure 1, are given in the following format:

```
    1: -5   5:  0   -5: -4   4:-64   0:  0  -5:  4   4:-64   0:
        -2   1:  2    1:-64 -64:
```

In this representation, the first number is the character number in the Hershey set, the first pair of numbers indicates the x coordinates of the limits of the character (-5,5), and the actual character begins at the next coordinate location (x=0,y=-5). It should be noted, in this context, that the coordinates are determined in a system where the x-axis is increasing from left to right, and the y-axis increases from top to bottom in accordance with standard printing notation where vertical deflections increase down the page. This is also the convention utilized by the Apple II for its graphics displays. The origin of the coordinate system is located approximately at the center of the character. Succeeding pairs of coordinates indicate the next point to be connected to draw the figure. A coordinate pair of -64 0 indicates that the next line is not to be drawn, but the stylus is to be moved to the next location. Finally, a coordinate pair of -64 -64 completes the figure.

For use on the Apple II computer system, we have chosen to store the characters as strings in a dimensioned array named X$. In order that the length of any particular string not exceed the maximum limit of 255 characters allowed, we have chosen to store

not the coordinates of the next point, but the offset required to arrive at that location. In addition, the character numbers have been deleted. In general, each string consists of a series of ASCII characters in the format:

W XY ? XY XY XY ? XY XY ? XY

where the first character, **W**, indicates the width of the character, the next pair, **XY**, indicates the offset to the start of the character, the character, **?**, indicates a change of pen status, etc. In order to obtain the maximum possible flexibility in the storage of the characters, each character stored is the ASCII character corresponding to the offset +93 (i.e. an offset of +1 is stored as CHR$(94)=^). In this scheme then, the string corresponding to the character, A, given by the Hershey vectors above is stored as:

'""?YT?!&?!t?WZ?!]?!^

where the vectors are now stored in a format where positive vertical displacements are in the upward direction. Note that this technique allows the character to be stored in 22 bytes whereas the previous representation would require at least 33 bytes even if the foregoing compression scheme were used to represent pen changes. In addition, the final vectors represent the offset necessary to advance the pen to the right hand character limit.

CANDY APPLE

    The program for manipulating these Hershey text files is given in Listing 1 and is written in Applesoft Basic (20). In general, the program is self-explanatory and menu-driven; it is broken up into the following sections:

Line numbers	Program function
10-199	Main menu and initialization; also contains the disk assignment routine.
1000-1460	Hershey character input routine using the tabulated Hershey vectors.
2000-2130	Write font to disk as a text file.
3000-3260	Routines to modify a font.

C.E. Dick and J.Hilsenrath

```
4000-4070              Font hires display index routine.
5000-5460              Font edit routines; also contains cursor
                       input routines.
6000-6180              Hires routine to display font.
7000-7110              Read font from disk.
8000-8090              Display single character as it is input.
9000-9010              Catalog input disk.
10000-10000            Exit routine.
11000-11060            Hires screen swap routines.
```

In the following sections, each segment of the program will be examined in more detail. Although some effort was made to write the most efficient code, the program was designed to be user friendly and hopefully to contain as small a number of bugs as possible. In Listing 2, the variable documentation program of South Western Data Systems (21) was used to document the usage of all the variables used in the program. Reference to this listing will be useful in the following discussions.

Lines 10-999 : THE MAIN MENU AND INITIALIZATION ROUTINES

This section of the program is used to display the main menu and to determine the disk drives that will be used for both input and output of the data. Before running the main body of the program however, the entire program is relocated so that the computer stores the Applesoft code beginning at memory location 24577 ($6001) using the &LOMEM: utility of Konzen (22). This utility shifts the entire Applesoft program and variable space above the locations that are needed for both pages of the hires graphics. After this shift, the program initializes the font array (line 50), pokes a hires shape table into locations 768-777 ($0300-$0309) and loads the appropriate hooks to the table on page 0. This shape is used as a cursor for the input of character vectors in lines 5000-5460. Lines 80-90 input the appropriate disk drive information and the rest of the section displays the main menu. The main menu is self-explanatory for the most part, and contains options that allow the user to CREATE, MODIFY, EDIT, or DISPLAY a font as well as perform certain disk operations.

This section contains the routines to input a character using the vectors listed by Wolcott and Hilsenrath. Initially, however, the user is given the choice to input these vectors or to enter the character using the character edit routines of the 5000 section. The directions given for the Hershey coordinate input indicate the procedure to follow to enter the data. Line 1100 is used to eliminate the initialization of the character number, L, if the font is being edited or modified. Line 1120 branches to the subroutine that draws the hires graphics that illustrates the character as it is input. The appropriate x- and y-offsets are entered using the GET command of Applesoft to input a character string. Inclusion in the string of the left arrow requires the re-entry of the entire string. Entering a space for the horizontal input signals a change of pen position for the following coordinate move only (the equivalent of :-64   0: in the Wolcott notation). Entering the symbol, /, for the horizontal coordinate signals the end of the character (:-64 -64:), and enters the appropriate vectors to advance the pen to the location of the right marker. The first two vectors input indicate the left and right markers respectively, and are used to determine the character width, W, in line 1360. The previous x and y coordinates are stored in the variables XT and YT and are used in line 1370 to determine the current x and y deflections. Line 1400 generates the string by concatenating the current values to the string X$(L). The rest of this section increments the character number, L, and continues the input process. These routines are terminated in one of two fashions; if the horizontal coordinate is entered as E, then the current string is set equal to "EOF", and the user may write the font to the disk or return to the main menu (lines 1230-1250); if the routine has been called by the EDIT or MODIFY routines, after the insertion of a single character, control is returned to the calling routine (line 1160).

Lines 2000-2130 : WRITE FONT TO DISK

This section is a standard routine to output the font to the disk specified by drive number DO. The initial lines of this routine give the user the option of changing either the filename or the disk drive number. All routines in this program assume that the disk controller card is inserted in slot #6. After completion of the write routine, control is returned to the main menu beginning at line 100.

Lines 3000-3260 : ROUTINES TO MODIFY A FONT

The routines in this section allow the user to modify a font in a number of ways. The menu for this section (lines 3000-3050) indicates that the user may APPEND to a font, INSERT a character, REPLACE a character, REMOVE a character, or EXIT to the display routine of line 4000. In all modes but EXIT, the font insert flag, (FI), is set and the font is displayed. if the APPEND mode is selected, then the character location variable, L, is set to the end of the font and control is passed to the character input routine of line 1000- (line 3100). In the INSERT, REPLACE, and REMOVE modes, the following procedures take place:

INSERT (line 3150-) requests the location to insert, LO, moves all font characters from LO to the end of the font, LMAX, up one location (line 3170), displays the adjusted font and saves it on hires page 2 (line 3180), jumps to the character input routine for the inputting of one character, displays the new font (line 3220), and returns to the menu.

REPLACE (line 3230-) requests the character number to replace, LO, displays the font (line 3210), jumps to the input routine to input a character, displays the new font, and returns to the menu.

REMOVE (line 3230-) requests the character number to delete, LO, moves all characters from LO to the end of the font, LMAX, down 1 (line 3250), displays the new font (line 3260), and returns to the menu.

In all of these routines, the current font is saved in the memory space reserved for the hires page 2 graphics ($4000-$6000) by the hires page flip routines of lines 11000-.

Lines 4000-4070 : DISPLAY THE FONT DRIVING ROUTINES

The routines of this section are used to display the fonts on
the hires graphics pages. In line 4000, a jump is made to the
disk access routine at 7000 to input the font from the disk if a
new filename is used to access the data. The routine of line 4010
displays the font, N$, character by character, if the memory flag,
MF, is not set. If the memory flag is set, then the font has been
saved on hires page 2, and a call to the memory flip routines at
11000- recalls it for display. Line 4030 returns control to the
calling routine if the font input flag, FI, is set. Otherwise,
the remainder of the routine allows the user to either write the
font to the disk, or return to the main menu.

Lines 5000-5460: THE CHARACTER EDIT ROUTINE

These routines allow the user to edit a character by entering
either the Hershey coordinates or by using a hires cursor input.
Line 5040 displays the entire font and the user is prompted (line
5060), to enter the number of the character to be changed. This
character is then displayed in a magnified version by the routine
of lines 8000-. Figure 2 illustrates this display for the Gothic
letter N. Lines 5070-5120 perform this display by determining the
appropriate x and y coordinates by decoding the string describing
the character. After the character display, line 5125 gives the
user the option of returning to the start of the edit routine if
the wrong character has been selected to edit. Otherwise, the
user has the option of either entering the Hershey coordinates
(lines 1000-), or redrawing the character using the cursor
control.   The cursor control input technique allows the
generation of custom characters by manipulating a high resolution
cursor on the high resolution display. Initially, the user is
requested to enter the limits of the character by defining the
left and right marker locations (lines 5150-5190). The cursor is
then placed on the left marker, and may be moved to the
appropriate location by use of the I, J, K, and M keys of the
keyboard. The current status of the pen is given on the text
display and may be changed at any time by striking the space bar.

C.E. Dick and J.Hilsenrath

These cursor moves and the pen position are entered by the GET
input command of line 5250. When the cursor is positioned at the
appropriate position, the return key draws a line (with the
appropriate pen mode) from the previous location to the current
cursor location, enters the values into the character string, and
returns for the next line input (lines 5340-5350). As in the
Hershey input mode, / terminates the character. If the cursor is
not centered at the right marker, vectors are added to the string
to complete this move (lines 5360-5380). After verification of
the validity of the character, the font is displayed and the
routine is terminated if either the font input, FI, or the edit
flag, EF, is set. The remainder of the routine is used to enter
characters under cursor control if the NEW FONT option of the main
menu is being performed. Note also, that the new font is saved by
the page transfer routine of lines 11000-.

