MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542

ISSUE 3
FEBRUARY 1980 PRICE $2.50

CONTENTS

MICRO-80 is registered for posting as a publication - CATEGORY B.

Printed by: The Shovel and Bull Printers,
312 A Unley Road, Hyde Park, SA 5061

All enquiries should be sent to :-
MICRO-80, P.O. BOX 213, GOODWOOD SA 5034
'Phone: (08) 381 8542

MICRO-80

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542

** ABOUT MICRO-80 **

MICRO-80 is the first Australian monthly magazine devoted entirely to the TRS-80 microcomputer. It is available by subscription $24.00 for 12 months or by mail order at $2.50 per copy. A cassette containing all the programs in each month's issue is available for an additional $3.50 or a combined annual subscription to both magazine and cassette, is available for $60.00. Special bulk purchase rates are also available to computer shops etatc. Please use the form in this issue to order your copy or subscription.

The purpose of MICRO-80 is to publish software and other information to help you get the most from your TRS-80 computer and its peripherals. MICRO-80 is in no way connected with the TANDY organisation.

** WE WILL BUY YOUR PROGRAMS **

Most of the information we publish is provided by our readers, to whom we pay royalties. An application form containing full details of how you can use your TRS-80 to earn some extra income is included in every issue.

** CONTENT **

Each month we publish at least one applications program in Level 1 BASIC, one in Level 2 BASIC and one in DISK BASIC (or Disk compatible Level 2). We aslo publish Utility programs in Level 2 BASIC and Machine language. At least every second issue has an article on hardware modifications or a constructional article for useful peripherals. In addition, we run articles on programming techniques both in Assembly language and BASIC, we print news from TRS-80 Users Clubs, letters to the Editor and new product reviews.

** ADVERTISING **

We accept camera ready copy for display advertising at the following rates:

- FULL PAGE (19cm wide x 24cm high) $120
- 1/2 PAGE (19cm wide x 14cm high) $ 60
- 1/4 PAGE (19cm wide x 7 cm high) $ 30

Classified ads are $8.00 for up to 50 words. Ads must be submitted by the 15th of each month in order to appear in the following month's issue. A Company Order or payment must be included with the advert.

** TRS-80 USERS CLUB NEWS **

We are prepared to print news of the activities of TRS-80 Users Clubs up to a maximum of 200 words per Club per month, space permitting. Copy must be TYPED with DOUBLE LINE SPACING and reach us NO LATER than the 15th of each month in order to appear in the following month's issue.

** COPYRIGHT **

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542

***** EDITORIAL *****

We are delighted to announce that Peter Hartley and Eddy Paay, two of our most regular contributors, have joined us as Associate Editors. Eddy will concentrate on Machine Language and Disk BASIC programs whilst Peter will concentrate on Level 1 and Level 2 programs. Both will help with hardware articles, answering readers letters etc.. We are very happy that Peter and Eddy have joined us not only because they are nice blokes and very competent but also because it will help us to achieve some of our objectives which, until now, we have not had the time to do. In particular, we want to give much more explanation about the inner workings of the programs we publish. This should help you in writing your own programs and make the magazine even more useful. Secondly, we should now be able to ensure that the programs written by our own staff are much more thoroughly debugged. Those of you who tried to use MONITOR in BASIC, published last month, will welcome that piece of news! Finally, a commercial, both Eddy and Peter are very experienced TRS-80 programmers and both are willing to do contract programming at very reasonable rates. So, if you want assistance in developing a program, you could do much worse than to get in touch with one or the other of them. (Use MICRO-80's address initially).

At the suggestion of several of our readers, we have introduced a new feature this month called - Reader's Requests. It is a list of articles, information, reveiws, programs etc., which you, our readers, have written in asking us to include in future issues. Why not send in your own requests? If you are thinking of writing a program or article for us to publish look at this list first and see what is needed. Incidentally, when you do write in for any reason, why not tell us what equipment you have (i.e. L1/4K, L2/16K etc.) that will help us assess the type of programs we should be publishing, too.

** ABOUT EDUCATIONAL PROGRAMS **

We have had several educational programs submitted to us for publication. Unfortunately, we have had to reject them all so far. It only seems fair for us to state our policy in this area, so that all our readers know where we stand.

Firstly, we believe the role of microcomputers as educational tools is a vast and important one as yet largely unfulfilled. However, there is a number of important points to consider:

- most young children have never had an encounter with a computer and, if their first one is not pleasant, it could condition their future attitudes to their long term detriment in a world where computers will become more and more common.

- One of the most common means of learning is by trial and error. The role of the teacher is to assist the student to learn from mistakes as well as from success. Telling a student that he is a "DUMMY" because he gets a problem wrong, must violate just about every rule in the teacher's book. We would go so far as to say that a program which simply rejects wrong answers is not a teaching program at all but simply a grading or testing program. One that abuses the student for a wrong answer is totally unacceptable.

- If you want to develop a REAL teaching program and you are not a teacher yourself, why not find someone who is, to work alongside you. If there is a teacher reading this column, perhaps he or she would like to write

an article explaining the fundamentals of developing good teaching programs

- Above all, teaching programs should be interesting. In fact, one of the tests we carried out on the programs submitted for consideration, was to try them out on the Associate Editorial offspring. The "wrong Dummy" type of programs held their attention for only a few minutes. Whereas, Hangman, particularly when they were entering the words themselves, held their attention for hours. This is a test we shall continue to use in assessing the worth of educational programs for the young.

Finally, don't get us wrong, we would dearly love to be involved in the publication of good educational programs. But they must be first rate and really educational before we will publish them.

** ABOUT OUR LIGHT PEN **
Last month, we told you that our prototype light pen was proving unreliable and that we would delay publishing it until we had it really sorted. That disappointed several of our readers. There are also several commercial pens being sold which work on the same principle as our prototype. Peter Hartley has tried at least one of these and made an improvement on it, so we asked him to write a constructional article describing how you can make your own light pen for well under $5.00. We will still develop an improved design, for publication in a few months time, but for those who can't wait, you will find Peter's Simple Light Pen article in this issue. The secret to this type of pen is the software, which is where MICRO-80 can really help you and, from time to time, we will publish programs to use with your light pen.

** THE ANALOGUE CLOCK **
Last month we promised you an analogue clock program, ie. one that has a dial like a normal clock. The special feature of this clock is that it uses this month's SET 2 program to redraw the hands very rapidly. This month has turned into a real bumper issue and we ran out of space, so something had to be held over and that's the analogue clock. Look for it next month, without fail.
-00000-

***** NEXT MONTH'S ISSUE *****

The March issue of MICRO-80 will contain at least the following programs:

LEARNING NIM (L1)	The well known match stick game but, in one mode, the computer actually learns how to improve its game. This is a demonstration of a simple form of artificial intelligence.
LUNAR LANDER (L1)	If Apollo had been as difficult to control as this landing module, there would be even more craters on the moon! You fly blind using the instrument panel on the screen to guide you to almost certain destruction!

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542

RICOCHET (L2)	For all those Level 2 owners who thought converting a L1 program to run in L2 was a piece of cake. This is the same game described last month but this time it will run on a L2/4K machine.
INVADERS (L2)	Invaders is reputedly the most popular of the electronic arcade games. Now you can save 20cents each time you run it on your TRS-80. Get the invaders with your laser cannon before they get you.
TWO RANDOM NUMBER (L2) GENERATOR TESTS	These programs will enable you to test the randomness of your TRS-80's random generator. More than that though, it demonstrates how to draw bar-charts in BASIC.
ANALOGUE CLOCK (L2)	This clock program has been held over from the February issue due to lack of space. It features a very realistic clock face and the hands are redrawn very rapidly using the SET 2 command published this month.
BMON - Part 2 m/c language	Part 2 of this great utility program. Eddy hasn't told us yet which commands you will get in Part 2 but you know they will be useful.
ABBREVIATED ABBREVIATIONS IN LEVEL 1	Charlie Bartlett explains how to squeeze a quart into a pint pot on Level 1 machines using some abbreviations that even Tandy didn't know they had.
DOUBLE THE STORAGE CAPACITY OF YOUR DISKS.	If you have a Tandy Disk drive, you are probably only using one side of each disk. This article gives you full instructions and a template to enable you to use both sides of the disk. No modifications are required to the disk drives. This article alone will pay for your MICRO-80 subscription several times over!

March's issue will also contain the next installment on Assembly Language Programming.

-00000-

MICRO-80

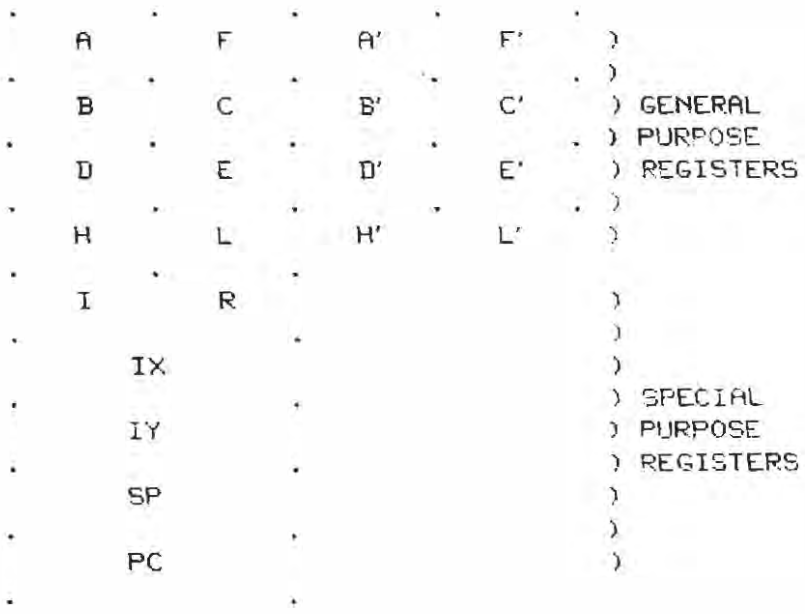
P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542

***** ASSEMBLY LANGUAGE PROGRAMMING - Part 2 *****

Last month, we described the memory mapping of the TRS-80 and the various types of memory used. We also introduced Hexadecimal arithmetic and the ASCII code for representing alphanumeric symbols. This month we will look inside the Z 80 Microprocessor itself to see how it is arranged (its architecture) and how to use it (its addressing modes). As you start to get into assembly language programming yourself, you will find it useful to have at least one reference book which gives you a list of all the Z 80 instructions, addressing modes etc. Tandy sells a book called "TRS-80 Assemby Language Programming". It is only $3.95 and is good value. Because it has been written for the TRS-80, it is also very relevant to your computer. We recommend that you buy this book to augment the information in this series of articles.

