ANYONE FOR TENNIS?

Make sense of machine code with this bouncey game, courtesy of Keith Hook and M Gaut.

The program can be altered to illustrate how the MTX's assembler works.

Although written for the MTX 500, this program is easily adapted for the more powerful MTX 512 (above).

The Memotech MTX's in-built assembler/debugger is ideal for programming in Basic with 'in-line' machine code subroutines. However, once Basic has been mastered, most users get the urge to write in assembler: the routines run faster and allow certain special effects which are impossible in Basic.

Unfortunately, very little has been written about getting the best from the MTX assembler, and many users abandon the idea of attempting machine code as a result.

This listing is a game of practice tennis which allows you to alter any section of the code and experiment to see how it affects the game. For example it is quite easy to create a two player version, or to alter the angle at which the ball bounces off the walls or the bat.

Even the novice programmer can experiment and gain some ideas for future use. The source listing is fully documented for those of you who have not yet used the assembler.

How it works

The code starts at 8007 Hex — this is because it was written on a MTX 500 which only has 32K RAM. MTX 512 owners should type in the code exactly as printed, but when the assembler is entered by typing ASSEM 10, the code should start at 4007 Hex. The resulting code will align with the listing, but will always be prefixed by 4 instead of 8.

RST 10 instructions are used throughout the listing (see £800B). This call is an easy way to use the MTX's built-in graphic functions. Basic screen routines use these ROM calls, and because of the way the operating system is structured, it is easier to take advantage of the routines rather than write your own.

The RST 10 instruction expects certain information to follow the call, and depending upon this, the call will follow a certain course of action. These calls can draw lines, plot lines, create sprites, move sprites, print text and graphics to the screen, and much more.

The format for using RST 10 is:

```
RST 10
DB £83
DB "TRY"
```

The above example will print TRY on the screen, at the current cursor position. The function decides which command is to be executed by the bit pattern of the first byte (£83), which is the command byte.

Bit 7 6 5 4 3 2 1 0
Patt 1 0 cs Number of bytes

Here, bit 7 is set [£80 = 1000 0000], and we are printing three characters, so RST 10 DB £83 tells the MTX to print 3 characters at the current cursor location. Bit 5 is the Carry:Stop bit. If this bit is set (1) the RST 10 call executes the immediate command and then carries on to look for another command byte:

```
RST 10
DB £A3,"TRY"
DB £86, "AGAIN"
```

The above code will result in RST 10 printing TRY AGAIN on the screen at the current cursor location before it returns to the calling program. (For a complete explanation of these calls see *PCN* issue 64).

The program uses 16 by 16 sprites and this mode is selected by setting bit 1 of VDP Register 1. In order that this register can also carry out its other functions, a value of £C2 must be sent to this register.

RST 10 followed by DB £4C will select Virtual Screen 4 and clear it on entry.

LD A,R is an easy way of getting a random number in the range 0 — 127. The Refresh register is constantly counting up to 127, so you get the same number twice in a row.

Finally, under MTX Basic, the sprite attribute table is located at 16128 decimal and each entry consists of four bytes:

Byte 1 Vertical distance from top of screen
Byte 2 Horizontal distance from left hand edge
Byte 3 Pointer to sprite number
Byte 4 Sprite colour