<div align="center">Lines 6000-6180 : HIRES FONT DISPLAY ROUTINES</div>

The routines of this section display the members of a font on
hires page 1 depending on their location in the font. Lines
6000-6020 determine the location at which the character is to be
displayed (with 50 characters displayed at a time), line 6030
removes any previously displayed character at that location, and
the remainder of the routine decodes the character string and
draws the character. The variables XP and YP are used for the
current cursor location, and DX and DY are used for the offset to
the next location. As always, the penflag, PF, is used to
determine whether the vector is to be plotted with the pen up
(HCOLOR BLACK) or down (HCOLOR WHITE).

<div align="center">Lines 7000-7110 : FONT INPUT ROUTINES</div>

The font N$ is inputted from disk drive DI, if it is not
already saved on hires page 2. If the font is to be brought in
from the disk, the memory flag, MF, is reset and the data pointer
is restored. Otherwise, the routine is staightforward and the
file is read until the occurrence of the end of file character
string, EOF. At the completion of the read routine, the file name
is saved in the variable O$.

Lines 8000-8090 : HIRES SINGLE CHARACTER GRAPHICS

These routines draw the hires single characters that are input
either by using the Hershey vectors or the cursor control. Lines
8000-8050 draw the box for the display of the character as
illustrated in Fig. 1. In this representation, each dot
corresponds to one unit in the character. Lines 8060-8090 plot
the figure from the decoded character string values. Line 8060
sets the magnification, M, to 5, and plots the left and right
boundaries relative to the center of the display (x=140,y=80).
Line 8070 sets the next point as the center of the left marker.
Subsequent values of X and Y are then plotted with the appropriate
pen control by the remainder of the routine. The current location
is stored in the variables X1 and Y1.

Lines 9000-9010 : CATALOG

This routine lists the catalog of the disk currently specified
as the input disk, DI.

Line 10000 : END

This is a standard exit routine.

Lines 11000-11060 : THE HIRES MEMORY FLIP ROUTINE

This routine copies the contents of hires page 1 to hires page
2 or vice versa. The starting address, the ending address, and
the first address of the destination field are read from the data
statement at line 11060. Depending on the state of the memory
flag, MF, page 1 is sent to page 2 (MF=0) or vice versa (MF=1).
The microprocessor's y and x register locations are zeroed by the
POKE statements of line 11030 and the Program Counter is
initialized to point to the Apple Monitor MOVE routine ($FE2C)
which is then called by the Apple Monitor GO routine ($FEB6).

The fonts that have currently been converted to the notation described in this report, are for the most part taken directly from the tables of Wolcott and Hilsenrath. For convenience, they are divided up into 32 character segments which we have chosen to name numbers, small, caps, and greek. Figures 3 to 8 illustrate the characters that make up the following fonts (in each subset, the characters are numbered from 1 to 32):

MATH SYMBOLS... A collection of commonly used symbols in mathematics taken from the Hershey characters between no.'s 2225 and 2278 (as referenced in the Wolcott and Hilsenrath tables).

CARTO FONT... Numbers and caps are taken from the Hershey cartographic symbols from no.'s 1 to 234. Although the Hershey fonts do not include small letters, these were custom designed and included as part of this font.

INDEX FONT... The members of this font were taken from the complex index size Hershey characters from no.'s 1001 to 1150.

SIMPLEX FONT... The members of this set were taken from the print size simplex Hershey characters from no.'s 501 to 551.

COMPLEX FONT... The members of this font were taken from the print size complex Hershey characters from no.'s 2200 to 2224.

GOTHIC FONT... This set is a mixture of two different Hershey fonts. The numbers are from the print size gothic Hershey characters from no.'s 3700 to 3729, while the small letters and caps are from the Hershey Italian gothic set (no.'s 3901 to 3950) and the Hershey print size English gothic set (no.'s 3700 to 3750), respectively.

SCRIPT FONT... This font is taken from the print size Hershey complex script characters from no.'s 2550 to 2576, 2650 to 2676, and 2750 to 2776.

CONCLUSION

Although the fonts that we have stored as text files using
CANDY APPLE are composed mainly of the Hershey characters, it is
possible using this program to custom design graphics symbols. In
the complex caps font illustrated in Fig. 6, the characters
numbered from 28 to 32 are custom designed for logos used at NBS.
This customization can be accomplished either by drawing the
desired figure on graph paper and then calculating the desired
coordinates or by freehand drawing using the cursor input routines
of lines 5000-. The current report illustrates over 700
characters which are stored on disk as text files. Listing 3
gives a catalog of the disk containing the programs CANDY APPLE
and &LOMEM: as well as the fonts. Reference to this listing shows
that the programs need 36 sectors for storage and the fonts use an
additional 181 sectors leaving more than 270 sectors avaiable for
applications programs. Such an application program, CANDY
WRAPPER, is described in Appendix A. Copies of CANDY APPLE,
&LOMEM:, CANDY WRAPPER, and the fonts may be obtained by
contacting the authors at the following mailing address:

C-216, Bldg. 245
National Bureau of Standards
Washington, DC 20234

REFERENCES

1. A.V. Hershey, "Calligraphy for Computers," NWL Technical Report No. 2101, August 1967, U.S. Naval Surface Weapons Center, Dahlgren, VA.

2. A.V. Hershey, "FORTRAN IV Programming for Cartography and Typography," NWL Technical Report No. TR-2339, Sept.1969, U.S. Naval Weapons Laboratory, Dahlgren, VA.

3. A.V. Hershey, "Preparation of Reports with the FORTRAN Typography System," NWL Technical Report No. TN-K/27-70, September 1970, U.S. Naval Weapons Laboratory, Dahlgren, VA.

4. A.V. Hershey, "Auxiliary Programming for the FORTRAN Typography System," NWL Technical Report No. TR-2645, October 1971, U.S. Naval Weapons Laboratory, Dahlgren, VA.

5. A.V. Hershey, "A Computer System for Scientific Typography," Computer Graphics and Image Processing, Vol.1, 373-385 (1972).

6. A.V. Hershey, "FORTRAN Programming for Surface Wave Trains," NWL Technical Report No. TR-2714, September 1972, U.S. Naval Weapons Laboratory, Dahlgen, VA.

7. A.V. Hershey, "Interpolations of Surface Wave Trains," NWL Technical Report No. TR-3064, November 1973, U.S. Naval Weapons Laboratory, Dahlgren, VA.

8. A.V. Hershey, "Comments on the Lagally Theorem," NWL Technical Report No. TR-3133, May 1974, U.S. Naval Weapons Laboratory, Dahlgren, VA.

9. A.V. Hershey, "Comments on the Fourier Theorem," NWL Technical Report No. TR-3145, June 1974, U.S. Naval Weapons Laboratory, Dahlgren, VA.

10. A.V. Hershey, "Advanced Computer Typography," Report NPS012-81-005, December 1981, U.S. Naval Postgraduate School, Monterey, CA.

11. P.M. Doyle, "An Adaptation of the Hershey Digitized Character Set for Use in Computer Graphics and Typesetting," Report AD/A-042291, June 1977, U.S. Naval Postgraduate School, Monterey, CA.

12. N.M. Wolcott and J. Hilsenrath, "A Contribution to Computer Typesetting Techniques: Tables of Coordinates for Hershey's Repertory of Occidental Type Fonts and Graphics Symbols," NBS Special Publication No. 424, National Bureau of Standards, Washington, DC, 20234.

13. N.M. Wolcott and J. Hilsenrath, "Tables of Coordinates for
     Hershey's Repertory of Occidental Type Fonts and Graphic
     Symbols," NBS Magnetic Tape 12, January 1977, National
     Bureau of Standards, Washington, DC, 20234.

14. N.M. Wolcott, "FORTRAN IV Enhanced Character Graphics," NBS
     Special Publication 500-32, April 1978, National Bureau of
     Standards, Washington DC, 20234.

15. P.K. Schenk, "Hershey Calligraphy Under the Hewlett-Packard
     Graphics Language," a forthcoming NBS report.

16. C.A.E. Kagan, SAM-76 Technical Note, August 1981, SAM 76 Inc.
     Box 257, RR1, Pennington, NJ, 08534

17. D. Macero, D. Holmes, T. Banks, and L. Burgess, "Graphics II
     by Selnar, High Resolution Copy from a DECwriter," BYTE,
     Vol.7, 172-196 (1982).

18. DOS 3.3, Copyright 1980, Apple Computer Inc., 10260 Bandley
     Drive, Cupertino, CA, 95014. Apple Product #A2L0036.

19. Applesoft Toolkit, Copyright 1980, Apple Computer Inc.,
     10260 Bandley Drive, Cupertino, CA, 95014. Apple Product
     #A2L0038.

20. Applesoft, Copyright 1978, Apple Computer Inc., 10260 Bandley
     Drive, Cupertino, CA, 95014. Apple Product #A2L0006.

21. Apple-doc, Copyright 1979, Roger Wagner, Southwestern Data
     Systems, Box 582, Santee, CA, 92071.

22. N. Konzen, "The &LOMEM: Utility," The Apple Orchard, Vol.1,
     p.21(Mar. 1981). Published by The International Apple
     Corps., P.O. Box 976, Daly City, CA, 94017. this utility
     is 159 bytes in length and is located at locations $0330 to
     $03CE.   The listing is as follows:

```
0330- A9 4C 8D F5 03 A9 40 8D F6 03 A9 03 8D F7 03 60
0340- A9 A4 20 C0 DE 20 67 DD 20 52 E7 A5 67 85 96 A5
0350- B0 85 97 C6 96 38 A5 50 85 94 E5 67 85 50 A5 51
0360- 85 95 E5 68 85 51 A5 AF E5 67 A8 A5 B0 E5 68 AA
0370- 18 65 95 85 95 C8 D0 01 E8 E8 C8 20 C3 D3 A2 69
0380- 20 C1 03 A2 AF 20 C1 03 A2 67 20 C1 03 A2 79 20
0390- C1 03 A5 B9 C9 02 F0 05 A2 B8 20 C1 03 A5 67 A4
03A0- 65 85 5E 84 5F A0 00 38 B1 5E 65 50 91 5E AA C8
03B0- B1 5E F0 0A 65 51 91 5E 86 5E 85 5F 90 E7 4C 6C
03C0- D6 38 B5 00 65 50 95 00 B5 01 65 51 95 01 60
```

The utility is invoked by the ampersand call &LOMEM: no.
where no. is the first memory location for the relocated
Applesoft program (See Line 20 of Candy Apple).

