

```

0010:          VIDEO PLUS  TM
0020:          BASIC SOFTWARE
0030:
0040:          WRITTEN BY ROBERT M. TRIPP
0050:
0060:          COPYRIGHT (C) MAY 1979 BY
0070:          THE COMPUTERIST, INC
0080:          P.O. BOX 3
0090:          SO. CHELMSFORD, MA 01824
0100:
0110:          ALL RIGHTS RESERVED
0120:
0130:          PAGE ZERO LOCATIONS
0140:
0150: 23B0      CRTCON *      $0000  CRT CONTROLLER
0160: 23B0      CRTREG *      $0002  CRT REGISTERS
0170: 23B0      MEMLOW *      $0004  MEMORY LOW
0180: 23B0      MEMHGH *      $0005  MEMORY HIGH
0190: 23B0      VIA   *      $0006  VIA 6522 POINTERS
0200: 23B0      TMPLOW *      $0008  TEMPORARY LOW
0210: 23B0      TMPHGH *      $0009  TEMPORARY HIGH
0220:
0230:          VIA REGISTERS
0240:
0250: 23B0      ORB   *      $0000  OUTPUT REGISTER B
0260: 23B0      ORA   *      $0001  OUTPUT REGISTER A
0270: 23B0      DDRB *      $0002  DATA DIRECTION REGISTER B
0280: 23B0      DDRA *      $0003  DATA DIRECTION REGISTER A
0290: 23B0      ACR  *      $000B  AUXILIARY CONTROL REGISTER
0300: 23B0      PCR  *      $000C  PERIPHERAL CONTROL REGISTER
0310: 23B0      IFR  *      $000D  INTERRUPT FLAG REGISTER
0320: 23B0      IER  *      $000E  INTERRUPT ENABLE REGISTER
0330:
0340:          ASCII CHARACTERS
0350:
0360: 23B0      BS   *      $0008  BS = BACKSPACE
0370: 23B0      HTAB *      $0009  TAB = END OF SCREEN
0380: 23B0      LF   *      $000A  LF = DOWN LINE
0390: 23B0      VTAB *      $000B  VERTICAL TAB = UPLINE
0400: 23B0      FF   *      $000C  FORMFEED = HOME
0410: 23B0      CR   *      $000D  CR = CARRIAGE RETURN
0420: 23B0      SCRLUP *      $000E  SO = SCROLL UP CONTROL N
0430: 23B0      SCRLDN *      $000F  SI = SCROLL DOWN CONTROL O
0440: 23B0      EOL  *      $0011  DC1 = CLEAR TO END OF LINE
0450: 23B0      EOF  *      $0012  DC2 = CLEAR TO END OF FILE
0460: 23B0      FEOL *      $0013  DC3 = FILL TO END OF LINE
0470: 23B0      FEOF *      $0014  DC4 = FILL TO END OF FILE
0480: 23B0      ESC  *      $001B  ESCAPE = PCG MODE
0490: 23B0      FS   *      $001C  FS = FORWARD SPACE
0500: 23B0      SPACE *      $0020  SPACE CHARACTER
0510: 23B0      PERIOD *      $002E  PERIOD CHARACTER
0520: 23B0      DELETE *      $007F  DELETE CODE
0530:
0540:          RAM MEMORY LOCATIONS
0550:
0560: 2000          ORG   $2000

```

```

0570:
0580: 2000 00      COLUMN =      $00
0590: 2001 00      MINCOL =     $00
0600: 2002 00      MAXCOL =     $00
0610: 2003 00      ABSCOL =     $00
0620: 2004 00      ROW         =     $00
0630: 2005 00      MINROW =     $00
0640: 2006 00      MAXROW =     $00
0650: 2007 00      ABSROW =     $00
0660: 2008 00      LPFIX =      $00
0670: 2009 00      CHAR        =     $00
0680: 200A 00      MEMSTR =     $00
0690: 200B 00      MEMEND =     $00
0700: 200C 00      PCGSTR =     $00
0710: 200D 00      PCGEND =     $00
0720: 200E 00      OFFLOW =     $00
0730: 200F 00      OFFHGH =     $00
0740:
0750:      INITIALIZATION TABLE
0760:      THIS MAY BE LOCATED ANYWHERE IN MEMORY
0770:      THE USER MUST SET UP A POINTER TO THE START
0780:      OF THE TABLE - IF MOVED ELSEWHERE - AND ENTER
0790:      THE INIT ROUTINE AT LOCATION USRINT
0800:
0810:      THE VALUES WILL BE CHANGED TO SUIT YOUR OWN
0820:      SYSTEM, MONITOR, DISPLAY REQUIREMENTS, ....
0830:      SEE THE 6845 MANUAL FOR DETAILS
0840:
0850:      CRT CONTROLLER VALUES
0860:
0870: 2010 7A      HTOTAL =     $7A
0880: 2011 50      HDISP  =     $50      80 DISPLAYABLE CHARACTERS
0890: 2012 60      HSYNC  =     $60
0900: 2013 0A      HWIDTH =     $0A
0910: 2014 13      VTOTAL =     $13      20 DISPLAYABLE LINES
0920: 2015 1E      VSLA   =     $1E      VERT. SCAN LINE ADJUST
0930: 2016 14      VDISP  =     $14
0940: 2017 14      VSYNC  =     $14
0950: 2018 00      IMODE  =     $00      NON-INTERLACE
0960: 2019 0C      MSLA   =     $0C      13 SCAN LINES PER CHAR
0970: 201A 4C      CURSTR =     $4C      BLINKING CURSOR AT BOTTOM LINE
0980: 201B 0C      CUREND =     $0C
0990: 201C 00      DSAH   =     $00      DISPLAY START ADDRESS
1000: 201D 00      DSAL   =     $00      DISPLAY START ADDRESS LOW
1010: 201E 00      CURHI  =     $00      CURSOR HIGH
1020: 201F 00      CURLO  =     $00      CURSOR LOW
1030:
1040:      SYSTEM ADDRESSES
1050:
1060: 2020 4F      MEMORY =     $4F      HIGHEST DISPLAY MEMORY PAGE
1070: 2021 05      LPXX   =     $05      FUDGE FACTOR FOR LIGHT PEN
1080:
ID=11

0010:      VIDEO PLUS
0020:
0030:      INIT WILL INITIALIZE THE VIDEO PLUS.
0040:

```

0000
00

```

0050:          IT FIRST MOVES A COUPLE OF PARAMETERS TO THEIR
0060:          WORKING LOCATIONS.  IT THEN INITIS THE CRT CONTROLLER.
0070:          IT INITIS THE KEYBOARD AND LIGHT PEN I/O.  IT
0080:          FINISHES BY HOMING THE CURSOR.
0090:
0100: 2022 A9 00      INIT   LDAIM $00      CLEAR ALL STATUS FLAGS
0110: 2024 48          PHA
0120: 2025 28          PLP
0130: 2026 A9 10      LDAIM HTOTAL GET INIT TABLE
0140: 2028 85 08      STA   TMPLOW SETUP INDIRECT POINTER
0150: 202A A9 20      LDAIM HTOTAL /
0160: 202C 85 09      STA   TMPHGH
0170:
0180:          USER INIT ENTRY POINT
0190:          USER MAY ENTER HERE WITH POINTERS TO A
0200:          TABLE OF INITIALIZATION VALUES DIFFERENT
0210:          FROM THOSE STORED ABOVE.
0220:          TMPLOW = LOW ADDRESS OF USER TABLE
0230:          TMPHGH = HIGH ADDRESS OF USER TABLE
0240:
0250: 202E A0 11      USRINT LDYIM $11      INIT COUNTER/POINTER
0260: 2030 B1 08          LDAIY TMPLOW GET FUDGE FACTOR FOR LIGHT PEN
0270: 2032 8D 08 20      STA   LPFIX
0280: 2035 88          DEY
0290: 2036 B1 08          LDAIY TMPLOW GET DISPLAY MEMORY HIGH PAGE NUMBER
0300: 2038 8D 0B 20      STA   MEMEND SAVE IT
0310: 203B 29 F0          ANDIM $F0      MASK TO DISPLAY MEMORY LOW PAGE NUMBER
0320: 203D 8D 0A 20      STA   MEMSTR
0330: 2040 18          CLC
0340: 2041 69 10          ADCIM $10      CALCULATE PROG. CHAR. GEN.
0350: 2043 8D 0C 20      STA   PCGSTR ADDRESS
0360: 2046 69 07          ADCIM $07      CALCULATE PCG HIGH PAGE
0370: 2048 8D 0D 20      STA   PCGEND
0380: 204B 69 01          ADCIM $01      CALCULATE CRT CONTROLLER ADDRESSES
0390: 204D 85 01          STA   CRTCON +01 SAVE PAGE ADDRESS
0400: 204F 85 03          STA   CRTREG +01
0410: 2051 85 07          STA   VIA      +01 GOOD FOR VIA TOO
0420: 2053 A9 00          LDAIM $00      SET LOW CONTROLLER ADDRESSES
0430: 2055 85 00          STA   CRTCON
0440: 2057 A9 01          LDAIM $01
0450: 2059 85 02          STA   CRTREG
0460: 205B A9 10          LDAIM $10      AND VIA LOW ADDRESS
0470: 205D 85 06          STA   VIA
0480: 205F A0 0F          LDYIM $0F      SET POINTER FOR CRT TABLE
0490: 2061 A2 00          LDXIM $00      FIX INDEX FOR ILOOP
0500:
0510: 2063 98          ILOOP TYA          Y VALUE = REGISTER VALUE
0520: 2064 81 00          STAIX CRTCON SET CONTROL REGISTER
0530: 2066 B1 08          LDAIY TMPLOW GET REGISTER CONTENTS
0540: 2068 81 02          STAIX CRTREG PUT INTO CRT REGISTER
0550: 206A 88          DEY
0560: 206B 10 F6          BPL   ILOOP CONTINUE FOR ALL VALUES
0570:
0580:          SET UP CRT SCREEN LIMITS BASED ON CRT VALUES
0590:
0600: 206D A0 01          LDYIM $01      GET WIDTH FROM HORZ. DISP.

```



```

0610: 206F B1 08          LDAIY TMPLOW
0620: 2071 38             SEC          CONVERT NO. CHAR TO
0630: 2072 E9 01          SBCIM $01    LAST CHAR NUMBER
0640: 2074 8D 03 20      STA  ABSCOL
0650: 2077 8D 02 20      STA  MAXCOL
0660: 207A A0 06          LDYIM $06    GET LINES FROM VERT. DISP.
0670: 207C B1 08          LDAIY TMPLOW
0680: 207E 38             SEC
0690: 207F E9 01          SBCIM $01
0700: 2081 8D 07 20      STA  ABSROW
0710: 2084 8D 06 20      STA  MAXROW
0720: 2087 A9 00          LDAIM $00    SET MINIMUM ROW/COLUMN
0730: 2089 8D 01 20      STA  MINCOL
0740: 208C 8D 05 20      STA  MINROW
0750: 208F 20 1E 21      JSR  HOME    SET CURSOR AND MEMORY TO START
0760:
0770:                   INITIALIZE THE KEYBOARD AND LIGHT PEN
0780:
0790: 2092 A0 0E          LDYIM IER    DISABLE ALL INTERRUPTS
0800: 2094 A9 7F          LDAIM $7F
0810: 2096 91 06          STAIY VIA
0820: 2098 A0 0D          LDYIM IFR    CLEAR ANY INTERRUPTS PENDING
0830: 209A A9 FF          LDAIM $FF
0840: 209C 91 06          STAIY VIA
0850: 209E A0 03          LDYIM DDRA   SET DATA DIRECTION FOR INPUT
0860: 20A0 A9 00          LDAIM $00
0870: 20A2 91 06          STAIY VIA
0880: 20A4 A0 0B          LDYIM ACR    SET TO LATCH KEYBOARD DATA
0890: 20A6 A9 01          LDAIM $01
0900: 20A8 91 06          STAIY VIA
0910:
0920:                   KEYBOARD CA1 SET ON POSITIVE TRANSITION (01)
0930:                   LIGHT PEN CA2 SET ON POSITIVE TRANSITION
0940:                   WITH NO PORT A INVOLVEMENT (06)
0950:                   CB2 MAY BE USED TO DISPLAY OR BLANK
0960:                   CONTROL CODES HEX 00-1F AND 80 TO 9F
0970:                   SEE APPLICATION NOTE 1 FOR DETAILS
0980:
0990: 20AA A0 0C          LDYIM PCR    SET TRANSITION DIRECTION
1000: 20AC A9 06          LDAIM $06    KBRD POS CA1, LP POS CA2
1010: 20AE 91 06          STAIY VIA
1020: 20B0 60             RTS          INITIALIZATION COMPLETE
1030:
ID=12

0010:
0020:                   CONTROL SUBROUTINE
0030:                   THIS HANDLES ALL NORMAL CONTROL CODES
0040:
0050:                   JSR CONTRL
0060:                   C = 1, NORMAL CONTROL CODE
0070:                   C = 0, UNUSUAL CONTROL CHARACTER IN A
0080:                   A, Y, X USED
0090:
0100: 20B1 C9 0D          CONTRL CMPIM CR    CARRIAGE RETURN ?
0110: 20B3 D0 1D          BNE  LFEED NO
0120: 20B5 A9 00          CRET  LDAIM $00    RESET COLUMN
0130: 20B7 8D 00 20      STA  COLUMN

```

```

0140: 20BA AD 04 20 DOWN LDA ROW TEST LAST ROW
0150: 20BD CD 06 20 CMP MAXROW
0160: 20C0 D0 05 BNE DOWNOK NOT LAST ROW
0170: 20C2 20 ED 21 JSR DNSCR LAST ROW, SO AUTOMATIC SCROLL
0180: 20C5 B0 03 BCS SET IF SET, THEN AT END OF MEMORY
0190: 20C7 EE 04 20 DOWNOK INC ROW ELSE, BUMP ROW POINTER
0200: 20CA 20 74 22 SET JSR RCMEM CONVERT ROW, COLUMN TO MEMORY
0210: 20CD 20 5B 22 JSR SETCUR THEN SET CURSOR
0220: 20D0 38 SEC
0230: 20D1 60 RTS
0240:
0250: 20D2 C9 0A LFEED CMPIM LF TEST LINE FEED
0260: 20D4 F0 E4 BEQ DOWN YES.
0270:
0280: 20D6 C9 0B UPLINE CMPIM VTAB VERTICAL TAB ?
0290: 20D8 D0 12 BNE RIGHTX NOT LINE FEED
0300: 20DA AD 04 20 UP LDA ROW TEST TOP OF SCREEN
0310: 20DD CD 05 20 CMP MINROW
0320: 20E0 D0 05 BNE UPOK
0330: 20E2 20 9B 21 JSR UPSCR ALREADY AT TOP OF SCREEN
0340: 20E5 B0 E3 BCS SET IF SET, NO SCROLL
0350: 20E7 CE 04 20 UPOK DEC ROW SET NEW ROW
0360: 20EA 10 DE BPL SET ALWAYS BRANCH
0370:
0380: 20EC C9 1C RIGHTX CMPIM FS FORWARD SPACE ?
0390: 20EE D0 13 BNE LEFTX NO.
0400: 20F0 EE 00 20 RIGHT INC COLUMN
0410: 20F3 AD 00 20 LDA COLUMN TEST END OF LINE
0420: 20F6 CD 02 20 CMP MAXCOL
0430: 20F9 D0 CF BNE SET OKAY IF NOT EQUAL
0440: 20FB AD 01 20 LDA MINCOL ELSE, RESET TO LEFT
0450: 20FE 8D 00 20 STA COLUMN SIDE OF SCREEN
0460: 2101 10 C7 BPL SET ALWAYS BRANCH
0470:
0480: 2103 C9 08 LEFTX CMPIM BS BACK SPACE ?
0490: 2105 D0 13 BNE HOMEX NO
0500: 2107 CE 00 20 LEFT DEC COLUMN DECREMENT TO MOVE LEFT
0510: 210A AD 00 20 LDA COLUMN TEST FOR OUT OF RANGE
0520: 210D CD 01 20 CMP MINCOL
0530: 2110 10 B8 BPL SET IF POSITIVE, THEN OKAY
0540: 2112 AD 02 20 LDA MAXCOL ELSE, RESET TO RIGHT
0550: 2115 8D 00 20 STA COLUMN SIDE OF SCREEN
0560: 2118 10 B0 BPL SET ALWAYS BRANCH
0570:
0580: 211A C9 0C HOMEX CMPIM FF HOME COMMAND ?
0590: 211C D0 0E BNE ENDFLD NO
0600: 211E AD 01 20 HOME LDA MINCOL RESET TO UPPER
0610: 2121 8D 00 20 STA COLUMN LEFT CORNER
0620: 2124 AD 05 20 LDA MINROW
0630: 2127 8D 04 20 STA ROW
0640: 212A 10 9E BPL SET ALWAYS BRANCH
0650:
0660: 212C C9 09 ENDFLD CMPIM HTAB SET TO END OF FIELD
0670: 212E D0 0E BNE CEOLX NO
0680: 2130 AD 02 20 SETEND LDA MAXCOL
0690: 2133 8D 00 20 STA COLUMN

```



```

0700: 2136 AD 06 20      LDA  MAXROW
0710: 2139 8D 04 20      STA  ROW
0720: 213C 10 8C        BPL  SET    ALWAYS BRANCH
0730:
ID=13

0010: 213E C9 11      CEOLX  CMPIM  EOL    CLEAR TO END OF LINE?
0020: 2140 D0 24        BNE  CEOFX  NO
0030: 2142 20 4A 21    CLREOL JSR    CLEAR
0040: 2145 20 3E 22    FIXCUR JSR    GETCUR  SHOW CURSOR AND RETURN
0050: 2148 38          SEC          C = 1 = NORMAL RETURN
0060: 2149 60          RTS
0070:
0080: 214A A9 20      CLEAR  LDAIM  SPACE  FILL WITH SPACES
0090: 214C 85 08      FILLIT STA  TMPLOW  SAVE FILL CHARACTER
0100: 214E 20 74 22    FWRITE JSR    RCMEM  SET MEMORY POINTERS
0110: 2151 A0 00        LDYIM $00    TO CURRENT ROW AND COLUMN
0120: 2153 A5 08      LDA  TMPLOW  RESTORE FILL CHARACTER
0130: 2155 91 04      STAIY MEMLOW AND WRITE IT
0140: 2157 20 91 22    FBUMP  JSR    INCR   BUMP POINTER
0150: 215A B0 09      BCS  FRTN   C = 1 = END OF LINE
0160: 215C AD 01 20    LDA  MINCOL TEST FIRST COLUMN FOR
0170: 215F CD 00 20    CMP  COLUMN START OF NEXT LINE
0180: 2162 D0 EA      BNE  FWRITE OLD LINE, SO CONTINUE
0190: 2164 18        CLC          NEW LINE, SO DONE
0200: 2165 60      FRTN  RTS
0210:
0220: 2166 C9 12      CEOFX  CMPIM  EOF    CLEAR TO END OF FILE
0230: 2168 D0 07        BNE  FEOLX  NO
0240: 216A 20 4A 21    CLREOF JSR    CLEAR  CLEAR TO END OF LINE
0250: 216D 90 FB      BCC  CLREOF NOT LAST ROW
0260: 216F B0 D4      BCS  FIXCUR DONE
0270:
0280: 2171 C9 13      FEOLX  CMPIM  FEOL   FILL TO END OF LINE
0290: 2173 D0 09        BNE  FEOFX
0300:
0310: 2175 A0 00      FILEOL LDYIM $00    GET CURRENT CHARACTER
0320: 2177 B1 04        LDAIY MEMLOW
0330: 2179 20 4C 21    JSR  FILLIT FILL TO END OF LINE
0340: 217C 50 C7      BVC  FIXCUR ALWAYS BRANCH
0350:
0360: 217E C9 14      FEOFX  CMPIM  FEOF   FILL TO END OF FILE?
0370: 2180 D0 10        BNE  SCRUP  NO
0380: 2182 A0 00      FILEOF LDYIM $00
0390: 2184 B1 04        LDAIY MEMLOW GET CURRENT CHARACTER
0400: 2186 20 4C 21    FILL  JSR  FILLIT
0410: 2189 B0 05      BCS  FEND   ONE LINE FILLED
0420: 218B 20 4E 21    FMORE JSR  FWRITE DO ANOTHER LINE
0430: 218E 90 FB      BCC  FMORE  NOT DONE YET
0440: 2190 50 B3      FEND  BVC  FIXCUR ALWAYS BRANCH
0450:
0460: 2192 C9 0E      SCRUP  CMPIM  SCRLUP  SCROLL UP ?
0470: 2194 D0 4E        BNE  SCRDN  NO
0480: 2196 20 9B 21    JSR  UPSCRL SCROLL UP
0490: 2199 50 47      BVC  NOSCRL
0500:
0510: 219B A0 00      UPSCRL LDYIM $00    CALC END OF SCREEN
0520: 219D AE 07 20    LDX  ABSROW

```

```

0530: 21A0 E8          INX
0540: 21A1 AD 0E 20    LDA  OFFLOW PLUS CURRENT OFFSET
0550: 21A4 38          UPSCA SEC          PLUS 1
0560: 21A5 6D 03 20    ADC  ABSCOL
0570: 21A8 90 01          BCC  UPSCB
0580: 21AA C8          INY          OVERFLOW
0590: 21AB 18          UPSCB CLC
0600: 21AC CA          DEX
0610: 21AD 10 F5          BPL  UPSCA
0620: 21AF 98          TYA          GET OVERFLOW
0630: 21B0 18          CLC
0640: 21B1 6D 0F 20    ADC  OFFHGH ADD TO HIGH OFFSET
0650: 21B4 6D 0A 20    ADC  MEMSTR
0660: 21B7 CD 0B 20    CMP  MEMEND TEST END OF MEMORY
0670: 21BA F0 02          BEQ  SCRLOK END NOT EXCEEDED
0680: 21BC 10 24          BPL  NOSCRL NO SCROLL
0690: 21BE AD 0E 20    SCRLOK LDA  OFFLOW CALC. NEW OFFSETS
0700: 21C1 38          SEC
0710: 21C2 6D 03 20    ADC  ABSCOL
0720: 21C5 8D 0E 20    STA  OFFLOW
0730: 21C8 AD 0F 20    LDA  OFFHGH
0740: 21CB 69 00          ADCIM $00    BUMP IF PRIOR CARRY
0750: 21CD 8D 0F 20    STA  OFFHGH NEW HIGH
0760: 21D0 EE 06 20    INC  MAXROW
0770: 21D3 EE 05 20    INC  MINROW
0780:
0790: 21D6 20 74 22    SCRF IN JSR  RCMEM CONVERT
0800: 21D9 20 5B 22          JSR  SETCUR
0810: 21DC 20 29 22          JSR  SETCD
0820: 21DF B8          CLV
0830: 21E0 18          CLC
0840: 21E1 60          RTS
0850:
0860: 21E2 38          NOSCRL SEC          SET CARRY
0870: 21E3 60          RTS          AND RETURN
0880:
0890: 21E4 C9 0F          SCRDN CMPIM SCRLDN SCROLL DOWN?
0900: 21E6 D0 2A          BNE  TOGGLE
0910: 21E8 20 ED 21          JSR  DNSCRL DOWN SCROLL
0920: 21EB 50 F5          BVC  NOSCRL
0930:
0940: 21ED AD 0F 20    DNSCRL LDA  OFFHGH TEST ALREADY AT TOP
0950: 21F0 D0 05          BNE  DNOK
0960: 21F2 AD 0E 20    LDA  OFFLOW NEAR TOP
0970: 21F5 F0 EB          BEQ  NOSCRL AT TOP
0980: 21F7 18          DNOK CLC
0990: 21F8 AD 0E 20    LDA  OFFLOW
1000: 21FB ED 03 20    SBC  ABSCOL
1010: 21FE 8D 0E 20    STA  OFFLOW
1020: 2201 AD 0F 20    LDA  OFFHGH
1030: 2204 E9 00          SBCIM $00
1040: 2206 8D 0F 20    STA  OFFHGH
1050: 2209 CE 06 20    DEC  MAXROW
1060: 220C CE 05 20    DEC  MINROW
1070: 220F 4C D6 21    JMP  SCRF IN
1080:

```

```

1090: 2212 C9 18      TOGGLE CMPIM $18      CONTROL X ?
1100: 2214 D0 0A      BNE      OTHER      NO
1110: 2216 A0 0C      LDYIM PCR      YES. TOGGLE CB2 TO CONTROL
1120: 2218 B1 06      LDAIY VIA      DISPLAY/NON-DISPLAY OF
1130: 221A 49 20      EDRIM $20      CODES 00 - 1F AND 80 - 9F
1140: 221C 91 06      STAIY VIA      SEE APPLICATION NOTE 1
1150: 221E 50 C2      BVC      NOSCRL     NORMAL RETURN
1160:
1170: 2220 18          OTHER CLC          UNUSUAL CONTROL CODE
1180: 2221 B8          CLV
1190: 2222 60          RTS
1200:
1210:                SETOFF SUBROUTINE - SETS HARDWARE OFFSET
1220:                FOR SCROLLING
1230:
1240:                A = LOW OFFSET ADDRESS
1250:                X = HIGH OFFSET ADDRESS
1260:
1270:                JSR SETOFF
1280:                A,Y USED X NOT MODIFIED
1290:
1300: 2223 8D 0E 20     SETOFF STA      OFFLOW
1310: 2226 8E 0F 20     STX      OFFHGH
1320: 2229 A2 00         SETCD LDXIM $00     MOVE OFFSET TO CRTC
1330: 222B A9 0C         LDAIM $0C          REGISTERS C,D
1340: 222D 81 00         STAIX CRTCON
1350: 222F AD 0F 20     LDA      OFFHGH
1360: 2232 81 02         STAIX CRTREG     FIX NEW HIGH
1370: 2234 A9 0D         LDAIM $0D
1380: 2236 81 00         STAIX CRTCON
1390: 2238 AD 0E 20     LDA      OFFLOW
1400: 223B 81 02         STAIX CRTREG
1410: 223D 60          RTS
1420:
1430:                GET CURSOR POSITION FROM CRT CHIP
1440:
1450:                JSR GETCUR
1460:                CURSOR VALUES MOVED FROM CRT CHIP TO MEMORY POINTERS
1470:                JSR GETCUR
1480:                A,X USED
1490:
1500: 223E A9 0E         GETCUR LDAIM $0E     HIGH CURSOR REGISTER
1510: 2240 A2 00         LDXIM $00
1520: 2242 81 00         STAIX CRTCON     SETUP REGISTER
1530: 2244 A1 02         LDAIX CRTREG     READ
1540: 2246 18          CLC
1550: 2247 6D 0A 20     ADC      MEMSTR     ADD TO MEMORY
1560: 224A 85 05         STA      MEMHGH
1570: 224C A9 0F         LDAIM $0F         CURSOR LOW REGISTER
1580: 224E 81 00         STAIX CRTCON
1590: 2250 A1 02         LDAIX CRTREG
1600: 2252 85 04         STA      MEMLOW
1610: 2254 B8          CLV
1620: 2255 18          CLC
1630: 2256 60          RTS
1640:
ID=14

```



```

0010:          VIDEO SUBROUTINES
0020:
0030:          SETMEM SUBROUTINE
0040:
0050:          SET MEMORY ADDRESS FROM A AND Y
0060:          THEN SET CURSOR TO MEMORY ADDRESS
0070:
0080:          JSR SETMEM
0090:          A = MEMORY LOW ADDRESS
0100:          Y = MEMORY HIGH ADDRESS
0110:          A, Y, X USED, C = 0, V = 0
0120:
0130: 2257 85 04   SETMEM STA  MEMLOW
0140: 2259 84 05           STY  MEMHGH
0150:
0160:          SETCUR SUBROUTINE
0170:
0180:          SET CRT CURSOR TO MEMORY ADDRESS
0190:
0200:          JSR SETCUR
0210:          A, X USED
0220:
0230: 225B A9 0E   SETCUR LDAIM $0E   CURSOR HIGH
0240: 225D A2 00           LDXIM $00
0250: 225F 81 00           STAIX CRTCON
0260: 2261 A5 05           LDA  MEMHGH CURRENT MEMORY HIGH
0270: 2263 38           SEC          SET BORROW
0280: 2264 ED 0A 20   SBC  MEMSTR DISPLAY RAM MEMORY START PAGE
0290: 2267 81 02           STAIX CRTREG WRITE TO CRT CONTROLLER
0300: 2269 A9 0F           LDAIM $0F   CURSOR LOW
0310: 226B 81 00           STAIX CRTCON
0320: 226D A5 04           LDA  MEMLOW CURRENT MEMORY LOW
0330: 226F 81 02           STAIX CRTREG
0340: 2271 18           DONE  CLC          NORMAL RETURN
0350: 2272 B8           CLV
0360: 2273 60           RTS
0370:
0380:          ROW/COLUMN TO MEMORY ADDRESS CONVERSION
0390:
0400:          JSR RCMEM
0410:          A, X USED
0420:
0430: 2274 AD 0A 20   RCMEM  LDA  MEMSTR INIT START OF MEMORY PAGE
0440: 2277 85 05           STA  MEMHGH
0450: 2279 AD 00 20   LDA  COLUMN CURRENT COLUMN
0460: 227C AE 04 20   LDX  ROW    CURRENT ROW
0470: 227F F0 0B           BEQ  RCMEMC SKIP IF ROW ZERO
0480:
0490: 2281 38           RCMEMA SEC          ADD ABSOLUTE COLUMN MAX + 1
0500: 2282 6D 03 20   ADC  ABSCOL
0510: 2285 90 02           BCC  RCMEMB
0520: 2287 E6 05           INC  MEMHGH BUMP MEMHGH ON CARRY
0530: 2289 CA           RCMEMB DEX          DECR ROW COUNTER
0540: 228A D0 F5           BNE  RCMEMA CONTINUE UNTIL DONE
0550:
0560: 228C 85 04   RCMEMC STA  MEMLOW SAVE LOW POINTER

```

```

0570: 228E 18          CLC          NORMAL RETURN
0580: 228F B8          CLV
0590: 2290 60          RTS
0600:
0610:          INCREMENT AND DECREMENT POINTERS
0620:
0630:          JSR INCR
0640:          A USED
0650:          C = 0 NOT END OF FIELD
0660:          C = 1 END OF FIELD
0670:
0680: 2291 EE 00 20     INCR   INC   COLUMN BUMP COLUMN POINTER
0690: 2294 AD 02 20          LDA   MAXCOL TEST OVERFLOW
0700: 2297 CD 00 20          CMP   COLUMN
0710: 229A 10 1B          BPL   IDRTN  STILL IN FIELD
0720: 229C AD 04 20          LDA   ROW   TEST LAST ROW
0730: 229F CD 06 20          CMP   MAXROW
0740: 22A2 D0 0A          BNE   INCRA
0750: 22A4 CE 00 20          DEC   COLUMN RESET COLUMN
0760: 22A7 10 03          BPL   INCRY  ALWAYS
0770: 22A9 EE 00 20     INCRX  INC   COLUMN
0780: 22AC 38          INCRY  SEC          SET CARRY FOR END OF FIELD
0790: 22AD 60          RTS          AND RETURN
0800:
0810: 22AE AD 01 20     INCRA  LDA   MINCOL RESET TO MINIMUM COLUMN
0820: 22B1 8D 00 20          STA   COLUMN
0830: 22B4 EE 04 20          INC   ROW   BUMP ROW
0840:
0850: 22B7 18          IDRTN  CLC          CLEAR CARRY FOR NORMAL RETURN
0860: 22B8 60          RTS
0870:
0880:          JSR DECR  DECREMENT POINTERS
0890:          A USED
0900:          C = 0 NOT END OF FIELD
0910:          C = 1 END OF FIELD
0920:
0930: 22B9 CE 00 20     DECR   DEC   COLUMN DECR. COLUMN
0940: 22BC AD 00 20          LDA   COLUMN TEST UNDERFLOW
0950: 22BF CD 01 20          CMP   MINCOL AGAINST MINIMUM COLUMN
0960: 22C2 10 F3          BPL   IDRTN  STILL IN FIELD
0970: 22C4 AD 04 20          LDA   ROW   TEST TOP ROW
0980: 22C7 CD 05 20          CMP   MINROW SINCE THERE IS NO WRAP-AROUND
0990: 22CA F0 DD          BEQ   INCRX  RESET TO FIRST POSITION
1000: 22CC AD 02 20          LDA   MAXCOL ELSE, SET MAXIMUM COLUMN
1010: 22CF 8D 00 20          STA   COLUMN
1020: 22D2 CE 04 20          DEC   ROW   AND DECR. ROW
1030: 22D5 50 E0          BVC   IDRTN  AND MAKE NORMAL RETURN
1040:
1050:          CRTSET SUBROUTINE
1060:
1070:          SET A VALUE INTO THE CRT CONTROLLER
1080:
1090:          A = VALUE
1100:          X = REGISTER
1110:          JSR CRTSET
1120:          A,X UNCHANGED

```

```

1130:
1140: 22D7 A0 00      CRTSET LDYIM $00
1150: 22D9 48          PHA          SAVE VALUE
1160: 22DA 8A          TXA
1170: 22DB 91 00      STAIY CRTCON
1180: 22DD 68          PLA          GET VALUE
1190: 22DE 91 02      STAIY CRTREG
1200: 22E0 60          RTS
1210:
1220:                CURSET SUBROUTINE
1230:                SET CURSOR DIRECTLY
1240:
1250:                A = COLUMN
1260:                X = ROW
1270:                IF EITHER VALUE IS = FF, THEN LEAVE THAT ALONE
1280:
1290:                JSR CURSET
1300:                A,X USED
1310:
1320: 22E1 C9 FF      CURSET CMPIM $FF      TEST DUMMY COLUMN VALUE IN A
1330: 22E3 F0 03      BEQ      SETROW
1340: 22E5 8D 00 20   STA      COLUMN NEW COLUMN
1350: 22E8 8A          SETROW TXA
1360: 22E9 C9 FF      CMPIM $FF      TEST DUMMY ROW IN X
1370: 22EB F0 03      BEQ      SETDUN
1380: 22ED 8D 04 20   STA      ROW      SET NEW ROW
1390: 22F0 20 74 22   SETDUN JSR      RCMEM CONVERT
1400: 22F3 4C 5B 22   JMP      SETCUR AND SET CURSOR
1410:
1420:                READ CHARACTER FROM CURRENT CURSOR POSITION
1430:                READ AND INCREMENT OR JUST READ
1440:
1450:                JSR RINC  READ AND INCREMENT
1460:                A = CHARACTER
1470:                C = 0 = NOT END OF FIELD
1480:                C = 1 = END OF FIELD
1490:
1500:                JSR READ  READ WITH NO INCREMENT
1510:
1520: 22F6 20 02 23   RINC   JSR      READ
1530: 22F9 48          PHA          SAVE A WHICH HAS CHARACTER
1540: 22FA 20 91 22   JSR      INCR
1550: 22FD 20 CA 20   JSR      SET      RCMEM AND SETCUR SUBROUTINES
1560: 2300 68          PLA          RESTORE CHARACTER IN A
1570: 2301 60          RTS
1580:
1590: 2302 20 74 22   READ   JSR      RCMEM
1600: 2305 A0 00      LDYIM $00
1610: 2307 B1 04      LDAIY MEMLOW
1620: 2309 60          RTS
1630:                WRITE CHARACTER TO CURRENT CURSOR POSITION
1640:                WRITE WITH INCREMENT OR JUST WRITE
1650:
1660:                JSR WINC  WRITE AND INCREMENT
1670:                C = 0 = NOT END OF FIELD
1680:                C = 1 = END OF FIELD

```



```

1690:
1700:          JSR WRITE
1710:          A = CHARACTER
1720:          A,Y,X USED
1730:
1740: 230A 20 16 23 WINC  JSR  WRITE
1750: 230D 20 91 22      JSR  INCR
1760: 2310 08          PHP          SAVE INCR STATUS FOR TESTING
1770: 2311 20 CA 20    JSR  SET          RCMEM AND SETCUR SUBROUTINES
1780: 2314 28          PLP          RESTORE INCR STATUS FOR TESTING
1790: 2315 60          RTS
1800:
1810: 2316 48          WRITE PHA          SAVE CHARACTER
1820: 2317 A0 00      LDYIM $00
1830: 2319 68          PLA          RESTORE CHARACTER
1840: 231A 91 04      STAIY MEMLOW
1850: 231C 60          RTS
1860:
ID=15

0010:          KEYBOARD SERVICE
0020:
0030:          KBTEST SUBROUTINE
0040:          TEST KEYBOARD FOR DATA PRESENT
0050:
0060:          JSR KBTEST
0070:          C = 0, NO DATA
0080:          C = 1, KEYBOARD HAS DATA
0090:          A,Y CLOBBERED
0100:
0110: 231D 18          KBTEST CLC
0120: 231E A0 0D      LDYIM IFR          READ FROM INTERRUPT REGISTER
0130: 2320 B1 06      LDAIY VIA          INDIRECT TO MAKE LIFE EASY
0140: 2322 29 02      ANDIM $02          MASK TO CA1 FLAG
0150: 2324 F0 01      BEQ  NODATA        IF ZERO, THEN NO DATA
0160: 2326 38          SEC          SET CARRY TO INDICATE DATA
0170: 2327 60          NODATA RTS          RETURN
0180:
0190:          KBREAD SUBROUTINE
0200:          TEST KEYBOARD AND READ DATA IF PRESENT
0210:          IF DATA NOT PRESENT, RETURN ANYWAY
0220:
0230:          JSR KBREAD
0240:          C = 0, NO DATA
0250:          C = 1, CHARACTER IN A
0260:          A,Y USED
0270:
0280: 2328 20 1D 23  KBREAD JSR  KBTEST TEST DATA PRESENT
0290: 232B 90 FA      BCC  NODATA
0300: 232D A0 01      KBDATA LDYIM ORA    READ DATA
0310: 232F B1 06      LDAIY VIA          FROM VIA PORT A
0320: 2331 60          RTS          AND RETURN
0330:
0340:          KBWAIT SUBROUTINE
0350:          TEST KEYBOARD AND WAIT FOR DATA.
0360:          DO NOT RETURN UNTIL DATA IS PRESENT
0370:
0380:          JSR KBWAIT

```

```

0390:          C = 1, CHARACTER IN A
0400:          A,Y USED
0410:
0420: 2332 20 1D 23 KBWAIT JSR   KBTEST DATA PRESENT?
0430: 2335 90 FB          BCC   KBWAIT IF NOT, TRY AGAIN
0440: 2337 B0 F4          BCS   KBDATA ELSE, READ DATA
0450:
0460:          GETCHR SUBROUTINE
0470:          WAIT FOR CHARACTER FROM KEYBOARD AND CONVERT TO
0480:          PCG CHARACTER IF REQUIRED
0490:
0500:          JSR GETCHR
0510:          C = 1, CHARACTER IN A
0520:          A,Y USED
0530:
0540: 2339 20 32 23 GETCHR JSR   KBWAIT WAIT FOR KEYBOARD DATA
0550: 233C C9 1B          CMPIM ESC   TEST ESCAPE CHARACTER
0560: 233E D0 05          BNE   NORMAL IF NOT, THEN NORMAL RETURN
0570: 2340 20 32 23          JSR   KBWAIT ELSE, WAIT FOR NEXT CHARACTER
0580: 2343 09 80          ORAIM $80   TURN ON BIT 80 FOR PCG
0590: 2345 60          NORMAL RTS   AND RETURN
0600:
0610:          INCHAR SUBROUTINE
0620:          GET CHARACTER WITH PCG CONVERSION, THEN
0630:          TEST FOR CONTROL CODES AND HANDLE THEM
0640:
0650:          JSR INCHAR
0660:          C = 1, NON CONTROL CHARACTER IN A
0670:          C = 0, UNUSUAL CONTROL CHARACTER IN A
0680:          A,Y,X USED
0690:
0700: 2346 20 39 23 INCHAR JSR   GETCHR GET AND CONVERT CHARACTER
0710: 2349 C9 20          CMPIM SPACE TEST FOR CONTROL CODES
0720: 234B 10 F8          BPL   NORMAL SPACE OR GREATER, MAKE NORMAL RETURN
0730: 234D 20 B1 20          JSR   CONTRL TEST CONTROL CODES
0740: 2350 B0 F4          BCS   INCHAR NORMAL CONTROL CODE HANDLED
0750: 2352 60          RTS   UNUSUAL CONTROL CODE, NOT HANDLED
0760:
0770:          LIGHT PEN TEST ROUTINE
0780:
0790:          JSR LPTEST
0800:          C = 0 = LIGHT PEN NOT DEPRESSED
0810:          C = 1 = LIGHT PEN DEPRESSED
0820:          A,Y USED
0830:
0840: 2353 18          LPTEST CLC
0850: 2354 A0 0D          LDYIM IFR   READ INTERRUPT FLAGS
0860: 2356 B1 06          LDAIY VIA
0870: 2358 29 01          ANDIM $01   TEST CA2 BIT
0880: 235A F0 01          BEQ   NOLP   BRANCH IF NOT SET
0890: 235C 38          SEC   ELSE, SET CARRY BIT
0900: 235D 60          NOLP  RTS   AND RETURN
0910:
0920:          LPPREAD SUBROUTINE
0930:          TEST LP DEPRESSED AND READ DATA IF PRESENT
0940:

```

```

0950:          JSR LPREAD
0960:          C = 0 = NO DATA
0970:          C = 1 = DATA
0980:          A = LOW ADDRESS
0990:          Y = HIGH ADDRESS
1000:
1010: 235E 20 53 23 LPREAD JSR   LPTEST TEST LP DEPRESSED
1020: 2361 90 FA          BCC   NOLP  RETURN ON NOT DEPRESSED
1030: 2363 A0 0D          LPDATA LDYIM IFR   LP DEPRESSED
1040: 2365 A9 01          LDAIM $01  CLEAR FLAG
1050: 2367 91 06          STAIY VIA
1060: 2369 A9 11          LDAIM $11  LIGHT PEN LOW REGISTER
1070: 236B A0 00          LDYIM $00
1080: 236D 91 00          STAIY CRTCON SET REGISTER 11
1090: 236F B1 02          LDAIY CRTREG GET LOW VALUE
1100: 2371 38          SEC          SUBTRACT FUDGE FACTOR
1110: 2372 E9 08          SBCIM LPFIX
1120: 2374 48          PHA          SAVE ON STACK
1130: 2375 A9 10          LDAIM $10  LIGHT PEN HIGH REGISTER
1140: 2377 91 00          STAIY CRTCON
1150: 2379 B1 02          LDAIY CRTREG
1160: 237B E9 00          SBCIM $00  SUBTRACT BORROW IF ANY
1170: 237D 6D 0A 20          ADC   MEMSTR ADD MEMORY START PAGE
1180: 2380 A8          TAY          PUT HIGH INTO Y REG
1190: 2381 68          PLA          RESTORE LOW TO A REG
1200: 2382 38          SEC          SET DATA PRESENT
1210: 2383 60          RTS          RETURN
1220:

```

```

1230:          LPWAIT SUBROUTINE
1240:          WAIT FOR LIGHT PEN DATA
1250:          A = HIGH ADDRESS
1260:          Y = LOW ADDRESS
1270:

```

```

1280: 2384 20 53 23 LPWAIT JSR   LPTEST TEST LIGHT PEN DEPRESSED
1290: 2387 90 FB          BCC   LPWAIT KEEP TRYING
1300: 2389 B0 D8          BCS   LPDATA OKAY, READ THE DATA
1310:
ID=16

```

0010: AN EXAMPLE PROGRAM

```

0020:
0030: 238B 20 22 20 TRYIT  JSR   INIT  INITIALIZE EVERYTHING
0040: 238E 20 1D 23 LOOP   JSR   KBTEST TEST KEYBOARD FOR DATA
0050: 2391 B0 0A          BCS   KBRD   C = 1 THERE IS DATA
0060: 2393 20 5E 23          JSR   LPREAD ELSE, TRY THE LIGHT PEN
0070: 2396 90 F6          BCC   LOOP   C = 0 NO LIGHT PEN ACTIVITY
0080: 2398 20 57 22 LPEN  JSR   SETMEM SET NEW COORDINATES
0090: 239B 90 F1          BCC   LOOP   C = 0 ALWAYS
0100: 239D 20 39 23 KBRD  JSR   GETCHR GET AND CONVERT IF NECESSARY
0110: 23A0 20 B1 20          JSR   CONTRL TEST CONTROL CODES
0120: 23A3 B0 E9          BCS   LOOP   CONTROL CODE DONE
0130: 23A5 20 0A 23          JSR   WINC   WRITE NORMAL CHARACTER
0140: 23A8 90 E4          BCC   LOOP   C = 0 NOT END OF FIELD
0150: 23AA 20 9B 21          JSR   UPSCRL SCROLL UP AT END OF FIELD
0160: 23AD 4C 8E 23          JMP   LOOP
0170:

```

```

0180:          THAT'S ALL, FOLKS!!!
ID=

```


SYMBOL TABLE

ABSCOL	2003	ABSROW	2007	ACR	000B	BS	0008
CEOFX	2166	CEOLX	213E	CHAR	2009	CLEAR	214A
CLREOF	216A	CLREOL	2142	COLUMN	2000	CONTRL	20B1
CR	000D	CRET	20B5	CRTCON	0000	CRTREG	0002
CRTSET	22D7	CUREND	201B	CURHI	201E	CURLO	201F
CURSET	22E1	CURSTR	201A	DDRA	0003	DDRIB	0002
DECR	22B9	DELETE	007F	DNOK	21F7	DNSCRL	21ED
DONE	2271	DOWN	20BA	DOWNOK	20C7	DSAH	201C
DSAL	201D	ENDFLD	212C	EOF	0012	EOL	0011
ESC	001B	FBUMP	2157	FEND	2190	FEOF	0014
FEOFX	217E	FEOL	0013	FEOLX	2171	FF	000C
FILEOF	2182	FILEOL	2175	FILL	2186	FILLIT	214C
FIXCUR	2145	FMORE	218B	FRTN	2165	FS	001C
FWRITE	214E	GETCHR	2339	GETCUR	223E	GP	20B1
HDISP	2011	HOME	211E	HOMEX	211A	HSYNC	2012
HTAB	0009	HTOTAL	2010	HWIDTH	2013	IDRTN	22B7
IER	000E	IFR	000D	ILOOP	2063	IMODE	2018
INCHAR	2346	INCR	2291	INCRA	22AE	INCRX	22A9
INCRY	22AC	INIT	2022	KBDATA	232D	KBRD	239D
KBREAD	2328	KBTEST	231D	KBWAIT	2332	LEFT	2107
LEFTX	2103	LF	000A	LFEED	20D2	LOOP	238E
LPDATA	2363	LPEN	2398	LPFIX	2008	LPREAD	235E
LPTEST	2353	LPWAIT	2384	LPXX	2021	MAXCOL	2002
MAXROW	2006	MEMEND	200B	MEMHGH	0005	MEMLOW	0004
MEMORY	2020	MEMSTR	200A	MINCOL	2001	MINROW	2005
MSLA	2019	NODATA	2327	NOLP	235D	NORMAL	2345
NOSCRL	21E2	OFFHGH	200F	OFFLOW	200E	ORA	0001
ORB	0000	OTHER	2220	PCGEND	200D	PCGSTR	200C
PCR	000C	PERIOD	002E	RCMEM	2274	RCMEMA	2281
RCMEMB	2289	RCMEMC	228C	READ	2302	RIGHT	20F0
RIGHTX	20EC	RINC	22F6	ROW	2004	SCRDN	21E4
SCRFIN	21D6	SCRLDN	000F	SCRLOK	21BE	SCRLUP	000E
SCRUP	2192	SETCD	2229	SETCUR	225B	SETDUN	22F0
SETEND	2130	SETMEM	2257	SETOFF	2223	SETROW	22E8
SET	20CA	SPACE	0020	TMPHGH	0009	TMPLOW	0008
TOGGLE	2212	TRYIT	238B	UP	20DA	UPLINE	20D6
UPOK	20E7	UPSCA	21A4	UPSCB	21AB	UPSCRL	219B
USRINT	202E	VDISP	2016	VIA	0006	VSLA	2015
VSYNC	2017	VTAB	000B	VTOTAL	2014	WINC	230A
WRITE	2316						

Year	Month	Day	Event	Location	Notes
1870	Jan	1
1870	Jan	2
1870	Jan	3
1870	Jan	4
1870	Jan	5
1870	Jan	6
1870	Jan	7
1870	Jan	8
1870	Jan	9
1870	Jan	10
1870	Jan	11
1870	Jan	12
1870	Jan	13
1870	Jan	14
1870	Jan	15
1870	Jan	16
1870	Jan	17
1870	Jan	18
1870	Jan	19
1870	Jan	20
1870	Jan	21
1870	Jan	22
1870	Jan	23
1870	Jan	24
1870	Jan	25
1870	Jan	26
1870	Jan	27
1870	Jan	28
1870	Jan	29
1870	Jan	30
1870	Jan	31
1870	Feb	1
1870	Feb	2
1870	Feb	3
1870	Feb	4
1870	Feb	5
1870	Feb	6
1870	Feb	7
1870	Feb	8
1870	Feb	9
1870	Feb	10
1870	Feb	11
1870	Feb	12
1870	Feb	13
1870	Feb	14
1870	Feb	15
1870	Feb	16
1870	Feb	17
1870	Feb	18
1870	Feb	19
1870	Feb	20
1870	Feb	21
1870	Feb	22
1870	Feb	23
1870	Feb	24
1870	Feb	25
1870	Feb	26
1870	Feb	27
1870	Feb	28
1870	Feb	29
1870	Feb	30
1870	Mar	1
1870	Mar	2
1870	Mar	3
1870	Mar	4
1870	Mar	5
1870	Mar	6
1870	Mar	7
1870	Mar	8
1870	Mar	9
1870	Mar	10
1870	Mar	11
1870	Mar	12
1870	Mar	13
1870	Mar	14
1870	Mar	15
1870	Mar	16
1870	Mar	17
1870	Mar	18
1870	Mar	19
1870	Mar	20
1870	Mar	21
1870	Mar	22
1870	Mar	23
1870	Mar	24
1870	Mar	25
1870	Mar	26
1870	Mar	27
1870	Mar	28
1870	Mar	29
1870	Mar	30
1870	Mar	31