Listing

```
8007        LD SP,(£FA96)      ; LOAD THE STACK POINTER FROM SYSTEM VARIABLES
800B        RST 10     ;RESTART 10
800C        DB £83,16,2,2      ;SET THE BACKGROUND COLOUR TO GREEN
8010 INIT:  RST 10     ;SWITCH TO VS 4 AND CLEAR IT
8011        DB £4C
8012        RST 10     ;RESTART 10
8013        DB £A3,16,3,15     ;SET THE PLOT INK COLOUR TO WHITE
8017        DB £A3,16,4,1      ;SET THE BORDER COLOUR TO BLACK
801B        DB £83,16,0,1      ;SET THE PRINT PAPER COLOUR TO BLACK
801F        LD A,£C2   ;SELECT 16$16 SPRITES
8021        OUT (2),A  ;BY SETTING BIT 1 OF VDP REGISTER 1
8023        LD A,£81
8025        OUT (2),A
8027        CALL MESGES        ;PRINT THE MESSAGES AT THE TOP OF THE SCREEN
802A        XOR A      ;CLEAR THE ACCUMULATOR
802B        LD (HIT),A ;SET HITS TO 0
802E        LD (MISS),A        ;SET MISS TO 0
8031        LD A,100
8033        LD (BAT),A ;SET THE Y COORDINATE OF THE BAT
8036        CALL PATTERN       ;SET THE SPRITE PATTERNS
8039        LD DE,(SATSAD)     ;DISPLAY THE BAT
803D        CALL VADOUT
8040        LD A,(BAT)
8043        OUT (1),A
8045        LD A,10
8047        CALL OUT
804A        LD A,0
804C        CALL OUT
804F        LD A,15
8051        CALL OUT
8054        CALL BORDER        ;DISPLAY THE BORDER AROUND THE COURT
8057 INIT1: CALL RANDOM        ;SET BALL AT A RANDOM POSITION
805A        CALL DISSCR        ;DISPLAY THE SCORES
805D LOOP:  CALL UPDATE        ;REDISPLAY THE BALL
8060        CALL DELAY ;DELAY THE PROGRAM FOR A SHORT TIME
8063 NOGO:  IN A,(2)
8065        BIT 7,A
8067        JR NZ,NOGO
8069        CALL KBOARD        ;SCAN THE KEYBOARD
806C INCX:  LD HL,XINC ;INCREASE THE X COORD BY THE VALUE IN XINC
806F        LD A,(XCOOR)
8072        ADD A,(HL)
8073        CP 9       ;TEST FOR BALL GOING OF SCREEN
8075        JP C,SERVE
```