In order to illustrate the use of the character fonts created
by use of the CANDY APPLE program, we have written a program
named CANDY WRAPPER to place these characters anywhere on the
hires screen. This program is given in Listing 4 and uses some of
the routines developed for CANDY APPLE to decode and plot the
characters. The program is menu driven with the following options
available to the user:

1. Change the disk drives for inputting and outputting
   a hires picture.
2. Display the catalog of a selected disk drive.
3. Set the environment for drawing the hires characters.
   This option allows the user to select foreground and
   background colors from the hires color set.
4. Load a previously saved hires picture from disk.
5. Save a hires picture to disk.
6. Add text to a hires picture.
7. Change from one Hershey font to another.
8. Clear the hires screen.
9. Display the hires screen.
10. Exit the program.

In general, the program is self-explanatory and uses a
non-destructive hires cursor to position the start of the Hershey
character on the screen. The program does not clear the hires
screen at initialization so that a previously generated picture
may be left in page 1 for the addition of text. In addition, the
program is relocated to protect the primary hires page. For
convenience, the fonts are read in as 32 character increments
called small, caps, numbers, and greek as stored by the program
CANDY APPLE. Figure 9 illustrates a logo produced by CANDY
WRAPPER as a frontispiece for a listing of CANDY APPLE. This
figure and all others used in this report were produced by a
screen dump of the hires graphic page produced by the programs to
a dot-matrix printer.

Listing 1. Candy Apple program.

```
10    TEXT : HOME : HTAB (14): VTAB (10): PRINT "CANDY APPLE": REM  A HERSHE
      Y CHARACTER FILEHANDLER FOR THE APPLE II
20    PRINT  CHR$ (4);"BRUN LOMEM:": &  LOMEM: 24576: REM     RELOCATES A/S T
      O START AT $6000 (ABOVE HGR 1 & 2)
30 BELL$ =  CHR$ (7)
40    HOME : HGR : HGR2
50 D$ =  CHR$ (4): DIM X$(160): REM  SAVE SPACE FOR 5 FONTS OF 32 CHARACTE
      RS
60    POKE 768,01: POKE 769,00: POKE 770,04: POKE 771,00: POKE 772,58: POKE
      773,36: POKE 774,45: POKE 775,54: POKE 776,07: POKE 777,0: POKE 232,0
      0: POKE 233,03: REM  LOAD SHAPE TABLE FOR HIRES CURSOR
70    TEXT : HOME : HTAB (14): VTAB (10): PRINT "CANDY APPLE": REM  START AN
      D INIT. DISK DRIVES
80    PRINT BELL$: INPUT "READ FROM DISK DRIVE:";DI: IF DI < 1 OR DI > 2 THEN
      80
90    PRINT BELL$: INPUT "WRITE TO DISK DRIVE:";DO: IF DO < 1 OR DO > 2 THEN
      90
100   HOME : REM  MAIN MENU
110   PRINT : PRINT "DO YOU WISH TO....": PRINT : PRINT  TAB( 10);"1.CREATE
       A NEW FONT": PRINT : PRINT  TAB( 10);"2.MODIFY AN EXISTING FONT": PRINT
      : PRINT  TAB( 10);"3.EDIT FONT CHARACTERS"
120   PRINT : PRINT  TAB( 10);"4.DISPLAY (TRANSFER) A FONT": PRINT : PRINT
       TAB( 10);"5.ASSIGN DISK DRIVES": PRINT : PRINT  TAB( 10);"6.CATALOG
      OF DISK :";DI: PRINT : PRINT  TAB( 10);"7.QUIT": PRINT BELL$: INPUT "
      WHICH?";C
130   IF C < 1 OR C > 7 THEN 100
140   IF C > 4 THEN 180
150   PRINT : PRINT BELL$;: IF N$ <  > "" THEN  VTAB (23): PRINT "RETURN US
      ES FILENAME: ";O$
160   VTAB (21): INPUT "FILENAME: ";N$: IF  LEN (N$) = 0 THEN N$ = O$
165 O$ = N$
170   HOME
180   ON C GOTO 999,3000,5000,4000,70,9000,10000
999   HGR2 : TEXT
1000   PRINT "* ROUTINE TO CREATE AN HERSHEY FILE     NAME: ";N$
1010   PRINT BELL$: PRINT "INPUT HERSHEY COOR.<H> OR USE CURSOR    CONTROL<
      C>?";: GET A$: PRINT A$: IF A$ <  > "H" AND A$ <  > "C" THEN 1010
1020 EF = 0: IF A$ = "C" THEN EF = 1
1030   IF EF THEN 1090
1040   PRINT BELL$: PRINT "* INPUT THE HERSHEY COORDINATE PAIRS ON THE PROM
      PTING BELL "
1050   PRINT : PRINT "* THE CHARACTER READ ROUTINE ASSUMES    THAT THE PEN
      STARTS IN THE UP POSITION. TO SIGNAL A PEN CHANGE, INSERT A SPACE  BE
      TWEEN THE COORDINATE PAIRS."
1060   PRINT : PRINT "* TO SIGNAL THE END OF THE CHARACTER    INSERT THE DE
      LIMITER /"
1070   PRINT : PRINT "* TO SIGNAL THE END OF THE FILE ENTER   THE EOF CHARA
      CTER <E>."
1080   PRINT : PRINT "* THE FIRST COORDINATE PAIR INDICATES   THE WIDTH OF
      THE CHARACTER...TO ENTER   A NULL CHARACTER ENTER 0,0"
1090   PRINT BELL$: PRINT "READY? ";: GET Q$: PRINT Q$: IF Q$ <  > "Y" THEN
      HOME : GOTO 1000
1100   HOME : IF FI THEN 1120
1110   IF C THEN L = 0
1120   GOSUB 8000: PRINT  TAB( 10);"FILENAME:";N$:A$ = ""
1130   PRINT : PRINT  TAB( 10);"CHARACTER STRING NO:";L:X$(L) = "":Z =  FRE
      (0):PF = 0
1140   IF L = 160 THEN X$ = "E": GOTO 1220
```

17

```
1150    IF EF THEN A$ = "N": GOSUB 5190: IF EF = 0 THEN X$ = "E": GOTO 1220
1160    IF EF AND FI THEN  RETURN
1170    IF EF THEN L = L + 1: GOTO 1120
1180 X$ = "":Y$ = ""
1190    PRINT BELL$: TAB( 5);"HOR.COOR:";
1200    PRINT A$;: GET A$: IF A$ < > " " AND A$ < >  CHR$ (13) THEN X$ = X
        $ + A$: GOTO 1200
1210    IF  RIGHT$ (X$,1) =  CHR$ (9) THEN  PRINT BELL$;"REINPUT :";X$ = "":
        GOTO 1190
1220    IF X$ = "E" THEN X$(L) = "EOF": PRINT "ND OF FILE...";: PRINT :LMAX =
        L
1230    IF X$ = "E" THEN  PRINT BELL$;"WRITE TO DISK<W> OR RETURN TO MENU<M>
        ?";: GET Q$: PRINT Q$: IF Q$ < > "W" AND Q$ < > "M" THEN 1230
1240    IF Q$ = "W" THEN 2000
1250    IF Q$ = "M" THEN  TEXT : GOTO 100
1290    IF X$ = "" THEN X$(L) = X$(L) +  CHR$ (127):PF =  NOT (PF): PRINT BE
        LL$;"PEN CONT.": GOTO 1180
1300    IF X$ = "/" AND  LEN (X$(L)) THEN X$(L) = X$(L) +  CHR$ (127) +  CHR$
        (XF - XT + 93) +  CHR$ (93 - YT): GOTO 1420
1310    IF X$ = "/" AND  LEN (X$(L)) = 0 THEN  PRINT BELL$;"ERROR...REINPUT"
        : GOTO 1170
1320    PRINT BELL$;" VERT.COOR:";
1330    PRINT B$;: GET B$: IF B$ < > " " AND B$ < >  CHR$ (13) THEN Y$ = Y
        $ + B$: GOTO 1330
1340    IF  RIGHT$ (Y$,1) =  CHR$ (9) THEN  PRINT BELL$;"REINPUT :";Y$ = "":
        GOTO 1320
1350    PRINT
1360    IF  LEN (X$(L)) = 0 THEN XT =  VAL (X$):XF =  VAL (Y$):W =  ABS (XF -
        XT):YT = 0:X$(L) =  CHR$ (W + 93): GOSUB 8060: GOTO 1170
1370 XC =  VAL (X$):YC =  -  VAL (Y$):X = XC - XT:Y = YC - YT: IF  ABS (X)
        > 32 OR  ABS (Y) > 32 THEN  PRINT BELL$;"TO LARGE A DISPLACEMENT,INP
        UT SMALLER VALUES": GOTO 1170
1380    GOSUB 8080
1390 XT = XC:YT = YC
1400 X$(L) = X$(L) +  CHR$ (X + 93) +  CHR$ (Y + 93): IF PF = 0 THEN X$(L)
        = X$(L) +  CHR$ (127):PF = 1
1410    GOTO 1170
1420    PRINT "CHARACTER WIDTH=";W
1430    PRINT BELL$;"CHARACTER O.K.?";: GET Q$: PRINT Q$: IF Q$ < > "Y" AND
        Q$ < > "N" THEN 1430
1440    IF Q$ = "Y" AND FI THEN  RETURN
1450    IF Q$ = "N" THEN  PRINT BELL$;"RE-INPUT ";
1460 L = L + 1: GOTO 1120
2000    TEXT : HOME : PRINT "WRITE TO FILENAME:";N$: PRINT "DISK DRIVE NO.";
        DO: PRINT BELL$: REM   PUT IT ON DISK DO
2010    PRINT "ARE THESE O.K.? ";: GET A$: PRINT A$: IF A$ < > "Y" AND A$ <
        > "N" THEN 2010
2020    IF A$ = "Y" THEN 2050
2030    PRINT BELL$;: INPUT "FILENAME: ";N$
2040    PRINT BELL$;: INPUT "WRITE TO DISK DRIVE#";DO: IF DO < 1 OR DO > 2 THEN
        2040
2050    PRINT D$;"OPEN ";N$;",D";DO
2060    PRINT D$;"DELETE ";N$
2070    PRINT D$;"OPEN ";N$
2080    PRINT D$;"WRITE ";N$
2090    FOR I = 0 TO L
2100    PRINT X$(I)
2110    NEXT I
2120    PRINT D$;"CLOSE ";N$
2130    TEXT : GOTO 100
```

