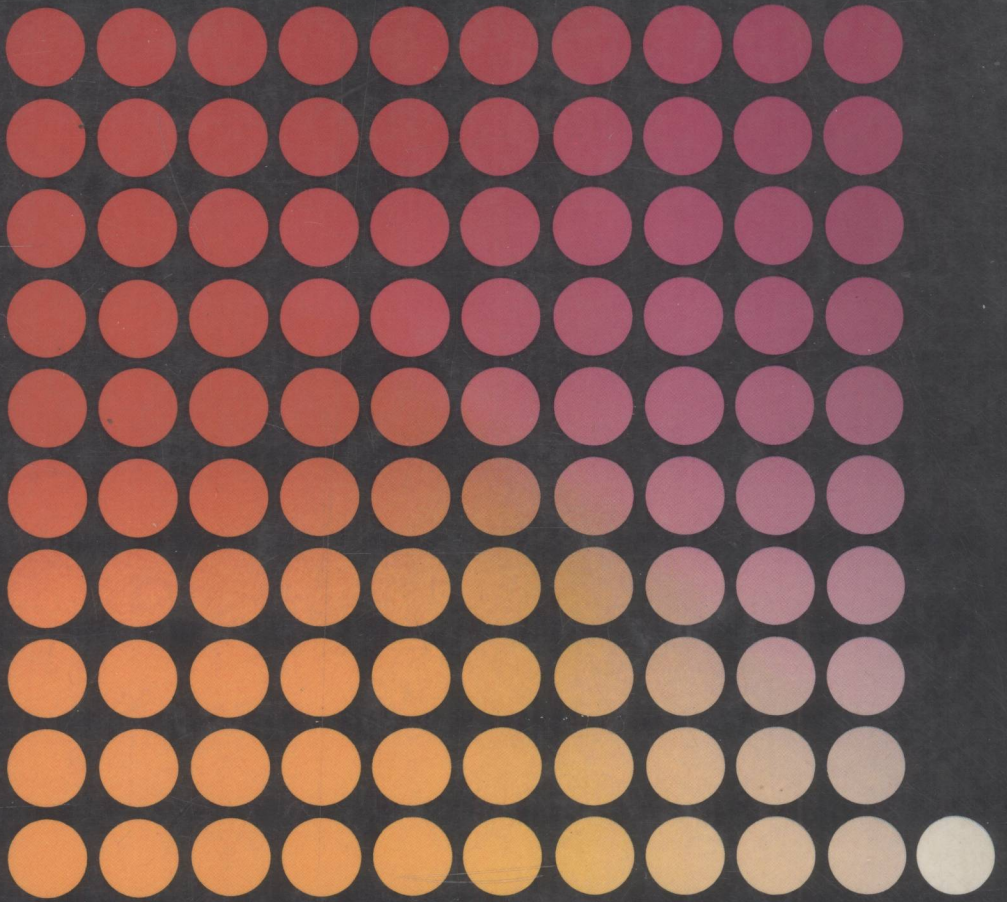


1940



101 Ready-to-Run Programs
and Subroutines for the
IBM PCjr.[®]
JEFF BRETZ & JOHN CLARK CRAIG

TP31
B30

8563740

101 Ready-to-Run Programs and Subroutines for the **IBM PCjr.**[®]

JEFF BRETZ & JOHN CLARK CRAIG



E8563740



TAB BOOKS Inc.

Blue Ridge Summit, PA 17214

Also by the Authors from TAB BOOKS Inc.

- No. 1350 *119 Practical Programs for the TRS-80® Pocket Computer*
No. 1540 *100 Ready-to-Run Programs and Subroutines for the IBM PC® Computer*
No. 1860 *IBM PC® Graphics*

Epson is a trademark of Epson America, Inc.
TRS-80 is a trademark of the Radio Shack Division of Tandy Corporation.
Microsoft is a trademark of Microsoft Corporation.
IBM PCjr. is a registered trademark of International Business Machines.

FIRST EDITION

FIRST PRINTING

Copyright © 1985 by TAB BOOKS Inc.
Printed in the United States of America

Reproduction or publication of the content in any manner, without express permission of the publisher, is prohibited. No liability is assumed with respect to the use of the information herein.