Listing (cont)

```
8078          CP 242     ;TEST FOR BALL HITTING RIGHT WALL
807A          JP NC,FIXI
807D          LD (XCOOR),A
8080 INCY:    LD HL,YINC ;INCREASE THE Y COORD BY THE VALUE IN YINC
8083          LD A,(YCOOR)
8086          ADD A,(HL)
8087 TESTY:   CP 173     ;TEST FOR BOTTOM OF SCREEN
8089          JP NC,FIXY
808C          CP 10      ;TEST FOR TOP OF SCREEN
808E          JP C,FIXY
8091 SAVY:    LD (YCOOR),A
8094 CONT:    CALL TEST  ;SEE IF BALL TOUCHING BAT
8097          JP LOOP    ;DO IT ALL AGAIN
809A DELAY:   LD BC,(DELT)       ;GET THE CURRENT DELAY TIME
809E DEL1:    DEC BC
809F          LD A,B
80A0          OR C
80A1          JR NZ,DEL1
80A3          RET
80A4 FIXX:    LD A,(XCOOR)       ;THIS ROUTINE NEGATES THE XINC
80A7          LD HL,XINC
80AA          SUB (HL)
80AB          LD (XCOOR),A
80AE          LD A,(XINC)
80B1          NEG
80B3          LD (XINC),A
80B6          RET
80B7 FIXY:    LD A,(YCOOR)       ;THIS ROUTINE NEGATES THE YINC
80BA          LD HL,YINC
80BD          SUB (HL)
80BE          LD (YCOOR),A
80C1          LD A,(YINC)
80C4          NEG
80C6          LD (YINC),A
80C9          JP CONT
80CC KBOARD:  LD A,&F8   ;TEST FOR UP KEY
80CE          OUT (5),A
80D0          IN A,(5)
80D2          CP &7F
80D4          JP Z,MOVUP ;IF IT IS,THEN MOVE BAT UP
80D7          LD A,&BF   ;TEST FOR DOWN KEY
80D9          OUT (5),A
80DB          IN A,(5)
80DD          CP &7F
80DF          JP Z,MOVDN ;IF YES,THEN MOVE BAT DOWN
80E2          RET
80E3 FIXX1:   CALL FIXX  ;BOUNCE BALL IN X AXIS
80E6          JP INCY    ;AND Y AXIS
80E9 VADIN:   PUSH AF    ;THIS ROUTINE SETS UP VRAM FOR READING
80EA          LD A,E     ;GET THE LOW PART OF THE ADDRESS
80EB          OUT (2),A  ;AND SEND IT DOWN PORT TWO
80ED          LD A,D     ;GET THE HIGH PART OF THE ADDRESS
80EE          AND 63     ;MAKE SURE THAT BITS 6 AND 7 ARE OFF.THIS COMBINATION TELLS
              THE VDP THAT IT IS REQUIRED TO SEND DATA BACK TO THE Z80
80F0          OUT (2),A  ;OUTPUT THIS VALUE DOWN PORT TWO
80F2          POP AF
80F3          RET        ;RETURN
80F4 SATSAD:  DW 16128   ;SPRITE ATTRIBUTE TABLE START ADDRESS
80F6 PATTERN: LD B,64    ;SET COUNTER
80F8          LD HL,DATA ;POINT HL AT PATTERN DATA
80FB          LD DE,14336        ;SET VRAM ADDRESS FOR WRITING TO THE PATTERN GENERATOR TABLE
80FE          CALL VADDUT
8101 PL1:     LD A,(HL)  ;READ A BYTE FROM THE PATTERN DATA
8102          OUT (1),A  ;AND SEND IT TO THE CORRECT PLACE IN VRAM
8104          INC HL     ;INCREMENT THE POINTER IN THE PATTERN DATA LIST
8105          DJNZ PL1   ;DECREMENT COUNTER AND DO IT AGAIN
8107          RET        ;RETURN
8108 VADDUT:  PUSH AF    ;THIS ROUTINE SETS UP VRAM FOR WRITING
8109          LD A,E     ;GET THE LOW PART OF THE ADDRESS
810A          OUT (2),A  ;AND SEND IT OUT THROUGH PORT TWO
810C          LD A,D     ;GET THE HIGH PART OF THE ADDRESS
810D          OR 64      ;MAKE SURE THAT BIT 6 OF THE VALUE IS SET
810F          AND 127    ;AND MAKE SURE THAT BIT 7 IS OFF.THIS COMBINATION OF BITS INFORMS
              THE VDP THAT IT IS ABOUT TO RECEIVE DATA
8111          OUT (2),A  ;SEND THIS VALUE DOWN PORT TWO
8113          POP AF
8114          RET        ;RETURN
8115 OUT:     OUT (1),A  ;THIS ROUTINE IS USED,BECAUSE A DELAY OF AT LEAST 8 MICROSECENDS
              IS NECESSARY BETWEEN SUCCESSIVE READS OR WRITES
8117          RET        ;RETURN
8118 MOVUP:   LD A,(BAT) ;GET THE BAT COORDINATE
8118          CP 18      ;TEST TO SEE IF IT IS AT THE TOP OF THE SCREEN
811D          RET C      ;IF IT IS THEN RETURN WITHOUT MOVING IT.
811E          DEC A      ;OTHERWISE TAKE TWO OFF THE BAT POSITION
811F          DEC A
8120          LD (BAT),A ;AND STORE NEW BAT POSITION
8123          LD DE,(SATSAD)     ;SET UP THE ADDRESS OF THE SPRITE ATTRIBUTE TABLE
8127          CALL VADDUT
812A          OUT (1),A  ;SEND THE Y COORDINATE OF THE BAT
812C          RET        ;AND RETURN
812D MOVDN:   LD A,(BAT) ;GET THE BAT COORDINATE
8130          CP 168     ;TEST TO SEE IF IT IS AT THE BOTTOM OF SCREEN
8132          RET NC     ;IF IT IS,THEN RETURN WITHOUT MOVING THE BAT
8133          INC A      ;OTHERWISE ADD TWO TO THE BAT COORDINATE
8134          INC A
8135          LD (BAT),A ;STORE IT AGAIN
8138          LD DE,(SATSAD)     ;SET UP TO WRITE TO THE SPRITE ATTRIBUTE TABLE
813C          CALL VADDUT
813F          OUT (1),A  ;AND SEND THE NEW Y COORDINATE
8141          RET        ;RETURN
8142 TEST:    LD A,(XINC)        ;TEST TO SEE IF XINC IS POSITIVE
8145          BIT 7,A    ;BY TESTING BIT SEVEN.IF IT IS OFF,THEN THE VALUE IS POSITIVE
8147          RET Z      ;IF IT IS,RETURN BECAUSE THE BALL IS TRAVELLING FROM THE LEFT TO THE
              RIGHT,AND THERFORE CAN'T HIT THE BALL
8148          LD A,(XCOOR)       ;TEST X COORDINATE OF BALL
814B          CP 22      ;IF NOT WITHIN X RANGE,RETURN
814D          RET NC
814E          CP 20
8150          RET C
8151          LD A,(YCOOR)       ;TEST Y COORD TO SEE IF BALL IS BELOW BAT
8154          LD B,A     ;OR BAT BELOW BALL
8155          LD A,(BAT)
8158          CP B
8159          JP C,TESTD ;IF THE BALL IS BELOW THE BAT,THEN JUMP TO A DIFFERENT TESTING ROUTINE
815C          SUB B      ;IF THE BAT IS BELOW THEN SUBTRACT THE BAT COORD FROM THE BALL COORD
815D          CP 12      ;AND SEE IF THEY ARE WITHIN 12 PIXELS OF EACH OTHER
815F          RET NC     ;IF THEY AREN'T,THEN RETURN
8160          JP HITB    ;OTHERWISE JUMP TO THE 'HIT BALL' ROUTINE
8163 TESTD:   LD A,(BAT) ;GET THE BAT Y COORD
8166          LD B,A     ;STORE IT IN B
8167          LD A,(YCOOR)       ;GET THE BALL Y COORD
816A          SUB B      ;SUBTRACT THE BALL COORD FROM THE BAT COORD
816B          CP 17      ;ARE THEY WITHIN 17 PIXELS OF EACH OTHER?
816D          RET NC     ;IF THEY AREN'T,THEN RETURN
816E HITB:    CALL INCHIT        ;ADD ONE TO THE HIT COUNTER
8171          CALL FIXX  ;CALL THE ROUTINE TO BOUNCE THE BALL
8174          RET        ;AND RETURN
8175 SERVE:   CALL INCMIS        ;ADD ONE TO THE MISS COUNTER
8178          LD BC,0    ;MAKE A SHORT DELAY SO THAT THE NEW BALL ISN'T SERVED IMMEDIATELY
817B SRV1:    DEC BC
817C          LD A,B
817D          OR C
817E          JR NZ,SRV1
8180          JP INIT1   ;AND SERVE A NEW BALL
8183 RANDOM:  LD A,R     ;GET A NUMBER FROM THE REFRESHER REGISTER
8185          ADD A,10   ;MAKE SURE THAT IT IS MORE THAN 10
8187          LD (YCOOR),A       ;STORE IT IN THE BALLS Y COORDINATE
818A          LD A,12    ;STORE THE X COORDINATE OF THE BALL
818C          LD (XCOOR),A
818F          LD A,1     ;RESET X AND Y INCREMENTS TO 1.THESE VALUES COULD BE CHANGED TO MAKE
              THE BALL GO FASTER RELATIVE TO THE BAT
8191          LD (XINC),A
8194          LD A,1
8196          LD (YINC),A
8199 UPDATE:  LD DE,(SATSAD)     ;GET THE START OF THE SPRITE ATTRIBUTE TABLE
819D          INC DE     ;AND ADD FOUR TO IT,TO POINT TO THE START OF THE BALL BLOCK
819E          INC DE
819F          INC DE
81A0          INC DE
```

Listing (cont)

```
81A1        CALL VADOUT       ;SET UP THIS ADDRESS FOR WRITING
81A4        LD A,(YCOOR)      ;GET THE BALL Y COORDINATE
81A7        CALL OUT  ;SEND THE Y COORD
81AA        LD A,(XCOOR)      ;GET THE X COORDINATE
81AD        CALL OUT  ;SEND THE X COORD
81B0        LD A,4
81B2        CALL OUT  ;SEND THE PATTERN NUMBER
81B5        LD A,15
81B7        CALL OUT  ;SEND THE COLOUR (WHITE)
81BA        LD A,208
81BC        OUT (1),A ;MARK THE END OF THE SPRITE TABLE
81BE        RET
81BF BORDER: RST 10     ;RESTART 10
81C0        DB £A5,2,0,7,255,7,£A5,2,255,7,255,175,£85,2,255,175,0,175 ;THIS DATA DRAWS THE
            LINES AROUND THE EDGE OF THE SCREEN
81D2        RET       ; RETURN
81D3 MESSES: LD HL,MES1 ;POINT HL AT MESSAGE 1
81D6        LD DE,£010A       ;SET COORDINATES
81D9        CALL MESS ;CALL MESSAGE PRINTING ROUTINE
81DC        LD HL,MES2 ;DO IT AGAIN,BUT USE DIFFERENT MESSAGES
81DF        LD DE,£0101
81E2        CALL MESS
81E5        LD HL,MES3
81E8        LD DE,£0117
81EB        CALL MESS
81EE        RET
81EF MESS:  LD (CORD),DE      ;STORE THE COORDINATES
81F3        RST 10    ;RESTART 10
81F4        DB £83,3  ;DATA TO SET CURSOR COORDINATES
81F6 CORD:  DW 0
81F8 MLOOP: LD A,(HL) ;GET A CHARACTER FROM THE MESSAGE
81F9        CP 0      ;IS IT ZERO
81FB        RET Z     ;IF YES,RETURN
81FC        LD (CHAR),A       ;OTHERWISE STORE CHARACTER
81FF        RST 10    ;RESTART 10
8200        DB £81    ;AND PRINT THE CHARACTER
8201 CHAR:  DB 0
8202        INC HL    ;MOVE THE MESSAGE POINTER ON BY ONE
8203        JR MLOOP  ;GO BACK AND DO IT AGAIN
8205 INCHIT: LD A,(HIT) ;GET THE NUMBER OF HITS
8208        ADD A,1   ;ADD ONE TO IT
820A        DAA       ;DECIMAL ADJUST IT FOR BCD ARITHMETIC
820B        LD (HIT),A ;STORE IT BACK AGAIN
820E        CALL DISSCR       ;DISPLAY THE SCORES
8211        LD HL,(DELT)      ;GET THE DELAY TIME
8214        LD DE,13  ;TAKE 13 OF IT
8217        AND A
8218        SBC HL,DE
821A        LD (DELT),HL      ;STORE IT BACK AGAIN
821D        LD A,(HIT) ;GET THE NUMBER OF HITS
8220        CP £10    ;IS IT 10?
8222        JP Z,WON  ;IF YES,THEN GO TO WON ROUTINE
8225        RET       ;OTHERWISE RETURN
8226 INCMIS: LD A,(MISS)       ;GET THE NUMBER IF MISSES
8229        ADD A,1   ;ADD ONE TO IT
822B        DAA       ;DECIMAL ADJUST IT FOR BCD ARITHMETIC
822C        LD (MISS),A       ;STORE I BACK AGAIN
822F        CALL DISSCR       ;DISPLAY THE SCORES
8232        LD A,(MISS)       ;GET THE NUMBER OF MISSES
8235        CP £20    ;IS IT 20?
8237        JP Z,OVER ;IF YES,JUMP TO THE GAME OVER ROUTINE
823A        RET       ;OTHERWISE RETURN
823B DISSCR: RST 10     ;RESTART 10
823C        DB £83,3,30,1     ;SET THE CURSOR COORDINATES TO 30,1
8240        LD A,(MISS)       ;GET THE NUMBER OF MISSES
8243        CALL CONV ;CONVERT IT INTO PRINTABLE CHARACTERS
8246        LD A,B    ;GET THE 'TENS' VALUE
8247        CALL DIS  ;DISPLAY IT
824A        LD A,C    ;GET THE 'UNITS' VALUE
824B        CALL DIS  ;DISPLAY IT
824E        RST 10    ;RESTART 10
824F        DB £83,3,6,1      ;SET THE CURSOR COORDINATES TO 6,1
8253        LD A,(HIT) ;GET THE NUMBER OF HITS
8256        CALL CONV ;CONVERT IT INTO PRINTABLE CHARACTERS
8259        LD A,B    ;GET THE 'TENS' VALUE
825A        CALL DIS  ;DISPLAY IT
825D        LD A,C    ;GET THE 'UNITS' VALUE