18

```
3000    TEXT : HOME : PRINT : PRINT "*ROUTINES TO MODIFY A HERSHEY FONT": PRINT
        : PRINT  TAB( 10);"FONTNAME:";N$
3010    PRINT : PRINT  TAB( 5);"YOU MAY...": PRINT : PRINT  TAB( 5);"1.APPEN
        D TO A FONT"
3020    PRINT : PRINT  TAB( 5);"2.INSERT IN THE MIDDLE OF A FONT": PRINT : PRINT
        TAB( 5);"3.REPLACE A CHARACTER"
3030    PRINT : PRINT  TAB( 5);"4.REMOVE A CHARACTER": PRINT : PRINT  TAB( 5
        );"5.EXIT"
3040    PRINT : PRINT BELL$; TAB( 5);"WHICH";: INPUT C
3050    IF C < 1 OR C > 5 THEN 3000
3060    IF C = 5 THEN FI = 0:L = LMAX: GOTO 4000
3070    IF FI = 0 THEN FI = 1: GOSUB 4000: GOTO 3070
3080    IF FI AND C > 1 THEN  GOSUB 4010
3090    ON C GOTO 3100,3110,3110,3110
3100 FI = 0:C = 0:L = LMAX: TEXT : HOME : GOTO 1000
3110    IF MF THEN  GOSUB 11000
3120    VTAB (21): PRINT "NOTE...FIRST CHARACTER IS NO.0"
3130    PRINT BELL$;"LOCATION TO ";
3140    ON C - 1 GOTO 3150,3200,3230
3150    PRINT "INSERT";: INPUT LO: IF LO < 0 OR LO > LMAX THEN  PRINT BELL$;
        "OUT OF RANGE": GOTO 3130
3160 LMAX = LMAX + 1
3170    FOR K = LMAX TO LO + 1 STEP  - 1:X$(K) = X$(K - 1): NEXT K
3180    FOR L = LO + 1 TO LMAX: GOSUB 6000: NEXT : GOSUB 11000
3190    GOTO 3220
3200    PRINT "REPLACE";: INPUT LO: IF LO < 0 OR LO > LMAX - 1 THEN  PRINT B
        ELL$;"OUT OF RANGE": GOTO 3130
3210    GOSUB 11000
3220 L = LO: TEXT : HOME : GOSUB 1010:EF = 0: GOSUB 6000: GOTO 3000
3230    PRINT "DELETE";: INPUT LO: IF LO < 0 OR LO > LMAX THEN  PRINT BELL$;
        "OUT OF RANGE": GOTO 3130
3240 LMAX = LMAX - 1
3250    FOR K = LO TO LMAX:X$(K) = X$(K + 1): NEXT
3260    FOR L = LO TO LMAX: GOSUB 6000: NEXT : GOSUB 11000: GOTO 3000
4000    GOSUB 7000: REM  GET FONT AND DISPLAY IT
4010    HGR : IF MF = 0 THEN  FOR L = 0 TO LMAX: HTAB (12): VTAB (24): PRINT
        "CHAR.NO.";L: GOSUB 6000: NEXT L: GOSUB 11000: GOTO 4030
4020    GOSUB 11000: IF N$ = O$ THEN  GOSUB 11000
4030    IF FI THEN  RETURN
4040    VTAB (21): HTAB (10): PRINT N$
4050    PRINT BELL$;"WRITE TO DISK <W> OR RETURN TO MENU <M>";: GET A$: PRINT
        A$: IF A$ < > "W" AND A$ < > "M" THEN 4050
4060    IF A$ = "W" THEN 2000
4070    TEXT : HOME : GOTO 100
5000    TEXT : HOME : PRINT : PRINT "*ROUTINE TO EDIT INDIVIDUAL CHARACTERS"

5010    PRINT : PRINT "YOU WILL FIRST BE SHOWN ENTIRE FONT     AND THEN ASKE
        D TO CHOOSE THE CHARACTER  TO EDIT."
5020    PRINT : PRINT "THE CHARACTER WILL BE DRAWN AS IS AND   THEN YOU CAN
        REDRAW IT BY EITHER INPUT- TING HERSHEY VECTORS OR BY USING THE    ED
        ITING KEYS."
5030    PRINT BELL$: PRINT "READY?";: GET A$: PRINT A$: IF A$ < > "Y" THEN
        5000
5040 FI = 1: GOSUB 4000:FI = 0
5050    VTAB (24)
5060    PRINT BELL$;"NOTE...FIRST CHARACTER IS NO.0": INPUT "CHARACTER NO. T
        O EDIT?";L: PRINT : IF L < 0 OR L > LMAX - 1 THEN  PRINT BELL$;"OUT O
        F RANGE": GOTO 5060
5070    PRINT : GOSUB 8000:X$ = "": PRINT  TAB( 13);"OLD CHARACTER"
5080 W =  ASC ( LEFT$ (X$(L),1)) - 93:XF = W -  INT (W / 2):XT =  -  INT (
```