** Z 80 Architecture **

The key to the functioning of the Z 80 CPU is its register set. A register is really a memory location in the CPU. It is similar to the memory in a "memory calculator". The Z 80 has 22 registers, 17 of which contain 8 bits, 4 contain 16 bits and 1 contains 7 bits. These are shown disgrammatically below:-



The wide block in the diagram contains the 16, 8-bit, general purpose registers. These are subdivided into two blocks of eight registers each. The registers A, F, B, C, D, E, H and L are known as the MAIN register set whilst A', F', B', C', D', E', H' and L' are called the ALTERNATIVE register set or sometimes, the primed registers, since the symbol ' is pronounced "prime". The main register set is the one most commonly used, the alternative register set can only be accessed by two instructions, which exchange the contents of the main set with the alternative set. Only one of these two sets of registers can be active at any one time. It is not necessary to use the alternative set at all indeed, Level 2 BASIC never uses the alternative set although Disk BASIC does. Of the 16 general purpose registers, A and F are the two most important. A is sometimes called the "ACCUMULATOR" and many operations carried out by the Z 80 must use the contents of the A register as one of the operands with the other operand coming from memory, or one of the other registers, at the programmers discretion. F is usually called the FLAG register because the individual bits in it are used by the Z 80 to indicate

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542

the status of the CPU before and after operations. F is never used by the programmer in computations. Instead, it is automatically set by the results of computations in the other registers. The remaining general purpose registers can be used either as 8-bit registers or as 16-bit register pairs, in which case B is always paired with C, D with E and H with L to give BC, DE and HL.

The remaining 8-bit register is the Interrupt register I. The Z 80 has a very sophisticated interrupt handling capability which enables external devices to take over control of the computer when they need to. For example, you might want to use your TRS-80 as a burglar alarm as well as the computer. In that case, you could arrange the window and door switches to connect to the computer through an input/output port so that, when someone opens a window, a bit in the port is set, the computer senses this and operates an alarm, via another port. That would be fine, but if the TRS-80 spent all its time sitting looking at the switch port waiting for the occasional burglar you would have a pretty expensive relay. The way to handle this situation is to use the Z 80's interrupt handling capability. The operation of one of the burglar alarm switches would be sensed by one of the interrupt lines NMI (non-maskable interrupt - pin 17 on the Z 80) or INT (maskable interrupt - pin 16 on the Z 80). When an interrupt is received, the computer finishes its current instruction then proceeds to deal with the interrupt (unless the acceptance of a maskable interrupt on pin 16 has been inhibited by the programmer). It does this by jumping to a location in memory where a program is stored that will handle the interrupting device (in the case of the burglar alarm, the program will presumably ring a bell through an output port). After dealing with the interrupt, the Z 80 returns to the point in the program where it left off and proceeds as if nothing had happened. The I register is used to tell the Z 80 where abouts in memory it will find the interrupt handling program (for mode 2, maskable interrupts only). The whole subject of interrupt handling on the Z 80 is a fairly complex one and might be the subject of a separate article later on.

The 7-bit register is the R register. It is used to constantly refresh dynamic RAM's of the sort used in the TRS-80. These RAM's need to be refreshed by reading their contents every 2 milliseconds. The R register does this automatically during a time when the Z 80 is not accessing memory. The R register is not used for writing programs for the Z 80.

Of the four 16-bit registers, IX and IY are INDEX registers. They are used in the indexed addressing mode. This is a powerful mode of addressing particularly when tables of data are stored in memory. Indexed addressing will be described in detail in a later installment.

The other two 16-bit registers are SP, the STACK POINTER and PC the PROGRAM COUNTER. SP must always point to a free area in RAM which can be used for storage of values whilst the computer is executing a program. Trouble occurs if you let the SP get away from you so that it points to the program area or ROM or, if your program overwrites the stack. The program counter PC controls the order in which instructions are executed. While one instruction is being executed, PC always contains the address of the next instruction. This is done automatically by the Z 80 which adds the number of bytes in the instruction it has just FETCHED to the contents of the PC. When a jump is executed, the new memory location for the jump is forced into the PC so the PC causes the Z 80 to go to that new location.

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542

** DATA TRANSFERS **

The Z 80 terminology for transferring data backwards and forwards between a memory location and a register in the CPU is simpler than in most other computers which use terms like LOAD, STORE and MOVE depending on where the data originates from and where it ends up. Z 80 programmers use only one term for transferring data LOAD, which is abbreviated into mnemonic form as LD.

The direction of the data transfer is determined by the order in which the operands appear, whilst the nature of the data depends on the presence or absence of brackets.

Eg. LD A,(45)

means:- load the contents of memory address 45 into register A.

but LD (45), A

means:- load memory address 45 with the contents of register A.

The brackets around 45 indicate that 45 is the address of a memory location. If there were no brackets viz. :

LD A, 45

then the value 45 would be loaded into the A register.

The rule concerning brackets is a fundamental one in describing the Z 80 instruction set and programming in Assembly Language. Whenever brackets enclose an operand it means that that operand specifies an address, not a data value. This also applies to the movement of data between two registers:

Eg. LD A, B

puts the value in register B into register A.

LD A, (HL)

puts the value at the memory address represented by the value in the HL registered pair, into register A.

** USING THE REGISTERS **

As was stated earlier, the Accumulator or A register is the most important. All 8-bit logical and arithmetical operations require that the A register contains one of the operands and the result is always left in A. A number of the instructions which fetch or store a byte in memory also only allow A to be used. These operations could be carried out using another register but this would require an additional instruction.

The HL register pair is also very important. It acts as the Accumulator for 16-bit arithmetical operations (there are no 16-bit logical operations). All 16-bit operations use HL as one of the operand registers and the result is left in HL. Its other main use is to point to a memory address which contains the value required in an 8-bit operation. The BC and DE pairs can also be used this way but there are many more Z 80 instructions which involve HL.

Register B and register pair BC are often called the COUNT registers because they are frequently used to hold a count of the number of times an opertion is to be repeated, (rather like using a FOR/NEXT loop). For example, B is used in the instruction DJNZ which is the mnemonic for "Decrement B then Jump to the memory location specified if B is Not Zero". The BC register pair is used to indicate the length of a block of data which is to be moved from one memory location to another using the block transfer instructions LDI, LDIR, LDD etc. The C register is also the only register used for certain input and output operations.

The DE register pair can be used in much the same way as the HL pair but there are fewer instructions that use it. DE and, for that matter BC, can be used to specify addresses but only the accumulator HL can actually be involved in

the transfer.

Eg. LD A,(BC)
and LD (DE),A are valid

LD H,(DE) is not valid
but LD H,(HL) is valid

The flag register F must never be used to hold data. As stated earlier, it contains several bits that are set according to the results of other calculations. F is an 8-bit register but there are only 6 flags of which 4 are commonly used:- the carry flag C, the parity/overflow flag P/V, the zero flag Z and the sign flag S. The flags are arranged as follows:-

BIT	FLAGS
0	Carry
1	Subtract
2	Parity/overflow
3	Not used
4	Binary Coded Decimal half carry flag
5	Not used
6	Zero
7	Sign

The carry flag is set (ie. becomes equal to 1) whenever an ADD instruction produces a result which is one bit too large to be contained in a single register. The analogy in decimal arithmetic would be 7+8 = 15 ie. 5 CARRY 1. If you only had one column in which to write your result, you would need a carry flag to indicate that you had added 7 and 8 not 2 and 3 (say). The carry flag is also set when a subtraction produces a negative result or a borrow. The carry flag is also used by programmers when multiplying or dividing, since these operations can only be carried out in software - the Z 80 only performs addition and subtraction. The carry flag is also affected by shift and rotate instructions and it is cleared or set to zero, by logical operations.

The Parity/overflow flag has a dual purpose. When used to test parity, the flag is set to represent odd parity in the result of an operation. Strangely enough, odd parity occurs when the sum of the 8 bits of the result is even. The flag is reset on even parity ie. when the sum of the 8 bits of the result is odd. When used as an overflow flag, the flag is set if, after adding or subtracting two numbers of like signs (+ve or -ve) the sign of the result changes, indicating that the result is too large to be held in the register or register pair involved.

The Zero flag is only set if the result of certain is zero. These instructions are mainly concerned with arithmetical, logical and shift operations.

The sign flag is set if the result of certain instructions are negative and reset if they are positive. It has the same value as the sign bit (bit 7) in the accumulator.

It may seem, from the above that there is some duplication between the various flags. This is only true insofar as the concept is concerned. In practice, certain flags are affected only by the results of certain instructions and there is little if any duplication. To fully understand the operation of the flags, you will need to read the Z 80 data sheets.

** ADDRESSING MODES **
The Z 80 has 10 addressing modes :

1. Implied addressing
2. Immediate addressing
3. Extended immediate addressing
4. Register addressing
5. Register indirect addressing
6. Extended addressing
7. Modified page zero addressing
8. Relative addressing
9. Indexed addressing
10. Bit addressing

1. Implied Addressing
In this kind of addressing, the operation code of the instruction has no options and it always does exactly the same thing:
Eg. LD, SP, HL
This instruction takes the 16-bit contents of the HL register and transfers them to the SP register. The contents of the HL register remain unchanged and no flags are affected.
2. Immediate Addressing
In the immediate addressing mode, one of the bytes in the instruction itself is moved into a register. The data is IMMEDIATELY available.
Eg. ADD A,55
55 is added to the value in register A and the result is stored in A. The previous contents of A are lost.
3. Extended Immediate Addressing
This is the same as the above mode except that two bytes of data are required as an operand:
Eg. LD IY, FF00
In this case the Y index register is loaded with the value FF00H
4. Register Addressing
In this case one register is loaded with the value present in another one:
Eg. LD A,B
The value in B is loaded into A.
5. Register Indirect Addressing
This is similar to the previous mode but the data to be transferred is in the memory location pointed to by a register pair:
Eg. LD A,(BC)
This is a relatively inefficient way of transferring data since the register pair must first be loaded with the memory address required before executing this instruction. It is really a hang-over from the earlier 8080 instruction set and the Z 80 has more efficient methods of accessing data in memory
6. Extended Addressing
Sometimes called direct addressing, the instruction contains the address of the operand. This mode requires fairly long instructions, 3 or 4 bytes but any location in memory can be directly addressed:
Eg. LD A,(FF00)
A is loaded from location FF00H
7. Modified Page Zero Addressing
This mode applies only to the Restart Page Zero instruction - RST p. Its effect is to cause a branch to one of 8 page 0 locations after pushing the current contents of the program counter onto the stack. Page 0 is the area of external memory which can be addressed in 8 bits. Since 2 to the power 8 = 256, page 0 consists of memory locations 0 to 255 dec.