Library of Congress Cataloging in Publication Data

Bretz, Jeff.

101 ready-to-run programs and subroutines for the
IBM PCjr.

Includes index.

1. IBM PCjr (Computer)—Programming. 2. Computer programs. 3. Subroutines (Computer programs) I. Craig, John Clark. II. Title. III. Title: One hundred and one ready-to-run programs and subroutines for the IBM PCjr. IV. Title: One hundred one ready-to-run programs and subroutines for the IBM PCjr.

QA76.8.I2593B74 1984 001.64'2 84-8547

ISBN 0-8306-0940-7

ISBN 0-8306-1940-2 (pbk.)

Introduction

Most of the programs in this book are ready to read and run, but our main goal in writing this book was to provide a source of creative programming techniques. Every program demonstrates a technique, a trick, or a concept unique from the others in this book. For example, you'll find a wide variety of data input techniques in these programs. The `INKEY$`, `input`, `line input`, and other functions are used in several different ways to demonstrate the wide range of possible programming methods.

The programs have been written in an open, readable style so that you can study the techniques presented. If your IBM PCjr has limited memory, you might consider deleting the remark lines and restructuring the program lines by putting multiple statements on the same line. You'll find these programs easier to read and understand than many programs written in BASIC, partly because we were careful to open up the listings as described, and partly because of the excellent BASIC that IBM chose to use in their PCjr. This version of Microsoft BASIC has several outstanding features, but the feature most important to producing highly comprehensible program listings is the option that allows long variable names. For example, a variable named `"DATA.POINTER"` is much more self documenting than `"P2."` Using longer variable names involves a little extra typing, but it's a habit worth developing. Your programs will be neater, easier to read, and by far easier to understand.

We had a lot of fun developing these programs. The IBM PCjr is truly an outstanding product and is a joy to work with. We hope that these programs will help you enjoy your computer as much as we have enjoyed ours.



Contents

| | |
|--|------------|
| Introduction | vi |
| 1 Calendars, Clocks, and Time | 1 |
| Calendar—Clock—Sidereal Clock | |
| 2 Educational and Informative | 19 |
| Code Master—Frequency—Geosynchronous Satellite Antenna Aim—Ohm's Law—Power Calculator—Spelling Helper | |
| 3 Games and Other Fun | 45 |
| Amaze Creator—Biorhythms—Chomper—Fly Elimination Game—De-Jumble—Out on a LEM—Newsroom Printer—Tone Match Game—Trax Game—Watchout Game | |
| 4 Graphics Techniques and Demonstrations | 87 |
| High-Resolution Graphics—Imaginary Rods—Designs—Graphics Demonstrations—Beams—Kaleidos—Painting—Sphere—Spokes—Star—Tunnel—Wand | |
| 5 Mathematics | 137 |
| Complex Numbers—Fractions—Function Analysis—Plot 3-D—Prime Numbers—Simultaneous Equations—Double-Precision Sine and Cosine Routines—Vectors | |
| 6 Modem Communications | 177 |
| Communications Transfer Programs: Transmit and Receive | |
| 7 Programming Aids, Subroutines, and Utilities | 187 |
| Personalized DOS—Letter Writer—Circles—Timed Delays—Bare-Bones Text Creation—Automatic Startup Procedure—Color Monitor Alignment—Data-In—Memory Dump Routines—Filesort—Font-80—Message Center—New-Font—Printout—Rotate-a-Font—Rotate-a-Letter—Subroutines and Functions—XREF (Cross Reference) | |
| 8 Programs for a Small Business | 242 |
| Checkbook Master—Graph Master—Mail—Text—Textjr | |
| Index | 307 |

Chapter 1



Calendars, Clocks, and Time

The programs in this chapter use the graphics abilities of the IBM PCjr and its internal clock to create three useful programs dealing with the fourth dimension—time.

CALENDAR

The *Calendar* program demonstrates the use of several calendar-related subroutines that you may find useful. At the heart of the program are subroutines for finding the astronomical Julian day number for a given date, and the date for a given astronomical Julian day number. Both of these subroutines also find the day of the week. The span of dates that may be used ranges from the year 1582 to the year 3999.

```
100 *****
110 **          Calendar Functions          **
120 **
130 **          FILE      CALENDAR.BAS      **
140 **          VERSION  PCjr 1.00          **
150 **          DATE      March 12, 1984    **
160 **          AUTHORS   John Craig        **
170 **          Jeff Bretz                   **
180 *****
190 '
200 '
210 ' Initialization
220 CLEAR
230 SCREEN 1
240 SCREEN 0,1,0,0
```

```

250 COLOR 14,1,1
260 WIDTH 40
270 CLS
280 KEY OFF
290 OPTION BASE 1
300 DIM MONTH.NAME$(12),WEEK.DAY$(7)
310 FOR I = 1 TO 12
320 READ MONTH.NAME$(I)
330 NEXT I
340 DATA JANUARY,FEBRUARY,MARCH,APRIL,MAY,JUNE,JULY
350 DATA AUGUST,SEPTEMBER,OCTOBER,NOVEMBER,DECEMBER
360 FOR I = 1 TO 7
370 READ WEEK.DAY$(I)
380 NEXT I
390 DATA SUNDAY,MONDAY,TUESDAY,WEDNESDAY,THURSDAY,FRIDAY,SATURDAY
400 '
410 ' Draw the main menu screen
420 MENUFLAG = 0
430 LOCATE 1,9
440 PRINT "* * * CALENDAR * * *"
450 LOCATE 7,1
460 PRINT "F1. Sketch a one month calendar page
470 PRINT
480 PRINT "F2. Describe a given date
490 PRINT
500 PRINT "F3. Number of days between two dates
510 PRINT
520 PRINT "F4. Quit
530 LOCATE 25,5
540 PRINT "PRESS A SPECIAL FUNCTION KEY";
550 ON KEY(1) GOSUB 710
560 ON KEY(2) GOSUB 1500
570 ON KEY(3) GOSUB 1960
580 ON KEY(4) GOSUB 2370
590 KEY(1) ON
600 KEY(2) ON
610 KEY(3) ON
620 KEY(4) ON
630 '
640 WHILE QUIT = NOT.YET
650 KEY.BUFFER.CLEAR$ = INKEY$
660 IF MENUFLAG THEN 410
670 WEND
680 CLS
690 END
700 '
710 ' F1 Subroutine, sketch a month
720 CLS
730 LOCATE 7,10
740 INPUT "What month ";Q$
750 IF Q$ = "" THEN 1460
760 GOSUB 2510

```

```

770 GOSUB 2580
780 MONTH = VAL(Q$)
790 IF MONTH THEN 830
800 FOR I = 1 TO 12
810 IF LEFT$(MONTH.NAME$(I),3) = LEFT$(Q$,3) THEN MONTH = I
820 NEXT I
830 IF MONTH >= 1 AND MONTH <= 13 THEN 890
840 LOCATE 8,1
850 PRINT "I don't recognize the month you"
860 PRINT "entered ... try again."
870 BEEP
880 GOTO 730
890 LOCATE 8,1
900 PRINT SPACE$(40);
910 PRINT SPACE$(40);
920 LOCATE 9,10
930 INPUT "What year ";Q$
940 IF Q$ = "" THEN 1460
950 YEAR = VAL(Q$)
960 IF YEAR THEN 1020
970 LOCATE 10,1
980 PRINT "I don't recognize the year you"
990 PRINT "entered ... try again."
1000 BEEP
1010 GOTO 920
1020 IF YEAR < 100 THEN YEAR = YEAR + 1900
1030 IF YEAR > 1581 AND YEAR < 4000 THEN 1080
1040 PRINT "The year must be in the range"
1050 PRINT "1582 to 3999 ... try again."
1060 BEEP
1070 GOTO 920
1080 DAY = 1
1090 GOSUB 2660
1100 DAYOFWEEK = WEEKDAY
1110 TITLE$ = MONTH.NAME$(MONTH)
1120 JFIRST = JULIAN
1130 MONTH = MONTH + 1
1140 IF MONTH > 12 THEN MONTH = 1
1150 IF MONTH = 1 THEN YEAR = YEAR + 1
1160 GOSUB 2660
1170 MONTHDAYS = JULIAN - JFIRST
1180 CLS
1190 LOCATE 1,17 - LEN(TITLE$) / 2
1200 PRINT TITLE$ ; YEAR + (MONTH = 1)
1210 DATE = 1
1220 ROW = 6
1230 COL = DAYOFWEEK * 4 + 2
1240 LOCATE ROW,COL - (DATE < 10)
1250 PRINT DATE
1260 DATE = DATE + 1
1270 IF DATE > MONTHDAYS THEN 1310
1280 DAYOFWEEK = DAYOFWEEK MOD 7 + 1

```



```

1290 IF DAYOFWEEK = 1 THEN ROW = ROW + 3
1300 GOTO 1230
1310 FOR ROWLINE = 4 TO ROW + 3 STEP 3
1320 LOCATE ROWLINE,6
1330 PRINT STRING$(28," ");
1340 NEXT ROWLINE
1350 FOR ROW2 = 4 TO ROW + 1
1360 FOR COL2 = 5 TO 33 STEP 4
1370 LOCATE ROW2,COL2
1380 IF ROW2 = 4 THEN PRINT " "; ELSE PRINT "!";
1390 NEXT COL2,ROW2
1400 FOR I = 1 TO 7
1410 LOCATE 3,4 * I + 2
1420 PRINT LEFT$(WEEK.DAY$(I),3);
1430 NEXT I
1440 BARMESS = 1
1450 GOSUB 2430
1460 CLS
1470 MENUFLAG = 1
1480 RETURN
1490 '
1500 ' F2 Subroutine, describe a date
1510 CLS
1520 LOCATE 7,1
1530 PRINT "Enter a date ..."
1540 LINE INPUT "(any reasonable format) ";CAL$
1550 IF CAL$ = "" THEN 1920
1560 GOSUB 2960
1570 IF YEAR THEN 1630
1580 PRINT
1590 PRINT "Your date is unrecognizable,"
1600 PRINT "or isn't a valid date ... try again."
1610 BEEP
1620 GOTO 1520
1630 CLS
1640 LOCATE 5,1
1650 BS$ = CHR$(29)
1660 PRINT MONTH;" / ";DAY;" / ";YEAR;"can also be written as "
1670 PRINT MONTH.NAME$(MONTH);DAY;BS$;",";YEAR;BS$;". "
1680 PRINT
1690 PRINT "The day of the week is ";WEEK.DAY$(WEEKDAY);"."
1700 IF YEAR < 1600 THEN 1880
1710 JULIAN2 = JULIAN
1720 MONTH2 = MONTH
1730 DAY2 = DAY
1740 YEAR2 = YEAR
1750 MONTH = 12
1760 DAY = 31
1770 YEAR = YEAR - 1
1780 IF YEAR < 1582 THEN 1830
1790 GOSUB 2660
1800 YEARDAY = JULIAN2 - JULIAN

```

```

1810 PRINT
1820 PRINT "It is day number"YEARDAY"of"YEAR2;BS$;". "
1830 YEAR = (INT(YEAR/100) - 1) * 100 + 99
1840 GOSUB 2660
1850 CENTDAY = JULIAN2 - JULIAN
1860 PRINT
1870 PRINT "It is the"CENTDAY"day of the century.
1880 PRINT
1890 PRINT "And the astronomical julian"
1900 PRINT "day number is";JULIAN2;BS$;". "
1910 GOSUB 2430
1920 CLS
1930 MENUFLAG = 1
1940 RETURN
1950 '
1960 ' F3 Subroutine, days between dates
1970 CLS
1980 LOCATE 7,1
1990 PRINT "Enter one date"
2000 LINE INPUT "(any reasonable format) ";CAL$
2010 IF CAL$ = "" THEN 2330
2020 GOSUB 2960
2030 IF YEAR THEN 2090
2040 PRINT
2050 PRINT "Your date is unrecognizable,"
2060 PRINT "or isn't a valid date ... try again."
2070 BEEP
2080 GOTO 1980
2090 MONTH3 = MONTH
2100 DAY3 = DAY
2110 YEAR3 = YEAR
2120 JULIAN3 = JULIAN
2130 LOCATE 10,1
2140 PRINT SPACE$(40);
2150 PRINT SPACE$(40);
2160 LOCATE 10,1
2170 LINE INPUT "Enter second date ... ";CAL$
2180 IF CAL$ = "" THEN 2330
2190 GOSUB 2960
2200 IF YEAR THEN 2260
2210 PRINT
2220 PRINT "Your date is unrecognizable,"
2230 PRINT "or isn't a valid date ... try again."
2240 BEEP
2250 GOTO 2160
2260 NUMDAYS = ABS(JULIAN3 - JULIAN)
2270 CLS
2280 LOCATE 7,1
2290 PRINT "Between";MONTH3;"/";DAY3;"/";YEAR3
2300 PRINT "and      ";MONTH;"/";DAY;"/";YEAR
2310 PRINT "there are";NUMDAYS;"days."
2320 GOSUB 2430

```

```

2330 CLS
2340 MENUFLAG = 1
2350 RETURN
2360 '
2370 ' F4 Subroutine, set quit flag
2380 QUIT = 1
2390 RETURN
2400 '
2410 '
2420 ' Subroutine, wait for user before proceeding
2430 LOCATE 25,8
2440 IF BARMESS = 0 THEN PRINT "PRESS ANY KEY TO PROCEED";
2450 K$ = INKEY$
2460 IF K$ = "" THEN 2450
2470 BARMESS = 0
2480 RETURN
2490 '
2500 ' Subroutine, de-space Q$
2510 SP = INSTR(Q$," ")
2520 IF SP = 0 THEN 2580
2530 Q$ = LEFT$(Q$,SP-1) + MID$(Q$,SP+1)
2540 GOTO 2510
2550 RETURN
2560 '
2570 ' Subroutine, just capitalize Q$
2580 FOR QP = 1 TO LEN(Q$)
2590 CHAR$ = MID$(Q$,QP,1)
2600 IF CHAR$ < "a" OR CHAR$ > "z" THEN 2620
2610 MID$(Q$,QP,1) = CHR$(ASC(CHAR$)-32)
2620 NEXT QP
2630 RETURN
2640 '
2650 ' Subroutine, MONTH,DAY,YEAR to JULIAN,WEEKDAY
2660 JULIAN = INT(365.2422# * YEAR + 30.44 * (MONTH-1) + DAY + 1)
2670 T1 = MONTH - 2 - 12 * (MONTH < 3)
2680 T2 = YEAR + (MONTH < 3)
2690 T3 = INT(T2 / 100)
2700 T2 = T2 - 100 * T3
2710 WEEKDAY = INT(2.61 * T1 - .2) + DAY + T2 + INT(T2 / 4)
2720 WEEKDAY = (WEEKDAY + INT(T3 / 4) - T3 - T3 + 77) MOD 7 + 1
2730 T4 = JULIAN - 7 * INT(JULIAN / 7)
2740 JULIAN = JULIAN - T4 + WEEKDAY + 7 * (T4 < WEEKDAY - 1) + 1721060#
2750 RETURN
2760 '
2770 ' Subroutine, JULIAN to MONTH,DAY,YEAR,WEEKDAY
2780 T5 = JULIAN
2790 YEAR = INT((JULIAN - 1721061!) / 365.25 + 1)
2800 MONTH = 1
2810 DAY = 1
2820 GOSUB 2660
2830 IF JULIAN <= T5 THEN 2860

```

```

2840 YEAR = YEAR - 1
2850 GOTO 2820
2860 MONTH = INT((T5 - JULIAN) / 29 + 1)
2870 GOSUB 2660
2880 IF JULIAN <= T5 THEN 2910
2890 MONTH = MONTH - 1
2900 GOTO 2870
2910 DAY = T5 - JULIAN + 1
2920 GOSUB 2660
2930 RETURN
2940 '
2950 ' Subroutine, convert CAL$ to MONTH,DAY,YEAR
2960 Q$ = CAL$
2970 GOSUB 2580
2980 CAL$ = Q$
2990 MONTH = 0
3000 DAY = 0
3010 YEAR = 0
3020 FOR I = 1 TO 12
3030 IF INSTR(CAL$,LEFT$(MONTH.NAME$(I),3)) THEN MONTH = I
3040 NEXT I
3050 FOR I = 1 TO LEN(CAL$)
3060 CHAR$ = MID$(CAL$,I,1)
3070 IF CHAR$ < "0" OR CHAR$ > "9" THEN MID$(CAL$,I,1) = ":"
3080 NEXT I
3090 IF INSTR(CAL$,":") THEN 3150
3100 IF LEN(CAL$) <> 6 AND LEN(CAL$) <> 8 THEN 3400
3110 MONTH = VAL(LEFT$(CAL$,2))
3120 DAY = VAL(MID$(CAL$,3,2))
3130 YEAR = VAL(MID$(CAL$,5))
3140 GOTO 3290
3150 VFLAG = 0
3160 FOR I = 1 TO LEN(CAL$)
3170 CALVAL = VAL(MID$(CAL$,I))
3180 IF CALVAL = 0 THEN VFLAG = 0
3190 IF CALVAL = 0 OR VFLAG = 1 THEN 3280
3200 IF MONTH THEN 3230
3210 MONTH = CALVAL
3220 GOTO 3270
3230 IF DAY THEN 3260
3240 DAY = CALVAL
3250 GOTO 3270
3260 YEAR = CALVAL
3270 VFLAG = 1
3280 NEXT I
3290 IF YEAR < 100 AND YEAR > 0 THEN YEAR = YEAR + 1900
3300 IF YEAR < 1582 OR YEAR > 3999 THEN YEAR = 0
3310 IF YEAR = 0 THEN 3400
3320 MONTH2 = MONTH
3330 DAY2 = DAY
3340 YEAR2 = YEAR
3350 GOSUB 2660

```

```

3360 GOSUB 2780
3370 IF MONTH2 <> MONTH THEN YEAR = 0
3380 IF DAY2 <> DAY THEN YEAR = 0
3390 IF YEAR2 <> YEAR THEN YEAR = 0
3400 RETURN

```

Three computations are available and are selected by pressing the appropriate special function key. Press F1 to generate a one-month calendar sheet. Press F2 to compute the day of the week and other facts about a given date. Press F3 to find the number of days between two dates. When you are finished, press F4 to quit.

You may type in the dates in just about any format you choose. A subroutine is provided that analyzes the entered date and figures out what month, day, and year it represents. Here are a few date entries that the subroutine can interpret correctly.

| | |
|--------------|------------|
| July 4, 1776 | 4 JUL 1776 |
| 7/4/76 | 07041776 |

Note that an entry of "7/4/76" is interpreted as July 4, 1976. If the century is not indicated, it is assumed that you mean the twentieth century.

A useful feature of these subroutines is their ability to check a date to see if it is real. The date is first converted to its Julian number, and then back to a date. The date is valid if the result matches the original date. Figures 1-1 through 1-4 show some of the displays that will appear on-screen when you run the *Calendar* program.

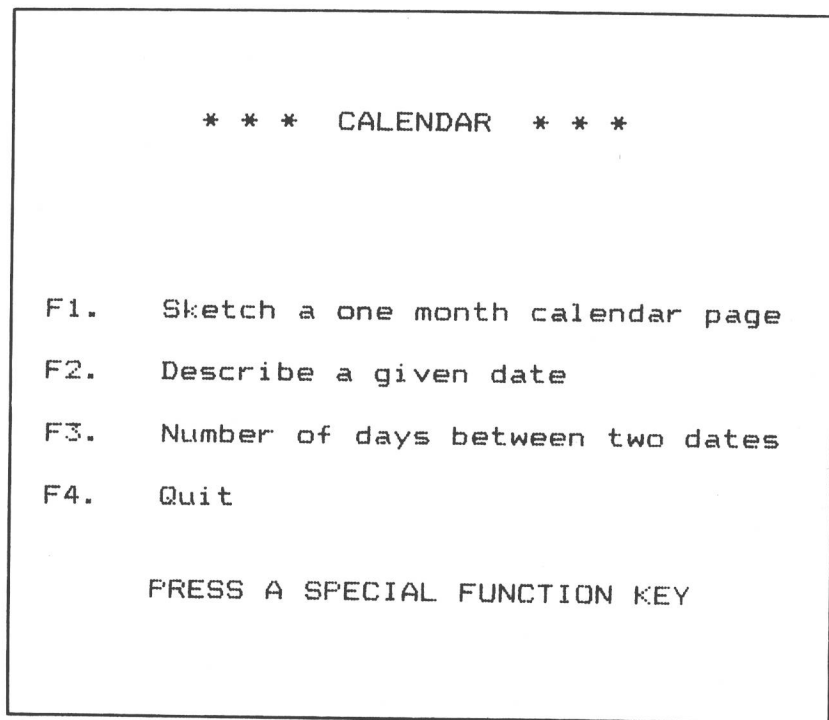


Fig. 1-1. The options you may use in the *Calendar* program.

Fig. 1-2. The calendar display from the *Calendar* program.

| JULY 1776 | | | | | | |
|-----------|-----|-----|-----|-----|-----|-----|
| SUN | MON | TUE | WED | THU | FRI | SAT |
| | 1 | 2 | 3 | 4 | 5 | 6 |
| 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 |
| 28 | 29 | 30 | 31 | | | |

12 / 25 / 1985 can also be written as
DECEMBER 25, 1985.

The day of the week is WEDNESDAY.

It is day number 359 of 1985.

It is the 31405 day of the century.

And the astronomical julian
day number is 2446425.

PRESS ANY KEY TO PROCEED

Fig. 1-3. Sample results of the *Calendar* program.

Fig. 1-4. Sample results of the option F3 in the *Calendar* program.

Between 7 / 4 / 1776
and 7 / 4 / 1985
there are 76335 days.

PRESS ANY KEY TO PROCEED

CLOCK

Here's a creative way to set the clock hidden inside your IBM PCjr! The most straightforward method of setting the time is by typing in a value for the "variable" TIME\$. A much more impressive method is demonstrated by this program!

```
100 '*****
110 '**          Clock                      **
120 '**          **                        **
130 '**          FILE      CLOCK.BAS      **
140 '**          VERSION PCjr 1.00        **
150 '**          DATE      March 14, 1984  **
160 '**          AUTHORS John Craig       **
170 '**          Jeff Bretz              **
180 '*****
190 '
200 '
210 ' Initialization
220 CLEAR
230 SCREEN 1
240 WINDOW SCREEN (0,0)-(639,199)
250 CLS
260 KEY OFF
270 DIM NUM(12),PLT%(32)
280 COLORFLAG = 1
290 '
300 ' Draw the clock
310 XCENT = 319
320 YCENT = 99
330 PI = 3.141593
340 DEF FNANG(TIME) = PI / 2 - PI * TIME / 30 - 2 * PI * (TIME > 15)
350 LINE (120,20)-(519,178),1,BF          ' draw big box
360 LINE (XCENT,0)-(0,YCENT),0           ' slice off four corners
370 LINE (XCENT,0)-(639,YCENT),0
380 LINE (XCENT,199)-(0,YCENT),0
390 LINE (XCENT,199)-(639,YCENT),0
400 PAINT (126,20),0                      ' paint out four corners
410 PAINT (512,175),0
420 PAINT (512,20),0
430 PAINT (126,175),0
440 PAINT (0,0),2,1                      ' alter background color
450 RADIUS = 179
460 CIRCLE (XCENT,YCENT),RADIUS,0         ' draw dark circle inside
470 PAINT (XCENT,YCENT),0                ' color entire circle
480 FOR D = 1 TO 12                      ' put numbers on clock face
490 GOSUB 1240
500 XD = XCENT + 165 * COS(FNANG(D*5)) - 22 + 3 * (D > 9)
510 YD = YCENT - 68 * SIN(FNANG(D*5)) - 3
520 PUT (XD,YD),NUM
530 NEXT D
540 LOCATE 1,1
550 PRINT "F1. Inc hour  ";
```

```

560 LOCATE 2,1
570 PRINT "F3. Inc minute";
580 LOCATE 3,1
590 PRINT "F5. Inc second";
600 LOCATE 1,27
610 PRINT "F2. Dec hour  ";
620 LOCATE 2,27
630 PRINT "F4. Dec minute";
640 LOCATE 3,27
650 PRINT "F6. Dec second";
660 LOCATE 25,3
670 PRINT "F7. Stop or start the color changing";
680 '
690 ' Activate the function keys
700 ON KEY(1) GOSUB 1300
710 ON KEY(2) GOSUB 1360
720 ON KEY(3) GOSUB 1420
730 ON KEY(4) GOSUB 1480
740 ON KEY(5) GOSUB 1540
750 ON KEY(6) GOSUB 1600
760 ON KEY(7) GOSUB 1660
770 KEY(1) ON
780 KEY(2) ON
790 KEY(3) ON
800 KEY(4) ON
810 KEY(5) ON
820 KEY(6) ON
830 KEY(7) ON
840 '
850 WHILE NOT YET.TO.THE.END.OF.ALL.TIME....
860     WHILE T$ = TIME$
870     WEND
880     T$ = TIME$
890     SECOND2 = SECOND
900     MINUTE2 = MINUTE
910     HOUR2 = HOUR
920     SECOND = VAL(RIGHT$(T$,2))
930     MINUTE = VAL(MID$(T$,4))
940     HOUR = (VAL(LEFT$(T$,2)) MOD 12) * 5 + MINUTE / 12
950     LOCATE 23,17
960     PRINT T$;
970     CIRCLE (XCEN1,YCEN1),140,3,-FNANG(SECOND),-FNANG(SECOND)
980     CIRCLE (XCEN1,YCEN1),140,0,-FNANG(SECOND2),-FNANG(SECOND2)
990     IF MINUTE2 = MINUTE THEN 1010
1000     CIRCLE (XCEN1,YCEN1),120,0,-FNANG(MINUTE2),-FNANG(MINUTE2)
1010     CIRCLE (XCEN1,YCEN1),120,3,-FNANG(MINUTE),-FNANG(MINUTE)
1020     IF HOUR2 = HOUR THEN 1040
1030     CIRCLE (XCEN1,YCEN1),70,0,-FNANG(HOUR2),-FNANG(HOUR2)
1040     CIRCLE (XCEN1,YCEN1),70,3,-FNANG(HOUR),-FNANG(HOUR)
1050     PTR = PTR MOD 16 + 1
1060     IF COLORFLAG = 0 THEN 1210
1070     PLT1 = INT(RND*16)

```



```

1080      PALETTE 1,PLT1
1090      PLT2 = INT(RND*16)
1100      IF PLT1 = PLT2 THEN 1090
1110      PALETTE 2,PLT2
1120      PLT3 = INT(RND*16)
1130      IF PLT3 = PLT1 THEN 1120
1140      IF PLT3 = PLT2 THEN 1120
1150      PALETTE 3,PLT3
1160      PLT0 = INT(RND*16)
1170      IF PLT0 = PLT1 THEN 1160
1180      IF PLT0 = PLT2 THEN 1160
1190      IF PLT0 = PLT3 THEN 1160
1200      PALETTE 0,PLT0
1210 WEND
1220 '
1230 ' Subroutine, get a number for putting anywhere
1240 LOCATE 1,1
1250 PRINT D;
1260 GET (0,0)-(44,6),NUM
1270 LINE (0,0)-(44,6),0,BF
1280 RETURN
1290 '
1300 ' Key 1 subroutine
1310 GOSUB 1710
1320 H = (H + 1) MOD 24
1330 GOSUB 1770
1340 RETURN
1350 '
1360 ' Key 2 subroutine
1370 GOSUB 1710
1380 H = (H + 23) MOD 24
1390 GOSUB 1770
1400 RETURN
1410 '
1420 ' Key 3 subroutine
1430 GOSUB 1710
1440 M = (M + 1) MOD 60
1450 GOSUB 1770
1460 RETURN
1470 '
1480 ' Key 4 subroutine
1490 TEMP$ = MID$(STR$((VAL(MID$(TIME$,4))+59)MOD 60),2)
1500 M = (M + 59) MOD 60
1510 GOSUB 1770
1520 RETURN
1530 '
1540 ' Key 5 subroutine
1550 GOSUB 1710
1560 S = (S + 1) MOD 60
1570 GOSUB 1770
1580 RETURN
1590 '

```