19

```
      W / 2): GOSUB 8060
5090 PF = 0: FOR J = 2 TO  LEN (X$(L)) STEP 2
5100 X =  ASC ( MID$ (X$(L),J,1)) - 93: IF X = 34 THEN PF =  NOT PF:J = J -
     1: NEXT
5110 Y =  ASC ( MID$ (X$(L),J + 1,1)) - 93
5120  GOSUB 8080: NEXT J
5125  PRINT "READY TO CHANGE";: INPUT A$: IF A$ < > "Y" THEN 5000
5130  PRINT BELL$;"INPUT HERSHEY COOR.<H> OR USE CURSOR<C>?";: GET Q$: IF
     Q$ < > "H" AND Q$ < > "C" THEN 5130
5140  IF Q$ = "H" THEN FI = 1: GOSUB 1120:FI = 0: GOTO 5400
5150  GOSUB 8000
5160  PRINT "CURRENT WIDTH IS ";W: PRINT "LEFT MARKER AT:";XT: PRINT "RIGH
     T MARKER AT:";XF
5170  PRINT BELL$;"ARE THESE OK?";: GET A$: PRINT A$: IF A$ < > "Y" AND A
     $ < > "N" THEN 5170
5180  IF A$ = "N" THEN  PRINT BELL$;: INPUT "LEFT MARKER:";XT: PRINT BELL$
     ;: INPUT "RIGHT MARKER:";XF:W = XF - XT: GOTO 5160
5190 X$ =  CHR$ (W + 93): GOSUB 8060
5200  POKE 33,5: POKE 32,34: PRINT : PRINT " PEN": PRINT "MODE: ": POKE 33
     ,33: POKE 32,0:PF = 0
5210  SCALE= 1: ROT= 0:X = X1:Y = Y1: DRAW 1 AT X,Y: PRINT : PRINT "I,J,K,
     M KEYS MOVE CURSOR U,R,L,D"
5220  PRINT "RETURN ENDS MOVE, / ENDS CHAR."
5230  PRINT "SPACE FLIPS PEN, E ENDS FONT";: HTAB (37): PRINT  CHR$ (85 -
     17 * PF);
5240 DX = 0:DY = 0
5250  GET A$: PRINT BELL$;: IF  ASC (A$) < 73 OR  ASC (A$) > 77 THEN 5320
5260  XDRAW 1 AT X,Y: ON  ASC (A$) - 73 GOTO 5280,5290,5300,5300
5270 DY = DY - 1:Y = Y - 5: GOTO 5310
5280 DX = DX - 1:X = X - 5:  GOTO 5310
5290 DX = DX + 1:X = X + 5:  GOTO 5310
5300 DY = DY + 1:Y = Y + 5
5310  DRAW 1 AT X,Y: GOTO 5250
5320  IF A$ < > "E" AND A$ < > "/" AND A$ < > " " AND  ASC (A$) < > 13
      THEN 5250
5330  IF A$ = " " THEN PF =  NOT PF:X$ = X$ +  CHR$ (127): HTAB (37): PRINT
      CHR$ (85 - 17 * PF): GOTO 5250
5340  IF  ASC (A$) = 13 THEN X$ = X$ +  CHR$ (DX + 93) +  CHR$ (93 - DY): IF
     PF THEN  HPLOT X1,Y1 TO X,Y
5350  IF  ASC (A$) = 13 THEN X1 = X:Y1 = Y: GOTO 5210
5360  IF A$ = "/" THEN X$(L) = X$: IF PF = 1 AND X1 < > 140 + M * XF AND
     Y1 < > 80 THEN X$(L) = X$(L) +  CHR$ (127) +  CHR$ (XF + (140 - X1) /
     5 + 93) +  CHR$ ((Y1 - 80) / 5 + 93)
5370  IF A$ = "/" AND  LEN (X$(L)) = 1 THEN X$(L) = X$(L) +  CHR$ (127) +
      CHR$ (93) +  CHR$ (93)
5380  IF A$ = "/" THEN  PRINT BELL$: PRINT "CHARACTER O.K.?";: GET Q$: PRINT
     Q$: IF Q$ < > "Y" AND Q$ < > "N" THEN 5380
5390  IF Q$ = "N" THEN 5150
5400  POKE 33,40: GOSUB 11000: GOSUB 6000: GOSUB 11000
5410  IF A$ = "/" AND (FI OR EF) THEN  RETURN
5420  IF A$ = "E" AND EF THEN EF = 0: RETURN
5430  PRINT BELL$;"FINISHED EDITING?";: GET A$: PRINT A$: IF A$ < > "Y" AND
     A$ < > "N" THEN 5430
5440  IF A$ = "N" AND FI THEN FI = 0: GOTO 5060
5450  IF A$ = "N" THEN FI = 1: GOSUB 4010:FI = 0: GOTO 5060
5460 L = LMAX:X$ = "E": PRINT " E";: GOTO 1220
6000  HCOLOR= 0:Z = L - 50 *  INT (L / 50): IF (Z / 50) =  INT (Z / 50) AND
     (L > 0) THEN  PRINT BELL$;"STRIKE ANY KEY TO CONTINUE...";: GET Q$: HGR
     : REM  DISPLAY ROUTINE
6010  IF  LEN (X$(L)) = 0 THEN  RETURN
```

```
6020 XP = 280 * (Z / 10 -  INT (Z / 10)):YP = 32 *  INT (Z / 10) + 16
6030  FOR I = 1 TO 14: SCALE= I: DRAW 1 AT XP + 14,YP: NEXT : HCOLOR= 3
6040 W =  ASC ( MID$ (X$(L),1,1)) - 93
6050 XP = XP +  INT ((28 - W) / 2)
6060  IF W = 0 THEN  HCOLOR= 6:XP = XP - 5:DX = 10
6070  IF W = 0 THEN  HPLOT XP,YP TO XP + DX,YP
6080  HCOLOR= 3:PF = 0
6090  FOR J = 2 TO  LEN (X$(L)) STEP 2
6100 DX =  ASC ( MID$ (X$(L),J,1))
6110 DY =  ASC ( MID$ (X$(L),J + 1,1))
6120  IF DX = 127 THEN PF =  NOT (PF):J = J - 1: NEXT J
6130 DX = (DX - 93):DY = (DY - 93)
6140  IF  LEN (X$(L)) < 5 THEN  HCOLOR= 5
6150  IF PF THEN  HPLOT XP,YP TO XP + DX,YP - DY
6160 XP = XP + DX:YP = YP - DY: NEXT J
6170  HOME : VTAB 21
6180  RETURN
7000  HOME : IF N$ = O$ AND MF = 1 THEN  RETURN : REM  READ FROM DISK DI
7010  PRINT "NOW READING FILE FROM DISK #";DI
7020  IF FI THEN  PRINT : PRINT "ENTIRE FONT WILL BE DISPLAYED"
7030 MF = 0: RESTORE
7040  PRINT D$;"OPEN ";N$;",D";DI
7050  PRINT D$;"READ ";N$
7060 I = 0
7070  INPUT X$(I)
7080  IF X$(I) = "EOF" THEN 7100
7090 I = I + 1: GOTO 7070
7100  PRINT D$;"CLOSE ";N$
7110 LMAX = I:O$ = N$: RETURN
8000  HGR : HCOLOR= 5: HPLOT 65,5 TO 215,5 TO 215,155 TO 65,155 TO 65,5: REM
      EDIT GRAPHICS ROUTINES
8010  HCOLOR= 6: HPLOT 140,5 TO 140,15: HPLOT 140,145 TO 140,155: HPLOT 65
    ,80 TO 75,80: HPLOT 205,80 TO 215,80
8020  HCOLOR= 3: FOR J = 10 TO 150 STEP 5: FOR K = 70 TO 210 STEP 5: HPLOT
    K,J: NEXT : NEXT
8030  HPLOT 140,78 TO 140,82: HPLOT 138,80 TO 142,80
8040  VTAB (24)
8050  RETURN
8060 M = 5: HCOLOR= 3: HPLOT 140 + M * XT,75 TO 140 + M * XT,85: HPLOT 140
      + XF * M,75 TO 140 + XF * M,85
8070 X1 = 140 + M * XT:Y1 = 80: RETURN
8080  IF PF THEN  HPLOT X1,Y1 TO X1 + M * X,Y1 - M * Y
8090 X1 = X1 + M * X:Y1 = Y1 - M * Y: RETURN
9000  TEXT : HOME : PRINT D$;"CATALOG,D";DI
9010  PRINT : PRINT BELL$;"STRIKE ANY KEY TO CONTINUE";: GET A$: GOTO 100
10000  HOME : PRINT BELL$;"NICE WORKING WITH YOU": END
11000  READ F%,L%,D%: REM  MEMORY LOCATIONS FOR MOVE
11010 FH% = F% / 256:FL% = F% - 256 * FH%:LH% = L% / 256:LL% = L% - 256 *
    LH%:DH% = D% / 256:DL% = D% - 256 * DH%
11020  POKE 60,FL%: POKE 61,FH%: POKE 62,LL%: POKE 63,LH%: POKE 66,DL%: POKE
    67,DH%
11030  POKE 70,0: POKE 71,0: POKE 58,44: POKE 59,254: REM  SETUP YREG,XREG
    ,PCL,PCH
11040  CALL  - 327:MF =  NOT MF: IF MF = 0 THEN  RESTORE
11050  RETURN
11060  DATA  8192,16383,16384,16384,24575,8192

]
```

Listing 2. Candy Apple variable listing.

```
***********************************
*                                 *
*            CANDY APPLE          *
*                                 *
*      -->TABLE OF VARIABLES<--   *
*                                 *
***********************************
```

A$ - ANSWER STRING--ALSO USED TO ENTER HERSHEY X COORDINATE
1010 1010 1010 1010 1020 1120 1150 1200 1200 1200 1200 1200 2010
2010 2010 2010 2020 4050 4050 4050 4050 4060 5030 5030 5030 5125
5125 5170 5170 5170 5170 5180 5250 5250 5250 5260 5320 5320 5320
5320 5330 5340 5350 5360 5370 5380 5410 5420 5430 5430 5430 5430
5440 5450 9010

B$ - STRING USED TO ENTER HERSHEY Y COORDINATE
1330 1330 1330 1330 1330

BELL$ - CHR$(7) SOUNDS SPEAKER
30 80 90 120 150 1010 1040 1090 1190 1210 1230 1290 1310 1320
1340 1370 1430 1450 2000 2030 2040 3040 3130 3150 3200 3230 4050
5030 5060 5060 5130 5170 5180 5180 5250 5380 5430 6000 9010
10000

C - NUMERICAL CHOICE INPUT
120 130 130 140 180 1110 3040 3050 3050 3060 3080 3090 3100
3140

D$ - CHR$(4) DOS CONTROL
50 2050 2060 2070 2080 2120 7040 7050 7100 9000

D% - DESTINATION ADDRESS FOR MEMORY MOVE
11000 11010 11010

DH% - DESTINATION HIGH BYTE
11010 11010 11020

DI - DISK DRIVE FOR INPUT
80 80 80 120 7010 7040 9000

DL% - DESTINATION LOW BYTE
11010 11020

DO - DISK DRIVE FOR OUTPUT
90 90 90 2000 2040 2040 2040 2050

DX - INCREMENTAL X
5240 5280 5280 5290 5290 5340 6060 6070 6100 6120 6130 6130 6150
6160

DY - INCREMENTAL Y
5240 5270 5270 5300 5300 5340 6110 6130 6130 6150 6160

EF - EDIT FLAG 0= NO EDIT 1=EDIT UNDERWAY
```

1020 1020 1030 1150 1150 1160 1170 3220 5410 5420 5420

F% - STARTING ADDRESS FOR MEMORY MOVE ROUTINE
11000 11010 11010

FH% - STARTING ADDRESS HIGH BYTE
11010 11010 11020

FI - GENERAL PURPOSE FONT INPUT FLAG
1100 1160 1440 3060 3070 3070 3080 3100 4030 5040 5040 5140 5140
5410 5440 5440 5450 5450 7020

FL% - STARTING ADDRESS LOW BYTE
11010 11020

I - GENERAL INDEX
2090 2100 2110 6030 6030 7060 7070 7080 7090 7090 7110

J - GENERAL INDEX
5090 5100 5100 5100 5110 5120 6090 6100 6110 6120 6120 6120 6160
8020 8020

K - GENERAL INDEX
3170 3170 3170 3170 3250 3250 3250 8020 8020

L - INDEX FOR LOCATION IN FONT
1110 1130 1130 1140 1170 1170 1220 1220 1290 1290 1300 1300 1300
1310 1360 1360 1400 1400 1400 1400 1460 1460 2090 3060 3100 3180
3220 3260 4010 4010 4010 5060 5060 5060 5080 5090 5100 5110 5360
5360 5360 5370 5370 5370 5460 6000 6000 6000 6010 6040 6090 6100
6110 6140

L% - ENDING ADDRESS FOR MEMORY MOVE ROUTINE
11000 11010 11010

LH% - ENDING ADDRESS HIGH BYTE
11010 11010 11020

LL% - ENDING ADDRESS LOW BYTE
11010 11020

LMAX - LOCATION OF EOF CHARACTER--MAXIMUM CHARACTER IN FONT
1220 3060 3100 3150 3160 3160 3170 3180 3200 3230 3240 3240 3250
3260 4010 5060 5460 7110

LO - LOCATION TO INSERT,REPLACE,ETC.
3150 3150 3150 3170 3180 3200 3200 3200 3220 3230 3230 3230 3250
3260

M - MAGNIFICATION FOR HIGH RES ROUTINES
5360 8060 8060 8060 8060 8060 8070 8080 8080 8090 8090

MF - MOVE FLAG 0=XFER FROM P1 TO P2   1=XFER FROM P2 TO P1
3110 4010 7000 7030 11040 11040 11040

N$ - FONT NAME (CURRENT)
150 160 160 160 165 1000 1120 2000 2030 2050 2060 2070 2080
2120 3000 4020 4040 7000 7040 7050 7100 7110

O$ - FONT NAME (PREVIOUS)

150  160  165  4020  7000  7110


PF - PEN FLAG  0=PEN UP   1=PEN DOWN
1130  1290  1290  1400  1400  5090  5100  5100  5200  5230  5330  5330  5330
5340  5360  6080  6120  6120  6150  8080

Q$ - STRING FOR QUESTION ANSWERS
1090  1090  1090  1230  1230  1230  1230  1240  1250  1430  1430  1430  1430
1440  1450  5130  5130  5130  5140  5380  5380  5380  5380  5390  6000

W - LENGTH (WIDTH) OF CHARACTER IN FONT
1360  1360  1420  5080  5080  5080  5080  5160  5180  5190  6040  6050  6060
6070

X - X LOCATION IN CHARACTER COORDINATE SPACE
1370  1370  1400  5100  5100  5210  5210  5260  5280  5280  5290  5290  5310
5340  5350  8080  8090

X$ - STRING CONTAINING CURRENT CHARACTER
1140  1150  1180  1200  1200  1210  1210  1220  1230  1290  1300  1310  1360
1370  5070  5190  5330  5330  5340  5340  5360  5460

X$(*) - STRING ARAY FOR FONT
50  1130  1220  1290  1290  1300  1300  1300  1310  1360  1360  1400  1400
1400  1400  2100  3170  3170  3250  3250  5080  5090  5100  5110  5360  5360
5360  5370  5370  5370  6010  6040  6090  6100  6110  6140  7070  7080

X1 - HIRES X LOCATION
5210  5340  5350  5360  5360  8070  8080  8080  8090  8090

XC - CURRENT X COORDINATE VALUE
1370  1370  1390

XF - FINAL X COORDINATE OF CHARACTER
1300  1360  1360  5080  5160  5180  5180  5360  5360  8060  8060

XP - HIRES X POSITION
6020  6030  6050  6050  6060  6060  6070  6070  6150  6150  6160  6160

XT - TEMPORARY X COORDINATE
1300  1360  1360  1370  1390  5080  5160  5180  5180  8060  8060  8070

Y - Y LOCATION IN CHARACTER COORDINATE SPACE
1370  1370  1400  5110  5210  5210  5260  5270  5270  5300  5300  5310  5340
5350  8080  8090

Y$ - CURRENT Y COORDINATE STRING FOR INPUT
1180  1330  1330  1340  1340  1360  1370

Y1 - HIRES Y COORDINATE
5210  5340  5350  5360  5360  8070  8080  8080  8090  8090

YC - CURRENT Y COORDINATE
1370  1370  1390

YP - HIRES Y POSITION
6020  6030  6070  6070  6150  6150  6160  6160

YT - TEMPORARY Y COORDINATE
1300  1360  1370  1390.

24

Z - CHARACTER INDEX FOR DISPLAY ROUTINE
1130 6000 6000 6000 6020 6020 6020

END OF VAR. LIST

Listing 3. Catalog of Candy Apple disk.

```
DISK VOLUME 254

*A 033 CANDY APPLE
*B 003 LOMEM!
*T 007 MATH SYMBOLS
*T 004 CARTO NUMBERS
*T 005 CARTO SMALL
*T 005 CARTO CAPS
*T 007 INDEX NUMBERS
*T 009 INDEX SMALL
*T 008 INDEX CAPS
*T 009 INDEX GREEK
*T 005 SIMPLEX NUMBERS
*T 006 SIMPLEX SMALL
*T 005 SIMPLEX CAPS
*T 004 SIMPLEX GREEK
*T 007 COMPLEX NUMBERS
*T 009 COMPLEX SMALL
*T 009 COMPLEX CAPS
*T 010 COMPLEX GREEK
*T 008 SCRIPT NUMBERS
*T 009 SCRIPT SMALL
*T 012 SCRIPT CAPS
*T 010 GOTHIC NUMBERS
*T 011 GOTHIC SMALL
*T 020 GOTHIC CAPS
 A 035 CA2
*T 007 CANDY APPLE VFILE

]
```

Listing 4. Candy Wrapper program.

```
10    TEXT : HOME : HTAB (14): VTAB (08): PRINT "CANDY WRAPPER"
20    PRINT  CHR$ (4);"BRUN LOMEM:"; &  LOMEM: 16384: REM  RELOCATES A/S TO
      START AT $4000 (ABOVE HGR 1)
30 BELL$ =  CHR$ (7)
40 D$ =  CHR$ (4): DIM X$(160): REM  SAVE SPACE FOR 5 FONTS OF 32 CHARACTE
      RS
50    DEF  FN A(A) = 8192 + 40 *  INT (A / 64) + 128 *  INT ((A - 64 *  INT
      (A / 64)) / 8) + 1024 * (A - 8 *  INT (A / 8))
60    POKE 768,01: POKE 769,00: POKE 770,04: POKE 771,00: POKE 772,58: POKE
      773,36: POKE 774,45: POKE 775,54: POKE 776,07: POKE 777,0: POKE 232,0
      0: POKE 233,03: REM  LOAD SHAPE TABLE FOR HIRES CURSOR
70    ONERR  GOTO 12000
80 X = 139:Y = 95:FF = 0
90    TEXT : HOME : HTAB (14): VTAB (08): PRINT "CHARACTER PLOT"
100   PRINT BELL$: INPUT "READ FROM DISK DRIVE:";DI: IF DI < 1 OR DI > 2 THEN
      100
110   PRINT BELL$: INPUT "WRITE TO DISK DRIVE:";DO: IF DO < 1 OR DO > 2 THEN
      110
120   PRINT BELL$: INPUT "FONTS FROM DISK DRIVE:";DF: IF DF < 1 OR DF > 2 THEN
      120
130   TEXT : HOME : HTAB (10): VTAB (06): PRINT "OPTIONS..."
140   PRINT : PRINT "1. CHANGE DISK DRIVES": PRINT "2. CATALOG DISK": PRINT
      "3. SET ENVIRONMENT"
150   PRINT "4. LOAD HIRES PIX": PRINT "5. SAVE HIRES PIX": PRINT "6. ADD T
      EXT  TO PIX": PRINT "7. CHANGE FONT": PRINT "8. CLEAR HIRES SCREEN": PRINT
      "9. DISPLAY CURRENT PICTURE": PRINT "10. QUIT"
160   PRINT BELL$: INPUT "WHICH--> ";C: IF C < 1 OR C > 10 THEN 160
170   ON C GOTO 90,1000,8000,2000,2000,3000,4000,5000,6000,7000
1000   TEXT : HOME
1010   PRINT "WHICH DRIVE ";: INPUT D
1030   PRINT D$;"CATALOG,D";D
1040   PRINT : PRINT BELL$;"STRIKE ANY KEY TO CONTINUE";: GET A$: GOTO 130
2000   HOME : HTAB (5): VTAB (10): INPUT "FILENAME OF PICTURE-->";N$
2010   ON C - 3 GOTO 2020,2030
2020   HGR : POKE  - 16302,0:X = 139:Y = 95: PRINT D$;"BLOAD ";N$;",D";DI: FOR
      I = 1 TO 2000: NEXT : GOTO 130
2030   PRINT D$;"BSAVE ";N$;",A$20                 00,L$2000,D";DO
2040   PRINT D$;"LOCK ";N$: GOTO 130
3000   IF FF = 0 THEN  PRINT "NEED FONTS TO CONTINUE...": FOR I = 1 TO 1000
      : NEXT : GOTO 4000
3010   TEXT : HOME : PRINT "PROCEDURE TO ADD TEXT TO PICTURE"
3020   PRINT : PRINT "1. INPUT TEXT TO ADD": PRINT "2. POSITION CURSOR USIN
      G I,J,K,M KEYS": PRINT "NOTE:KEY MAY BE PRECEEDED BY A NUMBER,  E.G.
      50I MOVES 50 SPACES RIGHT."
3030   PRINT "3. STRIKE CTRL-P TO PRINT TEXT": PRINT "4. <ESC> TO RETURN TO
      MENU, <SPACE> TO       CHANGE TEXT."
3040   PRINT : PRINT "STRIKE ANY KEY TO START-->";: GET A$: IF  ASC (A$) =
      27 THEN 130
3050   GOSUB 20000
3060   SCALE= 1: ROT= 0: HCOLOR= 7: POKE  - 16304,0: POKE  - 16297,0: POKE
       - 16302,0
3070 Z = 0:Z$ = "0": GOSUB 25000: DRAW 1 AT X,Y
3080   GET A$: PRINT BELL$: IF (A$ > ",") AND (A$ < > ".") AND (A$ < ":") THEN
      Z$ = Z$ + A$:Z =  VAL (Z$): GOTO 3080
3090   IF  ASC (A$) < 73 OR  ASC (A$) > 77 THEN 3190
3100   GOSUB 30000: FOR I = 0 TO Z: IF (X > 10) AND (X < 270) AND (Y > 10) AND
      (Y < 182) THEN  ON  ASC (A$) - 72 GOTO 3110,3130,3150,3150,3170
3110 Y = Y - 1: IF Y = 10 THEN Y = 11
```

27