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542

Eg. RST 10

Control passes to memory location 10H.

8. Relative Addressing

This mode applies only to the jump relative (JR) instruction. The value of the byte in the instruction is added to the PC counter to obtain the branch address. Relative addressing allows the programmer to jump to ony address within -126 to +129 bytes of the current address.

Eg. JR 30

Jumps to the address 30 bytes after the current one. The advantage of relative addressing is that it uses only one byte for the address and code is position independent ie. a program which uses only relative addressing can be positioned anywhere in memory and will execute satisfactorily.

9. Indexed Addressing

The address of an operand is determined by adding a displacement byte to the value contained in one of the index registers IX or IY. These instructions are three or four bytes long:

Eg. LD (IX+20),30

loads the 30 into the memory location 20 bytes greater than that in IX.

10. Bit Addressing

Using this mode, it is possible to set (make equal to 1), reset (make equal to 0) or test an individual bit in one of the registers:

Eg. RES 2,C

bit 2 in register C is reset to 0

Well, that's all for this month. Next month we will start looking at the Z 80 instruction set in some detail. Also, from now on we will illustrate the text with some simple programs for you to try.

- 00000 -

***** LEVEL 1 INKEY SIMULATOR - by Charlie Bartlett *****

Those Level 1 owners who entered the game of RICOCHET in last month's issue, had the pleasant surprise that it simulated the INKEY function available on Level 2 and gave them a game that didn't stop and start waiting for input lines. For those of you who would like to incorporate this function into games of your own, here is a detailed explanation of the program lines that create INKEY and the different uses to which it can be put.

Using the game RICOCHET to examine the program lines to start with. Line 5 supplies the co-ordinates for 2 blocks to be set on the screen.

```
5 CLS: R=0: A(I)=0: Q=2: Z=1: S=4: T=40: U=1: V=43: GOS.3000
```

The variables underlined are the ones involved. Line 30 sets the two blocks which will act as non-interupt switches.

```
30 SET(S,T):SET(U,V):PRINT AT 798;"RICOCHET";
```

Lines 86 and 87 look to see which blocks are on and which are off. The format of these lines can change somewhat, depending on the program that they are built into. We will look further into this later.

```
86 IF(POINT(S,T)=0)*(POINT(U,V)=1) GOSUB 1000
87 IF(POINT(S,T)=0)*(POINT(U,V)=0) GOSUB 2000: SET(U,V)
```

Line 88 is the most important line of them all.

```
88 PRINT AT 832;" ";:IF Z=1 GOTO 96
```

The statement (PRINT AT 832:" ";) is essential and must appear in the program AFTER ANY OTHER "PRINT AT" STATEMENT. So, if you have, for example, five other PRINT AT statements, you must have 5 PRINT AT 832;" "; statements as

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 3818542

well.
Eg.:-

```
100 X=100
110 PRINT AT 50;"HELLO";
120 PRINT AT 832;" ";
130 IF Z=9 B=2
140 PRINT AT 80;"TRS-80";
150 PRINT AT 832;" ";
160 etc. etc.
```

What line 88 and others like it (if entered) do, is hold the Cursor at PRINT AT location 832 about 99% of the time. The small fraction of time the cursor is away, you won't notice.

We now have one graphic block set at X axis 4, Y axis 40 which is PRINT AT location 833, this places the cursor touching the left hand side of the block. If you now press any alphanumeric key, the key character will be printed at 833 thus blanking out the graphic block and causing line 86 to branch. The second graphic block is set at X axis 1, Y axis 43 which is PRINT AT location 896 which is directly beneath the cursor. If you now press the ENTER key the cursor will move down one line and knock both blocks out thus making line 87 branch.
Line 150 is the end loop back to line 50. Note that the SET command for the blocks is outside this loop. If they were not, the blocks would be set before line 86 and 87 could act.
All other graphic action to do with your program should, where possible, take place above Y axis 40 as pressing the ENTER key blocks out all graphics on that line, also steps should be taken to protect the simulator blocks from interference from your program, for example:-

```
        "WITH A MOVING DOT"
120 IF(Y)=39)*(X<=5)RESET(X,Y): X=X+1: Y=Y-1: SET(X,Y)
```

otherwise the graphic action of your program could trigger off line 86 and 87. As the simulator stands,, it can provide two functions:- press the (A)=(alphanumeric) key = move left or stop or up
press (ENTER) key = move right or start or down
With a small change to lines 86 and 87 we can increase the functions, for example
Press "A" key once = move left
Press "A" key again = move right
Press "ENTER" once = move up
Press "ENTER" again = move down
This is achieved by altering lines 86 and 87 as follows

```
85 PRINT AT 832;" ";
86 IF (POINT (S,T)=0)*(POINT(U,V)=1)*(Z=1)GOS.1000
87 IF (POINT (S,T)=0)*(POINT(U,V)=1)*(Z=2)GOS.1500
88 IF (POINT (S,T)=0)*(POINT(U,V)=0)*(B=1)GOS.2000
89 IF (POINT (S,T)=0)*(POINT U,V)=0)*(B=2)GOS.2500
90 IF Z-1 RESET(X,Y):X=X-1:SET(X,Y)
95 IF Z=2 RESET(X,Y):X=X+1:SET(X,Y)
100 IF B=1 RESET(X,Y):Y=Y-1:SET(X,Y)
105 IF B=2 RESET(X,Y):Y=Y+1:SET(X,Y)
110 GOTO 50
```

SUBROUTINES

```
1000 IF Z=1 Z=2: SET(S,T): SET(U,V): RET.
1500 IF Z=2 Z=1: SET(S,T): SET(U,V): RET.
```

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542


```
2000 IF B=1 B=2: SET(S,T): SET(U,V): RET,
2500 IF B=2 B=1; SET(S,T); SET(U,V); RET,
```

As you can see there are lots of possibilities, in fact if your program is only using alphanumeric characters in the display you can add a third graphic block to the simulator and use the "CLEAR" key as your extra input. Your branching control lines would then look like this:-

```
"Part of Imaginary Target Game"
80 IF (POINT (A,B)=0)*(POINT(C,D)=1)*(POINT(E,F)-1) GOS.1000
85 IF (POINT(A,B)=0)*(POINT(C,D)=0)*(POINT(E,F)=1) GOS.2000
90 IF (POINT(A,B)=0)*(POINT(C,D)=0)*(POINT(E,F)=0) GOS.3000
100 GOTO 50

1000 REM:PRESS "A" KEY: MOVE FIRING GRID LEFT
1005 X=X-1; RET.
2000 REM: PRESS "ENTER" KEY: MOVE FIRING GRID RIGHT
2005 X=X+1: RET.
3000 REM: PRESS"CLEAR" KEY: FIRE GUN
3005 K-K-1: RET.
```

With a little experimenting you will find all sorts of combinatons and now you can program your TRS-80 to play "real time" games such as time bombs, target games, beat the clock type games and many others.
One last helpful tip, if you press the "A" key or the "ENTER" key and nothing happens, it's a sure sign that you haven't got enough print at 832;" "; statements in your program and the cursor is off floating around the screen instead of being where it MUST be. Remember that, and you should have loads of fun.

-00000-

***** READER'S REQUESTS *****

This column will become a regular feature of MICRO-80. In it, we will list all those articles, programs etc., requested by our readers. We will then do our best to work our way through them in the coming months. If you are thinking of contributing an article or program, have a look at this list first, it will give you an idea of what our readers want.

** ARTICLES **
File handling on the TRS-80
Book reviews
Review of printers for the TRS-80
Reviews of commercially available software and hardware
Review of Chess programs
Description of the functions performed by the Expansion Interface

** SOFTWARE **
GAME OF LIFE relocated to start at 7000H
RICOCHET for L2/4K machines
M/c language program to use BREAK key like RESET when Expansion Interface is connected.

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542

** HARDWARE **
RS232 printer Interface
Interfacing the TRS-80 to external hardware
Lowercase modification

- 00000 -

***** BETTER BYTES *****

Hints from contributors to help us all

TO SAVE memory space, when writing print statements, use the right and downward arrow keys, rather than using multiple spaces. Each space uses one byte. So does each push of the arrow keys, and it gets you a lot further!

TO LISTEN to what's going on, put a transistor radio on the top of the keyboard (if you haven't got the numeric keypad fitted). You can hear data going to and from the cassette, and on those long jobs, the change in note will call you back when the job is done. If you have the keypad fitted, you can fix a "telephone tapper" coil to the circuit board with Sealastic, and connect it to a small amplifier.

DISK PROBLEMS are often caused by having the drives beside and to the left of the Video Monitor. There's a lot of magnetism from the flyback transformer, on the left of the T.V. which can weaken or even erase the data on disks.

ELECTROLUBE are now marketing a cleaner for "wiping switch contacts". This is a slightly abrasive, silicone impregnated card, which can easily be cut into pieces small enough to get at those noisy keyboard contacts. When the contacts are clean, the card stops changing colour, and it seems to last a long time.

MICRO-80 invites readers' contributions to this column. The best tip each month will be rewarded with a one-month extension to the appropriate subscription.

-00000-

*** DEF-USR-ING THE GAME OF LIFE *** by Peter Hartley.

We have received a few letters from readers experiencing some difficulty with the machine language part of LIFE (Micro-80, Jan '80). In all cases they have loaded a tape dump of the machine language routine and attempted to run it with "/ >> ENTER <<". This will not work. It was never intended that it should!

Line 10 of the Basic listing, that accompanied the m/c lang. listing, POKES the entry address for the m/c lang. program into the memory locations that are accessed by the USR function in BASIC. The last line of the Basic listing is a loop that calls the m/c lang. routine with "X = USR(0):"

This simple instruction forces the computer to store its present program location, and then to load the P.C. register with the entry address. POKED

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542

into 16526 and 16527 by line 10. When the computer encounters the final RET in the m/c lang. routine, it jumps back to line 210, and continues to run in BASIC.