825E        CALL DIS  ;DISPLAY IT
8261        RET       ;RETURN
8262 CONV:  LD D,A    ;STORE THE VALUE IN D
8263        AND £F0   ;MASK OFF THE RIGHT FOUR BITS
8265        SRL A     ;SHIFT IT RIGHT FOUR PLACES
8267        SRL A
8269        SRL A
826B        SRL A
826D        AND A     ;AND ADD 48,TO MAKE IN INTO THE CORRECT PRINTABLE
826E        ADC A,48  ;CHARACTER
8270        LD B,A    ;STORE IT IN B
8271        LD A,D    ;GET THE ORIGINAL VALUE
8272        AND £0F   ;MASK OFF THE LEFT FOUR BITS
8274        AND A     ;ADD 48 TO IT,TO TURN IT INTO A
8275        ADC A,48  ;PRINTABLE CHARACTER
8277        LD C,A    ;STORE IT IN C
8278        RET       ;RETURN
8279 DIS:   LD (CHAR1),A      ;STORE THE CHARACTER
827C        RST 10    ;RESTART 10
827D        DB £81    ;PRINT IT
827E CHAR1: DB 0
827F        RET       ;RETURN
8280 WON:   LD HL,MES9 ;POINT HL AT MESSAGE 9
8283        LD DE,£0B08       ;SET THE COORDINATES
8286        CALL MESS ;CALL THE MESSAGE PRINTING ROUTINE
8289        CALL MESSTO8      ;PRINT MESSAGES 5 TO 8
828C        RET       ;RETURN
828D OVER:  LD HL,MES4 ;POINT HL AT MESSAGE 4
8290        LD DE,£0B09       ;SET THE COORDINATES
8293        CALL MESS ;CALL THE MESSAGE PRINTING ROUTINE
8296 MESSTO8: LD HL,MES5 ;THE NEXT ROUTINE JUST PRINTS MESSAGES
8299        LD DE,£0A0A       ;5 TO 8
829C        CALL MESS
829F        LD HL,MES6
82A2        LD DE,£0B0C
82A5        CALL MESS
82A8        LD HL,MES7
82AB        LD DE,£0D09
82AE        CALL MESS
82B1        LD HL,MES8
82B4        LD DE,£0E0C
82B7        CALL MESS
82BA KLOOP: CALL £0079 ;GET A CHARACTER FROM KEYBOARD
82BD        CP 0      ;IS ONE BEING PRESSED?
82BF        JR Z,KLOOP ;IF NOT,JUMP BACK AND TRY AGAIN
82C1        CP 13     ;IS IT THE RETURN KEY
82C3        JP Z,INIT ;IF YES,START THE PROGRAM AGAIN
82C6        CP 3      ;IS IT THE BREAK KEY?
82C8        JP Z,END  ;IF YES,JUMP TO THE END
82CB        JR KLOOP  ;OTHERWISE JUMP BACK AND TEST AGAIN
82CD        RET       ;RETURN
82CE END:   LD SP,£FD46       ;POINT THE STACK POINTER AT THE CALLING ADDRESS
8201        RET       ;AND RETURN TO IT.
8202 DATA:  DB 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,0,0,0,0,3,4,-
            8,8,8,8,4,3,0,0,0,0,0,0,0,0,192,32,16,16,16,16,32,192,0,0,0,0
8312 DELT:  DW 1500   ;AS THIS VALUE BECOMES SMALLER,THE GAME SPEED WILL INCREASE
8314 XCOOR: DB 6      ;THIS IS THE XCOORDINATE OF THE BALL
8315 YCOOR: DB 0      ;THIS IS THE Y COORDINATE OF THE BALL
8316 XINC:  DB 1      ;THIS IS THE VALUE BY WHICH THE X COORD OF THE BALL INCREASES  WHEN
             IT MOVES
8317 YINC:  DB 1      ;THIS IS THE VALUE BY WHICH THE YCOORD INCREASES EACH TIME IT MOVES
8318 BAT:   DB 0      ;THIS IS THE BAT POSITION,MEASEURED IN PIXELS FROM THE TOP OF THE SCREEN
8319 HIT:   DB 0      ;THIS IS THE HIT COUNTER.
831A MISS:  DB 0      ;THIS IS THE MISS COUNTER
831B MES1:  DB "T E N N I S"  ;MESSAGE 1
8326        DB 0
8327 MES2:  DB "HITS:",0      ;MESSAGE 2
8320 MES3:  DB "MISSES:",0    ;MESSAGE 3
8335 MES4:  DB "£££GAME  OVER£££",0   ;MESSAGE 4
8346 MES5:  DB "PRESS <RET> TO",0     ;MESSAGE 5
8355 MES6:  DB "PLAY AGAIN",0 ;MESSAGE 6
8360 MES7:  DB "PRESS <BREAK> TO"     ;MESSAGE 7
8370        DB 0
8371 MES8:  DB "TERMINATE",0  ;MESSAGE 8
837B MES9:  DB "WELL DONE, YOU WIN",0 ;MESSAGE 9
838E        RET
```