```
3120   NEXT I: GOTO 3070
3130 X = X - 1: IF X = 10 THEN X = 11
3140   NEXT I: GOTO 3070
3150 X = X + 1: IF X = 270 THEN X = 269
3160   NEXT I: GOTO 3070
3170 Y = Y + 1: IF Y = 182 THEN Y = 191
3180   NEXT I: GOTO 3070
3190   IF ( ASC (A$) <  > 16) AND ( ASC (A$) <  > 32) AND ( ASC (A$) <  > 2
      7) THEN 3070
3200   IF  ASC (A$) = 27 THEN 130
3210   IF  ASC (A$) = 32 THEN 3000
3220   GOSUB 30000: GOSUB 22000: GOTO 3000
4000   REM  LETTER FONTS
4010   HOME
4020   PRINT : PRINT  TAB( 10);: INVERSE : PRINT "INPUT LETTER FONTS": PRINT
      : NORMAL : PRINT  TAB( 11);"FONTS AVAILABLE:"
4030   PRINT : PRINT  TAB( 10);"<1>  INDEX": PRINT  TAB( 10);"<2>  SIMPLEX"
      : PRINT  TAB( 10);"<3>  COMPLEX": PRINT  TAB( 10);"<4>  CARTOGRAPHIC"
      : PRINT  TAB( 10);"<5>  SCRIPT": PRINT  TAB( 10);"<6>  GOTHIC": PRINT
      : PRINT
4040 I = 1
4050 N$ = " NUMBERS": PRINT : INPUT "NUMBER FONT <1-6>:";N:N =  INT (N): IF
      N < 1 OR N > 6 THEN  PRINT "REINPUT ";: GOTO 4050
4060   GOSUB 4140:
4070 N$ = " CAPS": PRINT : INPUT "UPPER CASE FONT <1-6>:";N:N =  INT (N): IF
      N < 1 OR N > 6 THEN  POKE "REINPUT ";: GOTO 4070
4080   GOSUB 4140
4090 N$ = " SMALL": PRINT : INPUT "LOWER CASE FONT <1-6>:";N:N =  INT (N):
      IF N < 1 OR N > 6 THEN  PRINT "REINPUT ";: GOTO 4090
4100   GOSUB 4140
4110 N$ = " GREEK": PRINT : INPUT "GREEK FONT <1-3>:";N:N =  INT (N): IF N
      < 1 OR N > 3 THEN  PRINT "REINPUT ";: GOTO 4110
4120   GOSUB 4140
4130   PRINT :N$(5) = "MATH SYMBOLS": PRINT N$(5);" ALSO BEING LOADED": GOTO
      4210
4140   IF N = 1 THEN N$(I) = "INDEX"
4150   IF N = 2 THEN N$(I) = "SIMPLEX"
4160   IF N = 3 THEN N$(I) = "COMPLEX"
4170   IF N = 4 THEN N$(I) = "CARTO"
4180   IF N = 5 THEN N$(I) = "SCRIPT"
4190   IF N = 6 THEN N$(I) = "GOTHIC"
4200 N$(I) = N$(I) + N$:I = I + 1: RETURN
4210   PRINT : FLASH : PRINT "NOW LOADING FONTS": NORMAL
4220   FOR I = 1 TO 2: PRINT N$(I);"    ";: NEXT I: PRINT
4230   FOR I = 3 TO 4: PRINT N$(I);"    ";: NEXT I: PRINT : PRINT "MATH SYMB
      OLS"
4240 N = 0: FOR I = 1 TO 5
4250   PRINT D$;"OPEN ";N$(I);",D";DF
4260   PRINT D$;"READ ";N$(I)
4270   FOR JJ = 1 TO 32
4280   INPUT X$(N)
4290 N = N + 1
4300   NEXT JJ
4310   PRINT D$;"CLOSE ";N$(I)
4320   NEXT I:FF = 1
4330   HOME
4340   GOTO 130
5000   HGR : GOTO 130
6000   POKE  - 16304,0: POKE  - 16302,0: GET A$: GOTO 130
7000   HOME : PRINT BELL$;"NICE WORKING WITH YOU": END
```

28

```
8000   TEXT : HOME : PRINT "SET DRAWING ENVIRONMENT":BF = 0:DP = 0
8010   PRINT : PRINT "COLORS AVAILABLE..."
8020   PRINT "0. BLACK1": PRINT "1 . GREEN": PRINT "2. VIOLET": PRINT "3. W
       HITE1": PRINT "4. BLACK2": PRINT "5. ORANGE": PRINT "6. BLUE": PRINT
       "7. WHITE2"
8025   PRINT "8. NONE -- BACKGROUND ONLY"
8030   PRINT : INPUT "BACKGROUND COLOR:";BC: IF BC < 0 OR BC > 7 THEN BF =
       1:BC = 0
8040   PRINT : INPUT "FOREGROUND COLOR:";FC: IF FC < 0 OR FC > 7 THEN 8040
8045   PRINT : INPUT "SINGLE<1> OR DOUBLE<2> PRECISION:";N: IF N < 1 OR N >
       2 THEN 8045
8048   IF N = 1 THEN 8060
8050   DP = 1: PRINT : INPUT "CONTAST COLOR:";CC: IF CC < 0 OR CC > 7 THEN 8
       050
8060   GOTO 130
11000  READ F%,L%,D%: REM  MEMORY LOCATIONS FOR MOVE
11010  FH% = F% / 256:FL% = F% - 256 * FH%:LH% = L% / 256:LL% = L% - 256 *
       LH%:DH% = D% / 256:DL% = D% - 256 * DH%
11020  POKE 60,FL%: POKE 61,FH%: POKE 62,LL%: POKE 63,LH%: POKE 66,DL%: POKE
       67,DH%
11030  POKE 70,0: POKE 71,0: POKE 58,44: POKE 59,254: REM  SETUP YREG,XREG
       ,PCL,PCH
11040  CALL  - 327:MF =  NOT MF: IF MF = 0 THEN  RESTORE
11050  RETURN
11060  DATA  8192,16383,16384,16384,24575,8192
12000  REM  ERROR HANDLING ROUTINES
12010 GOOF =  PEEK (222)
12020  PRINT : PRINT BELL$;"ERROR NO.";GOOF;" ENCOUNTERED": PRINT "ERROR H
       ANDLER ACTIVE...";BELL$: FOR Z = 1 TO 1000: NEXT
12030  IF GOOF = 53 THEN  PRINT : PRINT BELL$;"NEED FONTS TO CONTINUE";BEL
       L$: FOR ZZ = 1 TO 1000: NEXT ZZ:B$(J + 6) = B$(J): HOME :B$(J) = "F":
       GOSUB 3000: RESUME
12040  IF (GOOF = 5) OR (GOOF = 6) THEN  PRINT "FILE NOT FOUND": FOR ZZ =
       1 TO 2000: NEXT : GOTO 130
12060  IF GOOF = 9 THEN  PRINT BELL$;"DISK FULL,INSERT NEW DISKETTE";BELL$

12070  IF GOOF = 10 THEN  PRINT BELL$;"WARNING!!, ";N$;" IS LOCKED": PRINT
       "DO YOU WISH TO DESTROY OLD FILE";: INPUT A$: IF  LEFT$ (A$,1) = "N" THEN
       GOTO 130
12080  IF  LEFT$ (A$,1) = "Y" THEN  PRINT D$;"UNLOCK ";N$
12090  IF GOOF = 255 THEN  STOP
12100  RESUME
20000  REM  *** TEXT GENERATOR ***
20020  PRINT : PRINT "CHARACTER STRING: ";:LC = 0:SC = 0:R$ = ""
20050  PRINT BELL$;: GET T$: IF T$ =  CHR$ (13) THEN LC = 0: NORMAL : GOTO
       20110
20060  IF T$ =  CHR$ (7) OR T$ =  CHR$ (19) THEN  FLASH :R$ = R$ + T$:SC =
       1: GOTO 20050
20070  IF T$ =  CHR$ (27) THEN LC =  NOT (LC): GOTO 20050
20080  IF  NOT (LC) AND  NOT (SC) THEN  INVERSE
20090  IF LC AND  NOT (SC) AND  ASC (T$) > 63 THEN  NORMAL :T$ =  CHR$ ( ASC
       (T$) + 128)
20100  PRINT T$;:R$ = R$ +  CHR$ ( ASC (T$)):SC = 0: GOTO 20050
20110  PRINT : PRINT BELL$: INPUT "SIZE :";IW:IW =  INT (IW): IF IW < 1 OR
       IW > 10 THEN  PRINT BELL$;"REINPUT :";: GOTO 20110
20130  PRINT BELL$: INPUT "HORIZ.<H> OR VERT.<V> CHARACTERS ?";L$: IF L$ <
       > "H" AND L$ < > "V" THEN  PRINT BELL$;"REINPUT :";: GOTO 20130
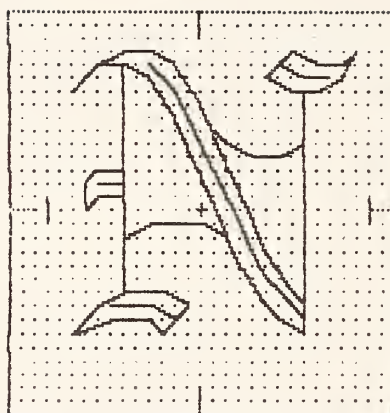20140 IV = 0: IF L$ = "V" THEN IV = 06
```

```
20180  IF  LEN (R$) = 0 THEN  PRINT BELL$;"ERROR-REINPUT:";: GOTO 20030
20190  RETURN
22000  REM  TEXT STRING DECODER
22010  HCOLOR= BC:SIZE = 0:W = 0
22020  PRINT : PRINT : FOR IC = 1 TO  LEN (R$)
22030 P$ =  MID$ (R$,IC,1):P =  ASC (P$): IF P = 7 OR P = 19 THEN IC = IC +
       1:P$ =  MID$ (R$,IC,1)
22040  REM  *** MAKE HERSHEY VECTORS ***
22050  IF P$ = "" THEN  PRINT BELL$;"ERROR-REINPUT ";:R$ = "": GOTO 20030
22060 L =  ASC (P$) - 32
22070  IF (P = 7 OR P = 19) AND (L > 95) THEN L = L - 128
22080  IF P = 7 THEN L = L + 64
22090  IF P = 19 THEN L = L + 96
22100  IF L > 159 THEN L = L - 96
22110 X$ = X$(L)
22120  IF SIZE = 0 THEN 23000
22130 PF = 0: HCOLOR= FC
22140  FOR G = 2 TO  LEN (X$(L)) STEP 2
22150 DX =  ASC ( MID$ (X$(L),G,1)) - 93
22160 DY =  ASC ( MID$ (X$(L),G + 1,1)) - 93
22170  IF DX = 34 AND  LEN (X$(L)) > 4 THEN PF =  NOT (PF)
22180  IF DX = 34 THEN G = G - 1: NEXT G
22190 DX = DX * IW:DY =   - DY * IW
22200  IF IV > 0 THEN TEMP = DX:DX = DY:DY =  - TEMP
22210  IF PF THEN  HPLOT X,Y TO X + DX,Y + DY
22215  IF PF AND DP THEN  HCOLOR= CC:X = X + 1: HPLOT X,Y TO X + DX,Y + DY
       :X = X - 1
22220 X = X + DX:Y = Y + DY
22230  NEXT G
22240  NEXT IC
22250  FOR I = 1 TO 2000: NEXT : RETURN
23000 W = W + ( ASC ( LEFT$ (X$,1)) - 93) * IW
23010  IF IV = 0 AND X + W > 278 THEN 23040
23020  IF IV > 0 AND Y - W < 2 THEN 23040
23030  NEXT IC:SIZE = 1: IF BF THEN 22020
23032  IF IV = 0 THEN  FOR I =  - 15 TO 15: HPLOT X - 2,Y + I * IW TO X +
       2 + W * IW,Y + I * IW: NEXT
23034  IF IV > 0 THEN  FOR I =  - 15 TO 15: HPLOT X + I * IW,Y + 2 TO X +
       I * IW,Y - 2 - W * IW: NEXT
23036  GOTO 22020
23040  TEXT : HOME : PRINT "TEXT TOO LONG...";BELL$;BELL$: GOTO 3020
25000 YU =  FN A(Y - 1):YM =  FN A(Y):YL =  FN A(Y + 1)
25010 XC =  INT (X / 7) - 1
25020  FOR I = 0 TO 2
25030 U(I) =  PEEK (YU + XC + I)
25040 M(I) =  PEEK (YM + XC + I)
25050 L(I) =  PEEK (YL + XC + I)
25060  NEXT I
25070  RETURN
30000  FOR I = 0 TO 2
30010  POKE YU + XC + I,U(I)
30020  POKE YM + XC + I,M(I)
30030  POKE YL + XC + I,L(I)
30040  NEXT I
30050  RETURN
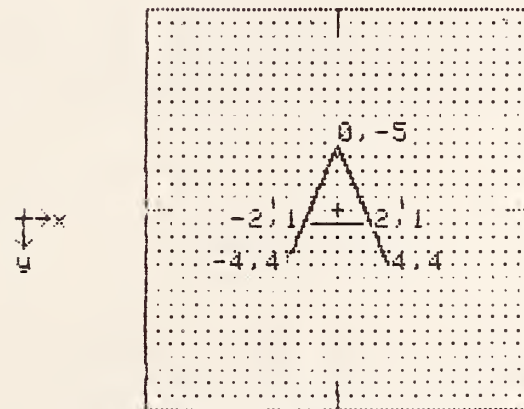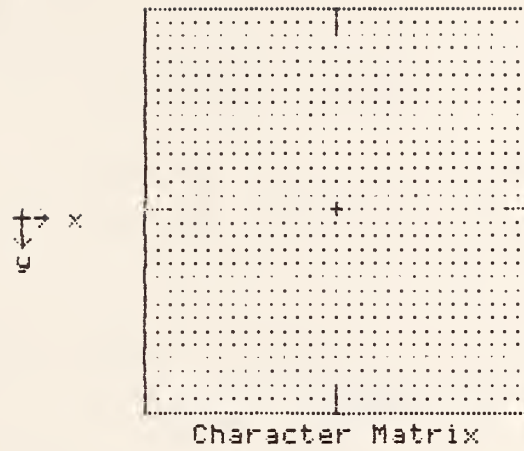
]
```

Character Matrix

Carto A : Hershey Char No.1

Fig.1. Cartographic letter A showing coordinate system.

Gothic N : Hershey Char. No.3514

Fig.2. Hires display of English gothic N.

± ∓ × · ÷ ≠ ≡ < > ≤

≧ ∝ ~ ⌢ √ [ ] → ↑ ←

↓ ∂ ∇ § ∫ ∮ ∞ } } †

‡ ∈

Math Symbols

… ! " # $ % & ' ( )

* + , − . / 0 1 2 3

4 5 6 7 8 9 : ; < =

> ?

Carto Numbers

□ a b c d e f g h i

j k l m n o p q r s

t u v w x y з · ' {

} ,

Carto Small

□ A B C D E F G H I

J K L M N O P Q R S

T U V W X Y Z Φ Ψ ±

∫ Ω

Carto Caps

Fig.3. Math symbols and Carto fonts.

33

─ ! " # $ % & ' ( )

* + , − . / 0 1 2 3

4 5 6 7 8 9 : ; < =

> ?

Index Numbers

° a b c d e f g h i

j k l m n o p q r s

t u v w x y z † ‡ {

} ★

Index Small

@ A B C D E F G H I

J K L M N O P Q R S

T U V W X Y Z Ω ℧ ±

∫ ☿

Index Caps

Ω α β γ δ ε ζ η ϑ ι

κ λ μ ν ξ ο π ρ σ τ

υ φ χ ψ ω Σ Π Δ Λ Γ

Θ Ξ

Index Greek

Fig. 4. Index font.

------ ! " # $ % & ' ( )
* + , – . / 0 1 2 3
4 5 6 7 8 9 : ; < =
> ?

Simplex Numbers

♠ a b c d e f g h i
j k l m n o p q r s
t u v w x y z ♡ ◇ [
] ♣

Simplex Small

@ A B C D E F G H I
J K L M N O P Q R S
T U V W X Y Z ' ` ±
∫ ∞

Simplex Caps

Ω α β γ δ ε ζ η ϑ ι
κ λ μ ν ξ ο π ρ σ τ
υ φ χ ψ ω Σ Π Δ Υ Γ
Θ Χ

Simplex Greek

Fig. 5. Simplex font.

_ ! " # $ % & ' ( )
* + , — . / 0 1 2 3
4 5 6 7 8 9 : ; < =
> ?

Complex Numbers

° a b c d e f g h i
j k l m n o p q r s
t u v w x y z † ‡ }
} ℵ

Complex Small

@ A B C D E F G H I
J K L M N O P Q R S
T U V W X Y Z C R 1982 ◈

Complex Caps

Ω α β γ δ ε ζ η ϑ ι
κ λ μ ν ξ ο π ρ σ τ
υ φ χ ψ ω Σ Π Δ Ξ Γ
Θ X

Complex Greek

Fig. 6. Complex font.

─ ! " # $ % & ' ( )
* + , ─ . / 0 1 2 3
4 5 6 7 8 9 : ; < =
> ?

Gothic Numbers

° a b c d e f g h i
j k l m n o p q r s
t u v w x y 3 ─ ─ }
} ─

Gothic Small

@ A B C D E F G H I
J K L M N O P Q R S
T U V W X Y Z ─ ─ ±
∫

Gothic Caps

Fig. 7. Gothic font.

_ ! " # $ % & ' ( )
* + , — . / 0 1 2 3
4 5 6 7 8 9 : ; < =
> ?

Script Numbers

a b c d e f g h i
j k l m n o p q r s
t u v w x y z ' }
}

Script Small

@ A B C D E F G H I
J K L M N O P Q R S
T U V W X Y Z ±
∫

Script Caps

Fig. 8. Script font.

38

Candy Apple
C. E. Dick
&
J. Hilsenrath
Hershey Characters
for the Apple II

Fig.9.   Candy Apple logo produced by Candy Wrapper.

U.S. DEPT. OF COMM. **BIBLIOGRAPHIC DATA** **SHEET** (See instructions)	1. PUBLICATION OR REPORT NO. NBS TN 1176	2. Performing Organ. Report No.	3. Publication Date June 1983

4. TITLE AND SUBTITLE

Utility Programs for Generating the Hershey Character Fonts on
Microcomputers and Laboratory Plotters

5. AUTHOR(S)

C. E. Dick and J. Hilsenrath

6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions) **NATIONAL BUREAU OF STANDARDS** **DEPARTMENT OF COMMERCE** **WASHINGTON, D.C. 20234**	7. Contract/Grant No.
	8. Type of Report & Period Covered Final

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP)

Same as in item 6 above.

10. SUPPLEMENTARY NOTES

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

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

Two programs are described that allow for the storage and manipulation of digitized fonts of graphic arts symbols and characters on the Apple II computer system. These fonts are based on the work of A. V. Hershey and provide the user with a repertory of digitized characters suitable for use in the preparation of camera-ready illustration in the laboratory environment. The programs described contain routines for reading files of the Hershey coordinates, storing them as text files, displaying them as individual characters or combined text on the high resolution display, and two methods for editing them or creating special symbols and graphics.

12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)

Applesoft basic programs; camera-ready illustrations; digitized graphic symbols; Hershey character fonts; microcomputers.

13. AVAILABILITY ☒ Unlimited ☐ For Official Distribution. Do Not Release to NTIS ☒ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402. ☐ Order From National Technical Information Service (NTIS), Springfield, VA. 22161	14. NO. OF PRINTED PAGES 45
	15. Price $3.75

NBS TECHNICAL PUBLICATIONS

PERIODICALS

JOURNAL OF RESEARCH—The Journal of Research of the National Bureau of Standards reports NBS research and development in those disciplines of the physical and engineering sciences in which the Bureau is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Bureau's technical and scientific programs. As a special service to subscribers each issue contains complete citations to all recent Bureau publications in both NBS and non-NBS media. Issued six times a year. Annual subscription: domestic $18; foreign $22.50. Single copy, $5.50 domestic; $6.90 foreign.

NONPERIODICALS

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

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

Special Publications—Include proceedings of conferences sponsored by NBS, NBS annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

Applied Mathematics Series—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NBS under the authority of the National Standard Data Act (Public Law 90-396).

NOTE: The principal publication outlet for the foregoing data is the Journal of Physical and Chemical Reference Data (JPCRD) published quarterly for NBS by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements available from ACS, 1155 Sixteenth St., NW, Washington, DC 20056.

Building Science Series—Disseminates technical information developed at the Bureau on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NBS under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NBS administers this program as a supplement to the activities of the private sector standardizing organizations.

Consumer Information Series—Practical information, based on NBS research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

Order the above NBS publications from: Superintendent of Documents, Government Printing Office, Washington, DC 20402.

Order the following NBS publications—FIPS and NBSIR's—from the National Technical Information Service, Springfield, VA 22161.

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NBS pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NBS Interagency Reports (NBSIR)—A special series of interim or final reports on work performed by NBS for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Service, Springfield, VA 22161, in paper copy or microfiche form.