In point of fact, the printed instructions for making your TBUG or other monitor dump of LIFE, gave a starting address of 0000. This is right at the start of ROM. Not because it is the correct address, but simply because we didn't need to put one in. therefor "/ >> ENTER << " will put the '80 into power up routine. The actual start address is 20798, or 513EH. The least significant byte (3E) converts to decimal 62 which is POKED into 16526, and the most significant byte (51) convverts to decimal 81 and this is POKED into 16527. To access this m/c lang. routine directly, therefore, would require either adding the start address of 513EH to the TBUG dump, or the use of "/ 20798 >> ENTER <<"

Disk Basic allows up to ten USR functions to be loaded into high memory at any one time, but Level II basic only allows one. I'm working up another article, probably for the April or May issue, that will explain how to write your own m/c lang. routine, using EDTASM, that allows you up to 254 USR calls in Level II!

I trust that this will have put a few people back on the rails of sanity? There certainly is nothing worse than poring over a printed Hex dump, and comparing it with your own on the screen, looking for an error that simply doesn't exist.

-00000-

***** SOFTWARE SECTION *****

The listings for these programs are at the back of the magazine. This section explains what each program is about and how to use it. A word about the typeface. The various members of the Editorial staff have taken to writing their contributions on their TRS-80's, using the Electric Pencil word processing program. We then transfer them around on tape and disk (the contributions, not the Editorial staff!). Unfortunately, Eddy has not yet fitted a lower-case modification to his machine, so all his articles are in upper-case only. We hope you won't mind.

** HANGMAN L1/8K with two 4K options. ** by Peter Hartley

This is a Level 1 version of the Hangman game published in last month's edition, although working with the lack of string handling facilities on Level 1 has left it barely recognisable.

This version needs about 8k to fit in all the features that Peter has included, but readers with only 4k can get two separate versions of the game out of the published listing by carefully following the directions in the REM lines at the beginning. One version allows the user to enter words and phrases of up to 15 characters, spaces and hypens, while the other lets the computer select from information stored in Data lines.

Both options are available in the full 8K version.

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542

The graphics from Bernie Simpson's Level 2 program have been retained, while Peter has added a visual "drop on the rope", if you fail to beat the game, using "point" graphics. Although this facility is slow, it adds an interesting twist to an old favourite. Unfortunately this added attraction has to be edited out to make the 4K versions.

The use of data statements for establishing the string value of entered keys, is an approach which both speeds execution and saves precious memory space. Users who want to add their own selection of words may need to change the random generator in the automatic selection routine. Each word is separated by data 99, while spaces can be entered with zeros and hyphens with data 27. The text data needs to be started and ended with Data 99.

The A array is used from A(0) through to A(57): A(0) to A(10) are used much in the way that calculator memories are employed: A(4) and A(7) are used to sense and score correctness of guesses: A(3) controls the selection of graphics routines: A(5) is the value entered by the player: A(11) to A(30) are used to store the values of the word being guessed: A(31) through to A(57) are used as toggles to check if a letter has previosuly been entered, so that the player is not penalised for entering the same letter twice: A(8) stores the word length: A(9) stores the print position for the next wrong letter selected.

Peter has again evolved some interesting formulae (you may remeber the scoring analysis from "Frustration" in last month's issue!). This time formulae have been used to centre the display of the word or phrase being guessed, and locate the correct print position for each letter as it is correctly selected.

As listed, the full 8K program has only six words and/or phrases included in data statements (there's plenty of memory left over in a 16K machine for adding more! Ed.) - most of them related to this computing hobby of ours - but has the option for entry of words from the keyboard when two players are competing with one another.

The short routine starting at line 3000 is not actually accessed by the program, but by entering G.3000. This routine converts your own words into numeric values which you can note down for entering in your own Data lines. ** N.B. You will cause the programme to crash if you add any Data lines before those in the sequence starting at line 600, since this Data is used first whenever the program runs.

Well you lucky Level 1 users, now you can have as much fun as Level 2 users had last month. Happy Hanging!!
entertainment.

** AMAZIN L2/4K ** by Peter Hartley
This program draws a series of mazes on the screen for you to find your way through. Just enter the listing shown and follow the instructions. We are expecting to get an even more sophisticated maze program in a month or two and when we do, we will publish a full explanation of the algorithm used and how the program works.

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542

** BIORYTHMS ** by Bernie Simpson

This program plots the 3 curves relating to the theory of Biorythms. The following is a brief description of the program:

Lines 100-240
Input & validation of todays date and birthdate. MT will contain maximum number of days in month (29 if February, and leap year) 182-184 compares birthdate with todays date.

Lines 250-260
Sets up parameters used in routine to calculate no. of days since 1/1/1900, where D1 becomes no. of days from 1/1/1900 till today, D2 becomes no. of days from 1/1/1900 till birthdate, and therefore D3 is no. of days since birth.

Lines 2000-2050
Routine to calculate days since 1/1/1900. 1 day is added for each leap year in between the two dates. The DATA statement in 130 is used to add the no. of days in the months gone.

Lines 280-300
P=no. of days into Physical cycle,
E=no. of days into Emotional cycle,
I=no. of days into Intellectual cycle

Lines 400-720
Sets up initial graph.
400-420 Headers and lines
480-600 Time scale on X-axis
620-720 Legend.

Lines 740-780
Sets up parameters used in curve drawing routine.
C indicates which cycle to plot,
N is no. of days gone into the cycle, and
F is the no. of days in each cycle (23,28,33).

Lines 2450-2800
SINE CURVE DRAWING ROUTINE.
This makes use of the SIN function in Level 2 BASIC. IN is the wavelength control, where reducing it from 0.28 increases the wavelength, and vice versa. DC is the no. of radians in a circle. The subroutine tests which cycle to plot, and 'sets' every calculated co-ordinate for P, every second one for E, and every fifth for I, in order to distinguish the curves on the screen. 2560 calculates the Y co-ordinate as the X co-ordiate increases. The constant at the end of this line is the amplitude control. Reducing this reduces the amplitude, and vice versa. The constant 19 in line 2580 is the wave displacement where increasing this causes the 'mean' of the curve to drop below the X-axis. The no. of days into the cycle is incremented before calculating the new co-ordinate.

Lines 800-840
Complete the description of the curves in relation to Active, Passive and Critical stages.

MICRO-80

P.O. BOX 213, GOODWOOO, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542

** FILES L2/16K ** by Eddy Paay
(Last month, we promised you a simple DATA-BASE MANAGEMENT SYSTEM. This is it, but Eddy has developed it to the point where it is a very useful program in its own right, rather than the simple demonstration of DATA statements we had originally envisaged. Eddy has also discovere a way to build a machine language sub-routine into a BASIC program that I've never seen before. I won't spoil Eddy's fun though. See if you can find it for yourselves. - Ed.)

THIS IS A PROGRAM DESIGNED FOR LEV 2 USERS WITH AT LEAST 16K OF MEMORY. IT ALLOWS THE USER TO ENTER DATA INTO A FILE, TO RECALL THIS DATA LATER AND LIST IT TO THE SCREEN, IN A FORMAT SET OUT BY THE USER. IT IS THEREFORE SUITABLE FOR SUCH FUNCTIONS AS A MAILING LIST CONTAINING NAMES, ADDRESSES AND PHONE NUMBERS OR A DIRECTORY TO CONTAIN, FOR INSTANCE, MAGAZINE ARTICLES, DATE, VOLUME NUMBER AND TYPE OF ARTICLE, ETC. THE PROGRAM IS THEN CAPABLE OF SEARCHING THE FILE FOR ANY DATA ASKED FOR BY THE USER AND DISPLAYING ALL FILES TO THE SCREEN WHICH CONTAIN THE DATA ASKED FOR.

THIS PROGRAM OVERCOMES MANY PROBLEMS ASSOCIATED WITH THIS KIND OF PROGRAM. IT AVOIDS LENGTHY, TIME CONSUMING DATA TRANSFERS FROM TAPE AS WOULD BE THE CASE IF THE INPUT#-1 COMMAND IS USED TO LOAD AND WRITE THE DATA TC TAPE SEPARATELY. WITH THIS SYSTEM ALL DATA STAYS WITH THE PROGRAM AT ALL TIMES AS DATA STATEMENTS, IT IS THEREFORE ONLY NECESSARY TO CLOAD OR CSAVE ONCE TO LOAD THE PROGRAM AND DATA FROM TAPE (OR DISK).

TYPING THE PROGRAM IN:
AS WAS STATED EARLIER, THIS IS A PART MACHINE LANGUAGE AND PART BASIC PROGRAM. NOW EXAMINE THE LISTING OF THE PROGRAM AND SEE IF YOU CAN FIND THE MACHINE LANGUAGE SUBROUTINE - - -
HAVE YOU GUESSED IT? - IT IS THE REMARK STATEMENT IN LINE 1. THIS REMARK STATEMENT CONTAINS NO REMARK BUT A M/L SUBROUTINE INSTEAD, THAT IS WHY IT APPEARS TO MAKE NO SENSE. THIS M/L SUBROUTINE ALLOWS THE USER TO TYPE IN DATA WITHOUT HAVING TO STOP THE PROGRAM AND TYPE IN COMPLETE DATA LINES MANUALLY.

THE M/L SUBROUTINE CREATES DATA LINES, NUMBERS THEM, AND CHANGES THE NECESSARY MEMORY POINTERS FOR BASIC SO THAT PROGRAM EXECUTION DOES NOT HAVE TO BE INTERRrUPTED. IT ALSO SAVES A LOT OF TIME AND EFFORT FOR THE USER. TO TYPE THIS PART OF THE PROGRAM IN DO NOT TYPE IN THE REM STATEMENT AS IT APPEARS IN THE LISTING IN THE MAGAZINE, BUT TYPE "1 REM" FOLLOWED WITH 255 SPACES THEN "ENTER" IT, THIS CLEARS THE REQUIRED AMOUNT OF MEMORY FOR THE M/L SUB. AFTER DOING THIS, USE A MONITOR (OR USE THIS MONTHS INSTALLMENT OF "BASIC MONITOR" (DISK BASIC USERS CAN USE DEBUG.) TO TYPE IN THE DATA FROM THE HEX. LISTING PROVIDED. STARTING AT THE FIRST ADDRESS (6A2AH FOR DISK BASIC OR 42EFH FOR LEV 2). (HINT! SOME MONITORS MAY DESTROY A BASIC PROGRAM WHEN RETURNING TO BASIC SO, TO BE SAFE, JUMP TO 6CCH. THIS IS THE BEST PLACE TO RETURN TO BASIC.)
AFTER RETURNING FROM YOUR MONITOR, JUST TYPE IN THE REST OF THE PROGRAM WHICH IS IN BASIC. THE ONLY OTHER THING TO BE MENTIONED IS THAT THE TEXT LINES SUCH AS LINES 2, 7 AND 50 MUST BE TYPED IN, EXACTLY AS LISTED. THIS REQUIRES THE USE OF THE CONTROL KEYS, THESE ARE THE ONES WITH ARROWS ON THEM.
AFTER THE FIRST DOUBLE QUOTE IN LINE 2 FOR INSTANCE, TYPE THE DOWN ARROW KEY AS MANY TIMES AS NECESSARY TO GET THE MESSAGE "*** INITIALIZING ***" RIGHT IN THE MIDDLE OF THE SCREEN.
ALSO NOTE THAT THE DATA STATEMENTS AT THE END OF THE PROGRAM DONT HAVE TO BE TYPED IN. THEY ARE THERE FOR DEMONSTRATION PURPOSES, AND SHOW HOW DATA IS STORED. THEY CAN BE TYPED IN HOWEVER SO THAT THE USER CAN PRACTICE USING THE PROGRAM. WHEN YOU HAVE FINISHED TYPING IN THE PROGRAM, CSAVE IT (DISK USRS CAN SAVE IT.

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542

HOW TO RUN

TO USE THIS PROGRAM, FIRST SET THE MEMORY SIZE TO : 32250 FOR THE 16K VERSION OR 49640 FOR 32K OR 65020 FOR 48K VERSIONS; THEN LOAD THE PROGRAM FROM TAPE USING THE CLOAD COMMAND (OR FROM DISK) THEN TYPE "RUN". THE SCREEN WILL THEN DISPLAY "INITIALIZING". WHEN INITIALIZATION IS COMPLETED, THE COMMAND LIST SHOULD BE DISPLAYED ON THE SCREEN. IF "CHECKSUM ERROR" IS DISPLAYED INSTEAD, THE M/L SUBROUTINE IS FAULTY, IN THAT CASE RELOAD IT FROM TAPE.
(WARNING !!! NEVER TYPE EDIT 1 UNDER BASIC, AS IT WILL DESTROY THE M/L SUBROUTINE IN LINE ONE).

THE COMMANDS :

(1) ENTER NEW DATA TO FILE.

THIS COMMAND ALLOWS THE USER TO ADD DATA TO THE FILE. ALL THAT IS REQUIRED IS FOR THE USER TO TYPE IN THE DATA IN THE CORRECT ORDER SEPARATED WITH COMMA'S. BECAUSE ALL DATA IS STORED AS STRINGS, THE DATA MUST BE STORED BETWEEN DOUBLE QUOTES. IT IS NOT NECESSARY HOWEVER FOR THE USER TO TYPE THESE QUOTES IN MANUALLY, BECAUSE AS SOON AS A COMMA IS TYPED IT IS IMMEDIATELY CONVERTED TO : "," AUTOMATICALLY.
IF DATA HAS BEEN TYPED IN INCORRECTLY, BASIC WILL RETURN WITH A SN ERROR IN THE DATA LINE CONCERNED WHEN IT READS THE DATA LATER, SO TAKE CARE.
WHEN A COMPLETE "SET" OR "GROUP" OF DATA HAS BEEN TYPED IN, TYPE "ENTER THEN, WHEN THE "READY" MESSAGE APPEARS TYPE ENTER AGAIN. YOU WILL THEN BE ABLE TO ENTER THE NEXT SET OF DATA, ELSE HIT THE DOWN ARROW KEY TO RETURN TO THE DIRECTORY.
ALSO YOU WILL NOTICE THAT THE AMOUNT OF FREE MEMORY LEFT IS DISPLAYED ON THE SCREEN SO THAT YOU WILL KNOW WHEN YOU COME TO THE END OF YOUR MEMORY.

DATA FORMAT.

THE FIRST SET OF DATA YOU TYPE IN WILL BE USED AS HEADINGS FOR ALL OTHER DATA, EXAMINE THE FIRST DATA STATEMENT IN THE LISTING OF THE PROGRAM TO SEE WHAT I MEAN.(OR BETTER STILL TYPE IN DATA AS IN LISTING AND TRY)
THE FORMAT OF THE FIRST SET OF DATA IS DIFFERENT FROM ALL OTHERS, IT CONTAINS A NUMBER AT THE BEGINNING OF THE DATA STATEMENT. THE NUMBER TELLS THE PROGRAM HOW MANY COLUMNS OF DATA THERE ARE IN EACH GROUP OF DATA.
FOLLOWING THIS NUMBER ARE THE SUB-HEADINGS, (I.E. IF THE NUMBER IS 3, THEN IT MUST BE FOLLOWED BY 3 SUB-HEADINGS IN THE FIRST LINE AND ALL OTHER DATA GROUPS ARE EXPECTED TO HAVE 3 LOTS OF DATA IN ITS GROUP OR SET.)
IF YOU LOOK AT THE LISTED DATA LINES YOU WILL NOTICE AN "@" FOLLOWED BY A NUMBER, THE "@" TELLS THE PROGRAM THAT THE DATA FOLLOWING CONTAINS THE LINE NUMBER. THIS IS NOT TYPED IN BY THE USER, IT IS DONE AUTOMATICALLY WHEN YOU ENTER DATA UNDER THE "ENTER DATA" COMMAND.

(2) SEARCH COMMAND.

THIS COMMAND WILL SEARCH THE FILE FOR THE DATA ASKED FOR BY THE USER. IT WILL THEN LIST THE FIRST OCCURENCE OF THE DATA CONCERNED. ANSWER THE QUESTION "CONT. OR STOP" WITH "C" TO SEARCH THE REST OF THE FILE OR WITH "S" TO STOP. NOTE ALSO THAT IT IS OFTEN NOT NECESSARY TO TYPE IN THE COMPLETE WORD OR SENTENCE YOU ARE SEARCHING FOR. FOR INSTANCE, IF YOUR FILE CONTAINS THE WORDS "BICYCLE AND MOTORCYCLE" THEN THE SEARCH COMMAND WILL LIST BOTH IF IT IS TOLD TO SEARCH FOR "CYCLE". THE SEARCH ROUTINE WILL TELL YOU IF DATA IS NOT FOUND OR IF ALL DATA HAS BEEN SEARCHED.

(3) EXIT FROM PROGRAM.

THIS COMMAND CAN BE USED TO STOP THE PROGRAM. UNDER LEV 2 THE BREAK KEY CAN ALSO BE USED TO STOP EXECUTION. UNDER DISK BASIC HOWEVER, THE EXIT COMMAND MUST ALWAYS BE USED TO STOP THE PROGRAM AND NOT THE BREAK KEY OR DISK I/O WILL

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542

NOT FUNCTION PROPERLY.

(4) EDIT AND LIST LINE NUMBERS.
THIS COMMAND TELLS YOU HOW TO EDIT THE DATA LINES BUT DOES NOT ACTUALLY PERFORM THE EDIT FUNCTION ITSELF. TO EDIT A FILE, STOP THE PROGRAM AND USE THE EDIT FUNCTION PROVIDED BY BASIC.(REMEMBER, DON'T EDIT LINE 1).
THIS COMMAND CAN HOWEVER SEARCH FOR A PARTICULAR STRING OF DATA IN ITS FILES AND LIST ALL LINE NUMBERS OF DATA LINES THAT CONTAIN THE STRING IN QUESTION SO THAT THE USER WILL KNOW WHICH LINES TO EDIT.

(5) LIST ALL.
THIS COMMAND WILL LIST ALL DATA TO THE SCREEN ONE FILE AT A TIME. TYPE "Y" TO CONTINUE LISTING AND "N" TO STOP, WHEN ASKED.

16K, 32K AND 48K VERSION DIFFERENCES.
THE FOLLOWING LINES MUST BE ALTERED FOR DIFFERENT VERSIONS : 3, 7, 150, 290, 300, 302
FOR 16K MACHINES, TYPE IN THE PROGRAM AS LISTED, FOR 32 AND 48K VERSIONS HOWEVER, THE FOLLOWING CHANGES HAVE TO BE MADE:

THE 32K VALUES ARE SHOWN FIRST FOLLOWED BY THE 48K VALUES BETWEEN THESE BRACKETS : "< >", "REST SAME" MEANS NO CHANGE FROM MAIN LISTING OF PROGRAM FOR THE REST OF THE LINE.
3 Y=-16896:(-512> (THE REST SAME)
7 (SAME UP TO "END" STATEMENT):ELSEPOKE16526,126:POKE16527,190 <254> (REST SAME)
150 A=2:BUFFER=-16641 <-257> (REST SAME)
290 POKE16405,0:POKE16526,0:POKE16527,190 <254> (REST SAME)
300 POKE16405,1:NU=PEEK(-16669 <-285>)+PEEK(-16668 <-284>) (REST SAME)
302 X=PEEK(-16672 <-288>)+PEEK(-16671 <-287>) (REST SAME)
APART FROM THE CHANGES SHOWN THE CHECKSUM WILL BE DIFFERENT, THEREFORE THE VALUE 27340 IN LINE 7 , MUST BE CHANGED SO IT WON'T RETURN A CHECKSUM ERROR. TO FIND THE CORRECT CHECKSUM VALUE FOR 32K AND 48K VERSIONS , RUN THE PROGRAM AND WHEN IT STOPS AND DISPLAYS "CHECKSUM ERROR" TYPE "?B" AND REPLACE 27340 IN LINE 7 WITH WHATEVER WAS PRINTED ON THE SCREEN.(FOR 32K VERSION CHECKSUM SHOULD BE 28748)

DISK BASIC :
FOR DISK BASIC SEVERAL CHANGES SHOULD BE MADE , FIRST ALL POKE STATEMENTS CONNECTED WITH THE X=USR(0) COMMAND MUST BE DELETED.
FOR A 16K DISK BASIC VERSION FOR INSTANCE THE LAST PART OF LINE 7 SHOULD READ : ELSEDEFUSR0=&H7E7E:X=USR0(0), THE SAME APPLIES TO ALL LINES CALLING THE M/L SUBROUTINE.
NEXT CHANGE THE FOLLOWING LINES :
350 M=0: FORJ%=1TOLE :READB$(J%):M=M+INSTR(B$(J%),B$) :NEXTJ% :READH$:READH
540 POKE&H4034,&HBB:POKE&H4035,&H44:CLS:END
THEN DELETE LINES 800 - 810 INCLUSIVE.
CHANGING LINE 350 AS SHOWN ABOVE WILL MAKE THE SEARCH FUNCTION MANY TIMES FASTER UNDER DISK BASIC.

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542

** SET2 **

SET2 IS A MACHINE LANGUAGE PROGRAM DESIGNED TO INCREASE THE SPEED OF GRAPHICS AND TO MAKE PROGRAMMING EASIER. IT SIMPLY DRAWS OR RESETS A LINE BETWEEN ANY TWO POINTS ON THE SCREEN, INSTANTLY. IT ALSO MAKES PROGRAM EXECUTION FASTER AS IT DOES AWAY WITH MOST FOR/NEXT LOOPS, WHICH ARE NEEDED IF A LINE IS TO BE DRAWN USING THE ORDINARY SET(X,Y) COMMAND.

WHEN SET2 IS LOADED AND ENABLED, THERE WILL BE TWO NEW COMMANDS AVAILABLE FOR USE THROUGH YOUR BASIC PROGRAMS. THE SYNTAX FOR THESE NEW COMMANDS ARE : SET<A,B,C,D> AND RESET<A,B,C,D> (NOTE THE DIFFERENT BRACKETS WHICH ARE USED). WHERE A,B,C OR D CAN BE ANY INTEGER VARIABLE ALLOWED UNDER BASIC. IT IS USUALLY EASIER HOWEVER, TO USE A " DEFINT " STATEMENT AT THE BEGINNING OF YOUR PROGRAM TO DEFINE ALL VARIABLES USED IN THE NEW SET/RESET COMMAND.

SOME RULES WILL HAVE TO BE CONSIDERED, THESE MUST BE STRICTLY ADHERED TO OR BASIC WILL RETURN A SYNTAX ERROR. THEY ARE :

1. NO BLANKS MUST BE PUT BEFORE THE SET/RESET COMMAND OR ANYWHERE IN THE PROGRAM LINE CONTAINING THE NEW SET/RESET COMMAND.
2. THE COMMAND MUST BE IN A PROGRAM LINE ALL BY ITSELF, AND NOT IN A MULTISTATEMENT LINE. (EVERYTHING THAT IS TYPED AFTER IT IS IGNORED)
3. IT CAN ONLY BE USED AS A STATEMENT IN A PROGRAM AND NOT AS A DIRECT STATEMENT.
4. IF THE COMMAND READS : SET<X1,Y1,X2,Y2> THEN IT WILL DRAW A LINE FROM X1,Y1 TO X2,Y2 ON THE SCREEN. (NOTE THAT X1,Y1 DOES NOT HAVE TO BE THE SMALLER SET OF VARIABLES, ALL THAT IS REQUIRED IS FOR THE VARIABLES TO BE "X" FIRST FOLLOWED BY "Y".) 5. THE USER MUST ENSURE THAT THE VARIABLES DO NOT ATTAIN VALUES GREATER THEN 127 FOR X AND 47 FOR Y OR WRAP-AROUND WILL OCCUR.

TYPING THE PROGRAM IN

THOSE READERS WHO HAVE THE EDITOR/ASSEMBLER PROGRAM CAN TYPE IT IN USING THE MNEMONICS (THIS IS EVERYTHING TO THE RIGHT OF THE DECIMAL LINE NUMBERS), THOSE OF YOU WHO DO NOT HAVE THE EDITOR ASSEMBLER WILL HAVE TO USE A MONITOR AND TYPE IN THE HEXADECIMAL DATA TO THE LEFT OF THE DECIMAL LINE NUMBERS IN THE LISTING.

IN OTHER WORDS IF THE LISTING SAYS : 7B00 FD2135 THEN MEMORY LOCATION 7B00 CONTAINS FD, 7B01 CONTAINS 21 AND 7B02 SHOULD CONTAIN 35 (ALL NUMBERS IN HEXADECIMAL), THE REASON THE ONE MEMORY LOCATION HAS THREE BYTES LISTED BEHIND IT IS THAT THE COMMAND HAPPENS TO BE THREE BYTES LONG, IN THIS CASE.

AFTER TYPING IT IN COPY IT TO TAPE IMMEDIATELY (USE A MONITOR OR "THE BASIC MONITOR" IN THIS MONTHS ISSUE) USING THE FOLLOWING ADDRESSES: START=7DDE , END=7FFF , ENTRY=7DDE

LOADING THE PROGRAM

MAKE SURE THE MEMORY SIZE IS SET TO 32222. THEN LOAD IT FROM TAPE USING THE SYSTEM COMMAND AND ANSWER THE "*?" WITH SET2 (UNLESS YOU CALLED IT BY SOME OTHER NAME), THEN WHEN THE QUESTION MARK RETURNS TYPE IN A SLASH "/" FOLLOWED BY "ENTER". " READY " SHOULD THEN BE DISPLAYED ON THE SCREEN AND THE NEW COMMANDS WILL BE ENABLED AND READY TO BE USED.

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542

SAMPLE PROGRAM USING THE NEW COMMANDS.
TRY THE FOLLOWING PROGRAM:

```
10 DEFINTX,Y:CLS
20 X1=RND(127):Y1=RND(47):X2=RND(127:Y2=RND(47)
30 SET<X1,Y1,X2,Y2>
40 RESET<X1,Y1,X2,Y2>
50 GOTO20
```

THE PROGRAM ABOVE WILL RANDOMLY DRAW LINES ON THE SCREEN AND WILL DEMONSTRATE THE SPEED OF THE NEW COMMANDS IN LINES 30 AND 40.

** THE BASIC MONITOR - (BMON) L2/16K **
THIS IS A MACHINE LANGUAGE PROGRAM SUITABLE FOR A SYSTEM WITH 16K OR MORE MEMORY. THE PROGRAM PROVIDES THE USER WITH MANY POWERFUL UTILITIES COMMONLY NEEDED WITH LEVEL II BASIC. AFTER IT HAS BEEN INITIALIZED THE PROGRAM CAN SIMPLY BE ACCESSED BY TYPING A SHIFTED DOWN ARROW KEY AT ANY TIME. AFTER TYPING THE SHIFTED DOWN ARROW KEY THE COMMAND LIST WILL BE DISPLAYED ON THE SCREEN AND THE PROGRAM WILL BE READY TO ACCEPT A COMMAND. THE PROGRAM CONTAINS THE FOLLOWING FUNCTIONS :

EDIT MEMORY.
THIS UTILITY ALLOWS THE USER TO TYPE HEXADECIMAL DATA DIRECTLY INTO MEMORY AND ALSO DISPLAYS THE CONTENTS OF THE CURRENT MEMORY ADDRESS.

COPY MEMORY TO TAPE.
THIS COMMAND ALLOWS THE USER TO DUMP ANY PART OF MEMORY TO TAPE. THESE TAPES CAN THEN BE LOADED BACK INTO MEMORY AGAIN USING THE SYSTEM COMMAND PROVIDED BY BASIC.

GOTO HEX. ADDRESS.
THIS COMMAND ALLOWS THE USER TO JUMP TO ANY PART OF MEMORY DIRECTLY USING A HEXADECIMAL ADDRESS. (THIS AVOIDS HAVING TO CHANGE THE ADDRESS TO DECIMAL)

RESTORE BASIC PROGRAM.
THIS COMMAND WILL RESTORE A BASIC PROGRAM IF "NEW" HAS BEEN TYPED ACCIDENTALLY.

LIST VARIABLES.
THIS COMMAND LISTS ALL VARIABLES USED IN A BASIC PROGRAM. THIS IS A HANDY FUNCTION IF YOU ARE ADDING A PART TO A PROGRAM AND ARE NOT SURE IF A CERTAIN VARIABLE HAS ALREADY BEEN USED.

DECIMAL TO HEX. AND HEX. TO DECIMAL CONVERSIONS.
THE PROGRAM WILL CONVERT HEX. TO DECIMAL AND DECIMAL TO HEX WHEN ASKED AND PRINTS OUT THE ANSWER ON THE SCREEN.

RENUMBER.
THIS UTILITY WILL RENUMBER THE LINE NUMBERS IN A BASIC PROGRAM TO USER SUPPLIED PARAMETERS.

LOAD.
THIS PART LOADS BASIC PROGRAMS, DISPLAYS THEM TO THE SCREEN WHILE LOADING AND TELLS THE USER WHERE THE PROGRAM IS LOCATED AFTER LOADING IS COMPLETED.

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542

MERGE.
THIS LOADS A BASIC PROGRAM AND MERGES IT TO A BASIC PROGRAM ALREADY IN MEMORY THEN THE NEW PART IS AUTOMATICLY RENUMBERED SO THAT IT WILL FIT BELOW THE PROGRAM THAT WAS ALREADY IN MEMORY.

PROTECT PROGRAM.
THIS COMMAND WILL PROTECT A BASIC PROGRAM AND MAKE IT INVISIBLE TO BASIC.

CANCEL PROTECTION.
THIS IS THE REVERSE OF THE PROTECT FUNCTION AND WILL RECALL THE BASIC PROGRAM SO THAT IT CAN BE USED AGAIN.

BECAUSE OF THE SIZE OF THE PROGRAM IT WILL BE SPLIT UP INTO THREE INSTALLMENTS OVER THE NEXT THREE ISSUES. THE PARTS PRESENTED IN THIS ISSUE WILL BE ABLE TO RUN BY THEMSELVES. ALL THAT WILL BE REQUIRED EACH MONTH IS TO LOAD THE PREVIOUS BLOCKS, THEN TYPE IN THE NEW PARTS FROM THE MAGAZINE AND RELOAD THE PROGRAM ON TAPE, ALL THIS CAN BE DONE USING THE PROGRAM ITSELF.

THIS MONTH WILL CONTAIN THE EDIT MEMORY COMMAND, COPY MEM. TO TAPE, GOTO HEX. ADDRESS AND THE RETURN TO BASIC FUNCTIONS.

FOR THE FIRST INSTALLMENT YOU WILL HAVE TO USE A MONITOR TO ENTER THE DATA FROM THE HEX DUMP TO MEMORY, (OR USE THE MONITOR IN BASIC IN THIS MONTH'S ISSUE) AFTER THAT THE EDIT COMMAND CAN BE USED TO ENTER THE REST OF THE INSTALLMENTS INTO MEMORY. AFTER THE DATA HAS BEEN TYPED IN THE COPY COMMAND WILL ENABLE YOU TO LOAD THE PROGRAM TO TAPE.

DETAILED DESCRIPTION OF COMMANDS.
AS WAS SAID EARLIER, TO ACCESS THIS PROGRAM JUST TYPE <SHIFT> DOWN ARROW AT THE SAME TIME THEN CALL UP THE REQUIRED COMMAND.

EDIT MEMORY :
THIS UTILITY WILL ASK " HEX. ADDRESS ? " ANSWER THIS WITH THE ADDRESS AT WHICH YOU WANT TO ENTER YOUR DATA. IT WILL THEN DISPAY THE FIRST HEX. ADDRESS AND THE CONTENTS OF THIS ADDRESS.
IT LOOKS LIKE THIS:

8000 <DE>

THIS MEANS THAT LOCATION 8000 HEX. CONTAINS "DE" HEX.
YOU CAN THEN ENTER NEW DATA TO THIS LOCATION OR TYPE THE DOWN ARROW OR UP ARROW KEY TO INCREMENT OR DECREMENT THE ADDRESS COUNTER RESPECTIVELY .
OTHER COMMAND KEYS ARE "X" TO CHANGE THE EDIT ADDRESS AND THE BREAK KEY TO RETURN TO THE COMMAND LIST (THE BREAK KEY CAN BE USED TO ESCAPE FROM ANY COMMAND OR FUNCTION).

COPY MEM. TO TAPE.
THIS UTILITY WILL DISPLAY " START END ENTRY NAME " ON THE DISPLAY. JUST TYPE THE CORRECT VALUES UNDERNEATH EACH. AS SOON AS YOU TYPE THE SIXTH LETTER IN THE NAME, THE RECORDER WIL START TO RUN, AUTOMATICALLY. IF THE NAME HAS LESS THAN 6 LETTERS, TERMINATE IT WITH THE "ENTER" KEY. A GRAPHICS BLOCK WILL FLASH IN THE RIGHT LOWER CORNER AFTER EACH BLOCK IS DUMPED TO TAPE

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542

GOTO HEX. ADDRESS.
THIS COMMAND WILL ASK "HEX. ADDRESS ?" TYPE IN A 4 DIGIT HEX. ADDRESS AS ANSWER TO THIS AND CONTROL WILL BE TRANSFERED TO THIS ADDRESS IMMEDIATELY.

AFTER TYPING THE PROGRAM IN, JUMP TO 31641 (7B99H),(THE SYSTEM COMMAND CAN BE USED FOR THIS BY TYPING " /31641 " AFTER SYSTEM HAS BEEN CALLED UP.) THEN WHEN THE COPYRIGHT MESSAGE APPEARS, TYPE <SHIFT> AND DOWNARROW, THEN WHEN THE COMMAND LIST COMES UP ON THE SCREEN HIT THE "C" KEY THEN USE THE FOLLOWING ADDRESSES TO COPY THE PROGRAM TO TAPE : START = 7226H , END = 7EFFH , ENTRY = 7B99H .NAME = BMON (OR ANY OTHER NAME UP TO SIX CHARACTERS).

APART FROM THE UTILITIES ABOVE "BMON" ALSO HAS A KEY-DEBOUNCE ROUTINE BUILT IN. IF KEYBOUNCE IS STILL A PROBLEM EVEN WITH THE PROGRAM LOADED, INCREASE THE VALUE STORED IN 7D87H. SCANNING WILL THEN SLOW DOWN FURTHER AND LESS BOUNCE SHOULD OCCUR. (NOTE: THE KEY DEBOUNCE SUBROUTINE ONLY WORKS UNDER BASIC, NOT M/C LANGUAGE).

*** MONITOR IN BASIC REVISITED *** by Peter Hartley

Being lumbered once in a month is bad enough, but He did it to me twice! Clutching a cassette, allegedly containing a copy of last month's "Monitor in Basic" program (all lines from 6000 to 8200 were missing!), and an excellent letter from Ron. Sully of Loganholme, Qld, (who's turning into something of a regular correspondent already! - good on you, Ron!), this time it was a case of "see if you can sort this out - it doesn't work!"

Now, sometimes I have to admit a certain mental blockage towards someone else's way of approaching a computing problem, and this, unfortunately, was one of these occassions. Finally I gave up, got an understanding from Him that we would publish Ron's letter, and set about creating a new Monitor. Frankly, I hadn't even thought about this particular problem before, being blessed with several excellent machine language varieties that can be forced to do my will, whenever needed.

I sincerely trust that everyone will be happy with the result, which will give you a straight hex listing on the screen, like those we publish in MICRO-80, the ability to change the value stored in any R.A.M. location, and facilities to store to, and retrieve from, cassette. It's not very fast, but it's faster than last month's version, and will dump a 4K programme in twenty-three minutes (exactly 22 times as long as TBUG). It does, however, let you see exactly what's going out to tape, or what's coming in from tape, while it runs.

Unfortunately, it just will not squeeze into a 4K level II machine. Now then! Stop your grumbling at the back! The good news follows in a minute!

There is one sequence that is untested, and that's the error trap at the end. I've had the hardware cassette read modification fitted to my '80, and try as I did, it wouldn't miss a byte all evening, so if that particular sequence doesn't work you can all throw it away, together with the ONERRORGOTO bit at the beginning.

NOW, the Good News for all you 4K folk, out there.

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542

*** MINI MACHINE LANGUAGE LOADER IN BASIC *** by Uno Hoo

This is in two parts, the writer and the reader, and it leaves you about 1.5K in a 4k machine for all those high-speed goodies that MICRO-80 has coming up.

The writer is just a simple way of creating a data tape for the reader. To use the writer, there's no need to protect any memory, just enter it in and get a couple of HIGH QUALITY BLANK TAPES. Once you've entered the start, end, and entry points, and the name, you simply copy the MICRO-80 hex listings, and, from time-to-time, your '80 will lock up the keyboard while it dumps a whole 124 bytes out to tape. If you make a goof, while entering the data, just carry on with the corect data. Providing that you haven't hit >> ENTER << yet, it will not matter, as the program only looks at the last two alphanumerics in each entry. The only way that I could get it to glitch up was by entering spaces. You may like to put a trap into it, to check that the last two characters are within the range of 0-9-F, but I assumed that the majority of our readers were pretty bright sorts who wouldn't make those sorts of errors anyway!

The second part is the reader, largely lifted from the full scale program. Before entering this, set your memory protect level to 18800, to reserve 1679 bytes for the m.l. goodies.

Once it is entered, just set the cassette to play, and go and have a cuppa. Once the m.l. is entered, you can access it by BREAKing the basic routine, typing SYSTEM, >> ENTER <<, and then responding to the *? with / and the entry address IN DECIMAL.

Finally, a word about that cassette read modification that I metioned earlier. If you have troubles reading tapes into your '80, it really is worth your while getting this. If your '80 is still under warrenty, or if you are not very handy, I understand that Tandy will now supply and fix the item at no charge. You'll just have to wait while they ship the unit to Sydney and so forth. Otherwise, your local Tandy Store will obtain one of these for you, and charge only about $3.60 for it. The unit is just a tiny piece of vero-board with a couple of I.C.'s and about six wires, and should come with a very good instruction sheet, that not only explains how to install it, but also how it works. AND IT REALLY DOES WORK WELL. I can set the cassette volume at anything from 3 1/2 to 8 1/2 and it still reads it all! It'll save a great deal of frustration over the years.

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542

***** HARDWARE SECTION *****

how to save a lot of dollars, and still have a lot of fun,

BUILDING YOUR OWN T.R.S. LIGHT PEN FOR UNDER $3.88

by Peter Hartley

There I was, minding my own business, teaching our T.R.S. 80 to gamble on the horses without losing too much of my wife's hard earned cash, when who should telephone but a certain nameless magazine publisher. "You're not doing much these days, are you?" he warbled sweetly. Without giving me a moment to utter a single cautionary remark of over-commitment, he muttered darkly about something called an "Associate Editorship", "not much money for the first seven years", and hung up. Assuming that he was simply suffering from TRSDOSitis, or an over abundance of subscription applications, I resumed my earnest attempts at creating the world's first millionaire T.R.S. 80.

Precisely twenty-four hours later He (capitals for His Proprietorship) was on the phone again, very determined. "I seem to recall that you were building a light pen. I hope that it's working because I've promised the readers an article for this month, and we can't get our prototype to work." Before I could explain that my version suffered from every electronic complaint possible, and that the "project" was lying under fifteen feet of form-guides and copies of the Sunday result sheets, he announced that he "wanted the article by Tuesday! It's got to be at the printers then." He was gone and I was lumbered!

First of all, then, some of you may want to know what a light pen is, what it does, and how. It's really a device for lazy people with tired fingers, enabling them to communicate with a well-trained computer, without using any sort of key-board.

A light pen is not a torch, nor does it blast a laser beam into your video terminal, so powerful that is goes down all those funny little wires and comes up in your overheating RAM. It does, in fact, the exact opposite, picking up modulated or pulsed light from the video display, and, if your programming is all right, telling the beast where the pen is located.

Commercial light pens can cost anywhere from around $20 to $200. As with all things, you'll never get more than you pay for but you will very often get a great deal less. The unit that I'm going to describe here is the rock-bottom, cheap, horrid, unreliable little nasty for which you'd normally pay anything up to $50 or $60. If it costs you even one cent over $3.88 you've been ripped off by your local electronics component supply house. Unfortunately, you will not be able to buy one of the components at your local Tandy store.-

Because this version was designed firstly to be cheap, and only secondly to work, it may be necessary to make slight ajustments to the unit according to the eccentricities of your particular system and the ambient light levels in your den, lounge, or, if your wife is like mine, the little bit of passage outside the toilet and bathroom.

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542

First of all you will need to collect the necessary components, which are as follows;

1.) a plastic (black if possible) pen housing, with a reasonably large (2mm-3mm approx) opening where the writing business used to come out. This needs to be of a type that is made in two pieces, and is all screwed up in the middle - like some of those programs we've seen from time to time :$0.00 - should be one lying around.

2.) a Tandy 9 volt battery (Catalogue number 23-464):$0.59

3.) a flat connector for the battery (Tandy will sell you a pack of five (Catalogue number 270-325) for $0.99):$0.20

4.) a 1/8" miniature jack plug: Pack of two (274-286) for $1.39:$0.70

5) approx three feet of fine microphone cable - with two centre conductors and a single screen. N.B. First of all, it will not matter if this cable has more than two centre conductors, but you'll quickly come to hate the entire device if the cable is too thick and stiff. Tandy carry only one type, which, in my opinion, is far too heavy, stiff, and thick to fit into the pen, and they'll sell you a twenty foot pack which is enough to make up six or seven pens. Fortunately my local Tandy manager is fairly tame, and a computer genius into the bargain, and he had a piece under the counter that was just what I needed. However, we can't all live in the eastern suburbs of Adelaide, so you may have to hunt around for something suitable:$1.25 maximum.

6.) one resistor, 1/4 watt. The value will vary from one unit to another. In various tests I have found values from 470 ohms to 10 meg necessary, and the unit that I am presently using has dispensed with this component all-together!:$0.25

7.) one silicon phototransistor, N.P.N., similar to Fairchid type FPT-100. (Tandy catalogue number 276-130). N.B. You may find two types of this device in your local Tandy dealership, both with the same catalogue number. Fortunately they come in a clear bubble-pack so you can see what you are getting. The visible difference is that the lens end of the device (the end that doesn't have any wires on it) is flat on one of the versions and rounded on the other. Having tried both types I consider that for this particular application you must use the rounded lens type:$0.85

Now to fit it all together.

Solder and soldering iron ready? If you're a handyman type, then you'll have a drill lying around somewhere that's just right for creating an opening in the end of the pen to admit the cable. If, on the other hand, you're more of a "Norm", like me, you can simply melt a hole with the soldering iron, but make sure that there's plenty of ventilation, because if the body of the pen is made of P.V.C. then the fumes could be positively harmful.

The phototransistor needs to fit comfortably into the front half of the pen body. If it doesn't fit you'd better hunt around for a larger pen - don't try filing away at the phototransistor because you'll probably kill it, and Murphy's Law dictates that the very day this issue comes out will be the day that Tandy Australia starts to enjoy a national phototransistor shortage. The one you kill will be the last one they have, and they don't re-order until Thursday so the earliest they'll get any more is Monday week!

You'll find three wires in the base of the phototransistor, arranged in a triangle. Locate the "base" wire (the middle one), or follow the diagram on the back of the box (You didn't throw it away, did you?). In a normal transistor, the current into this pin would control the flow of current between the other two pins. In a phototransistor, the supply to the "base" is generated by light falling on it, and you can cut off this wire, carefully, very close to the body of the transistor.

Now feed one end of the cable through the still smouldering hole in the pen, and strip off the insulation, taking care not to rip the braid screen, which needs to be unpicked and twisted. You now have two thin insulated wires, from which you should remove about 4mm of insulation, and the twisted end of the screen. Tin the ends of the three wires, and solder one of the insulated wires to each of the two remaining legs on the phototransistor, about 7mm from the body. When you do this, hold the wire from the phototransistor with a pair of fine nosed pliers, close to the transistor body. This will shunt off any excess heat, minimising damage to both the transistor, and to your valuable programmers fingertips.

With the pins of the phototransistor facing you, and with the remains of the "base" pin at the bottom, note the colour of the wire that you have connected to the left hand pin; this is the signal lead.

After feeding the other end of the cable, together with the leads from the battery connector, through the body of the jack plug, carefully strip off the insulation at that end of the cable in a similar manner. The wires from the battery conector will already be stripped and tinned, but a little dab of solder here will do no harm.

With luck you'll have one of those dinky little table-top vices around the place, which are most useful when soldering miniature jack plugs. If you can't readily lay your hands on one of these, you'll find the next best holder is one of the sockets for which the plug is intended. If your daughter won't lend you her tranny for half-an-hour, you'll probably have to use the microphone socket in your own computer's tape deck.

It always helps to improve access for the soldering iron if you spread the solder lugs on the jack plug apart, before tinning them. Carefully solder the signal lead to the smaller of the two lugs on the jack plug, and the screen, together with the RED lead from the battery, to the other. The remaining two leads should be soldered together, taking care to make the joint as small and as tidy as possible. This last joint will have to be insulated in order to prevent it from coming into contact with either of the lugs inside the plug assembly. You could use a little dab of rubber glue, sealastic, or even epoxy putty. Probably the best idea is to slip a sleeve of discarded plastic insulation over the joint.

At this stage, the only wire not connected is the pen end of the screen braid. The sensitivity of the pen is controlled by varying the value of a resistor connecting the screen to one side of the phototransistor. you can make the final assembly of the plug at this stage, to enable testing and ajustment to be carried out. The two teeth on the side of the larger of the jack plug lugs should be bent down with your pliers, so that they both bite into the outer insulation of the main cable. This is to ensure that a sharp tug on the cable will not rip out all your careful soldering.

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542

You will need to enter the following test program into your T.R.S. 80.

```
10 CLS
20 OUT 255,4
30 FOR T = 0 TO 10: NEXT
40 PRINT @ 512, INP(255),
50 PRINT @ 0, CHR$(191);
60 OUT 255,4
70 FOR T = 0 TO 10: NEXT
80 PRINT @ 576, INP(255),
90 PRINT @ 0, CHR$(32);
100 GOTO 20
```

This program will cause a block of light in the top left-hand corner of the screen to flash on and off, while the value of the cassette port is checked, and the results displayed on the screen.

At this stage the phototransistor should not be installed in the pen body. Start the program running, and plug the light pen into the microphone port of the cassette recorder, which should not contain a tape. Disconnect the auxillary jack from the computer, and leave the earpiece jack connected.

Inside the cassette bay of the recorder, at the back on the left-hand side, you will find a small metal arm. This normally senses the presence or lack of the erase protection tab on the back of the cassette. Gently push this arm in, simultaneously setting the recorder into RECORD mode, and then connect the battery to the battery clip.

Hold the phototransistor, by the body - not the wires, with its lens touching the video screen over the flashing point in the top left-hand corner of the screen.

If all is well you should see the numbers 127 and 255 on the sceen. If both are showing 127, try turning the brightness of the display up until the lower number changes to 255. On the other hand, if both numbers are showing 255, there is too much light, and either the pen must be desinsitised, or the video display turned down slightly.

To reduce the sensitivity of the pen, connect progressively lower value resistors across the screen and one pin of the Phototransistor. I'm certain that for $3.88 you're prepared to spend a little time experimenting?

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542


```
1 '
5 ' SIMPLE EXAMPLE OF LIGHT PEN APPLICATION PROGRAMME.

10 DEFINTS
20 CLS:PRINT:PRINT:PRINT"M E N U";:PRINTTAB(35)"SELECT":S3=0
30 PRINT
40 S1=4 ' THIS LINE SETS THE MAXIMUM NUMBER OF OPTIONS
50 FORS2=1TOS1:READD$(S2):NEXT
60 FORS2=1TOS1:PRINTTAB(2)D$(S2):PRINT:NEXT
70 FORS2=1TOS1:GOTO120
80 PRINT@165+128*S2,STRING$(2,191):GOTO130
90 PRINT@165+128*S2,"  ":NEXT
100 ONS3GOTO150  ,160  ,170  ,180
110 GOTO70
120 OUT255,4:S4=INP(255):GOTO80
130 IF INP(255)<=S4 GOTO90   ELSE IF INP(255)>S4 GOTO140  ELSE GOTO90
140 S3=S2:GOTO100
150 PRINT@896,"YOU CHOSE LINE #1.":GOTO190
160 PRINT@896,"YOU CHOSE LINE #2.":GOTO190
170 PRINT@896,"YOU CHOSE LINE #3.":GOTO190
180 PRINT@896,"YOU CHOSE LINE #4."
190 FORTI=0TO1000:NEXT:RESTORE:GOTO20
200 DATA THIS IS LINE ONE, THIS IS LINE TWO
210 DATA THIS IS LINE THREE, THIS IS LINE FOUR
```

-00000-

***** LETTERS TO THE EDITOR *****

From: Mr. V.C. POTINES, Sunbury Vic.
I am looking forward to your Assembly Language Programming Course. Is it possible to obtain the source listing of LOADER and future SYSTEM programs?

(Yes, we can supply source listings separately, to order. In future, we will publish source listings of short programs in the magazine. Unfortunately, the source listings of some of the longer programs are as big as the magazine itself, so we will have to make a nominal charge for these. We will indicate the charge in the relevant software section. -Ed.)

From: JOHN C. HARDWICKE, Collaroy Plateau, NSW.
The Tandy Technical literature I have been able to get my hands on seems pretty sparse, I believe you could include articles on references to suitable reading material.

(Thanks for the suggestion John, we certainly will. In fact, we are currently in contact with a number of American publishers of TRS-80 literature with a view to importing the better publications for resale. We hope to be able to offer the first few Titles next month. -Ed.)

From: L.A. PEARSON, Somerton Park, S.A.
As an amateur user of Level 2 I find many things frustrating. The lack of wrap around on graphics, the fact that I have a Level 1 to Level 2 conversion tape and can't find out how to use it and errors in published programs. The "Ready, Aim, Fire" program in the Level 2 Manual does not work! I got a copy on a purchased games tape which would not run correctly either! Maybe you could publish a better version.

As other machines come into general use you should extend your magazine to include them. Even in US magazines there is as yet, little Apple II software advertised.
I have been reading Tandy Assembly Language book. I was not clear if T BUG was included in the Editor/Assembler but evidently it is not. I cannot see the point in using assembly language for mathematical functions in Level 2 when they are already in ROM. Search or graphics routines are probably better but not if Assembler takes 5K. I have not seen T BUG advertised as a tape program. I hope you get one on the market.
As regards hardware, I hope you will publish something on how to connect to the output connector without interface. Also, it seems that Centronics printer at about $600. is the most reasonable, moderate performance machine on the market at present. Comments on printers in your magazine would be appreciated.
As not all programs in any issue are of interest, will individual ones be available on tape (at same cost?) I am assembling a universal subroutines series program which may be of interest. Do you want typewritten program comments? I presume you publish LIST with your printer, from tape. The December issue - Electric typewriter, is more readable than the Jan. "printer" version.

(Taking your comments and questions in order.
No, