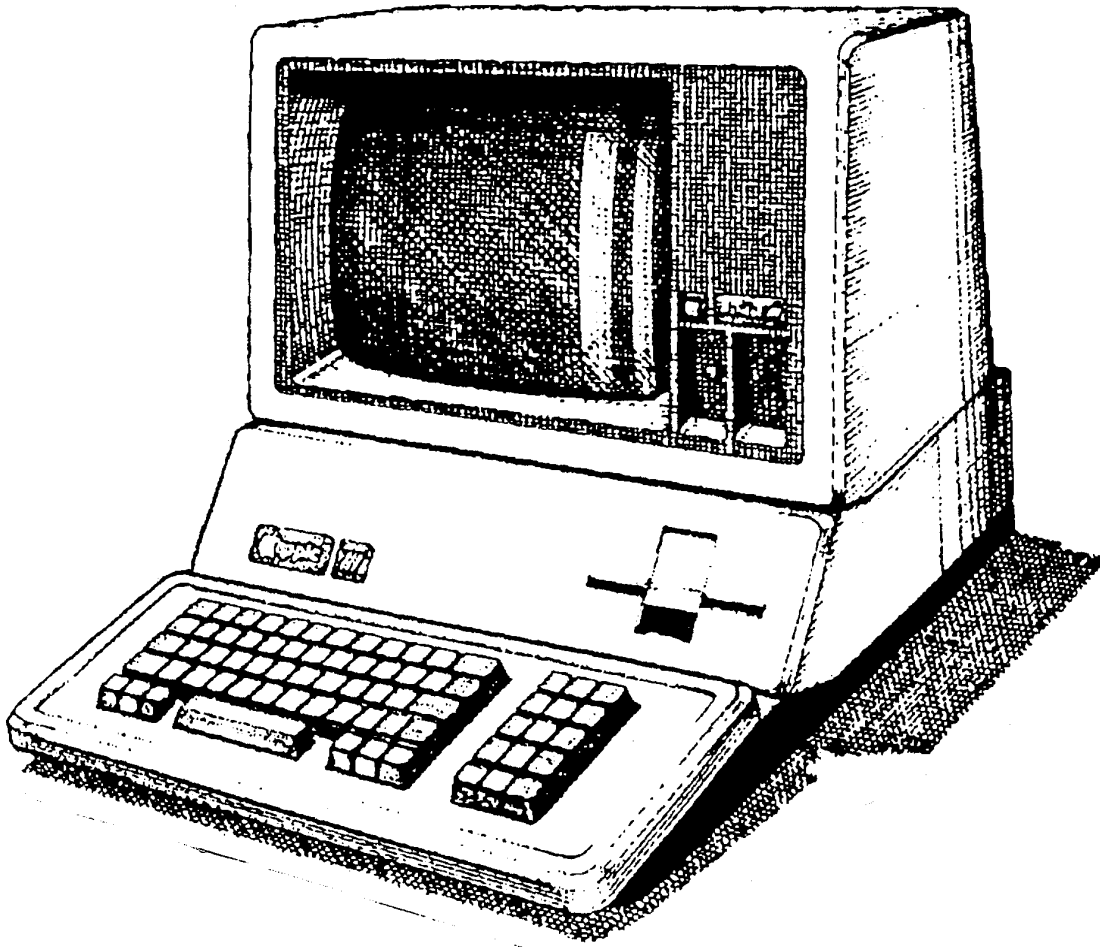




Apple /// Computer Information



DOCUMENT NAME	#
MANDEL-BROT SET PLOTTING PROGRAM FOR APPLE III (BUSINESS BASIC)	97

Ex Libris David T. Craig

2 pages

4/11/90 9:04 AM

Source Disk:a3.MANDELBROT.TEXT

Page 1

```

1  10-Nov-85                Subject: Mandelbrot Sets                From: Russ Altman
2
3  I have uploaded the basic program for native /// mode that will produce
4  Mandelbrot Sets in color 190 * 140 mode.  The description reminds you to do
5  >perform grafixon before running the program, so you can watch its progress.
6  It takes on the order of an hour + to finish a picture, so it is a very good
7  idea to turn off the display refresh (CONT-KEYPAD-5) to speed up the execution
8  (you can put it back on every now and then to admire the picture as it is
9  built).
10
11  If it does not work, please let me know SOON, since it worked before I
12  uploaded it, and I have used it a number of times.  However, I must admit
13  that I added some comments from a text editor, and if I did something stupid,
14  there might be problems.
15
16  A key thing to realize about the program is that by changing the parameters
17  xhi, xlo, yhi, ylo, you can "zoom in" on sections of the image that may be
18  interesting.  There are a lot of interesting patterns at the border which
19  deserve close scrutiny.  Enough said, let me know how it goes... Russ
20
21
22  1  REM *****  MANDELBROT SET PLOTTING PROGRAM FOR APPLE ///  *****
23  2  REM      Program requires "BGRAF.INV" takes 1-2 hours to complete
24  3  REM      an image (See Scientific American, October 1985).
25  4  REM      Please forgive bad basic form, I don't usually program
26  5  REM      in this language.      Russ B. Altman 11/5/85
27  6  REM XLO,XHI,YLO,YHI are boundaries of the box to be drawn on screen
28  7  REM Best starting values are -2,2 for both sets. Can change to
29  8  REM get closeups of certain areas.
30  9  INVOKE ".D1/BGRAF.INV":REM Change this path depending on where you keep
31 10  REM      your Basic invokables.
32 11  XLO=-2
33 12  XHI=2
34 13  YLO=-2
35 14  YHI=2
36 15  REM XMAX = Maximum number of points in X direction
37 16  REM YMAX = Maximum number of points in Y direction
38 17  XMAX=139
39 18  YMAX=191
40 19  REM XMAX and YMAX above are resolution of screen
41 20  REM Now compute the sample size DELTAX, DELTAY
42 21  DELTAX=(XHI-XLO)/XMAX
43 22  DELTAY=(YHI-YLO)/YMAX
44 23  PRINT DELTAX,DELTAY
45 24  REM Set max to be the maximum number of colors you can draw at each dot
46 25  MAX%=15
47 26  hits%=0
48 27  REM Now call all the functions to initialize graphics, clear screen
49 28  REM the ones below are for Apple ///
50 29  PERFORM INITGRAFIX
51 30  PERFORM GRAFIXMODE(%3,%1)
52 31  PERFORM fillcolor(%0)
53 32  PERFORM FILLPORT
54 33  REM Now just loop through all values of I,J and calculate corresponding
55 34  REM values of X, and Y, then GOTO the subroutine that calculates ANS.
56 35  REM Depending on what ANS is, either draw a dot of color ANS or not.
57 36  FOR I%=1 TO INT(XMAX)
58 37  PRINT"I = ",I%
59 38  FOR J%=1 TO INT(YMAX)
60 39  GOSUB 300
61 40  REM If ANS not equal to 0, then draw a dot at the point I,J
62 41  IF ANS<>0 THEN GOSUB 230
63 42  ANS=0
64 43  NEXT J%
65 44  NEXT I%
66 45  STOP
67 46  REM This is the first subroutine which just draws a dot at
68 47  REM point I,J of color ANS.  make sure arguments to functions

```

BUSINESS BASIC

4/11/90 9:04 AM

Source Disk:a3.MANDELBROT.TEXT

Page 2

```
69 227 REM are right data type (string, int, real, etc...)
70 230 XINT%=I%
71 240 YINT%=J%
72 245 REM Make sure we have an integer for color
73 250 A%=INT(ANS)
74 255 REM PENCOLOR sets the color of the dot to be drawn (on Apple ///)
75 256 REM DOTAT actually draws a dot at XINT,YINT = I,J
76 260 PERFORM PENCOLOR(%A%)
77 270 PERFORM DOTAT(%XINT%,%YINT%)
78 280 HITS%=HITS%+1
79 281 REM HITS just keeps track of how many non-zero dots--expendable
80 290 RETURN
81 295 REM this subroutine actually calculates ANS, or how long it takes
82 296 REM for a number to be squared and reach a limit (4)
83 300 X=I%*DELTAX+XLO
84 310 Y=J%*DELTAY+YLO
85 320 FIRSTX=X
86 330 FIRSY=Y
87 335 REM N is a counter for how many times we have calc'ed MOD2
88 340 N=0
89 350 N=N+1
90 360 IF N>MAX% GOTO 440
91 364 REM If MOD2 > 4 then we are done, so leave subroutine
92 365 REM else, we want to calculate a new set of X,Y'S and recalc MOD2
93 370 MOD2=X*X+Y*Y
94 380 IF MOD2>=4 GOTO 460
95 390 oldx=x
96 400 oldy=y
97 410 x=oldx*oldx-oldy*oldy+firstx
98 420 y=2*oldx*oldy+firsy
99 425 REM Loop back up and try calc'ing MOD2 again to see if reached limit
100 430 GOTO 350
101 433 REM If we haven't reached MOD2 by the time we have used up all
102 434 REM possible colors, then we just draw black (return ANS = 0)
103 440 ANS=0
104 450 RETURN
105 455 REM We set ANS to N, so that we know what color to draw. Note that
106 456 REM N and ANS refer to the number of times we calced MOD2 *and*
107 457 REM the color we are going to use to plot on the screen.
108 460 ANS=N
109 470 RETURN
110 480 REM End of program Mandelbrot.Bas
111
112 finis
```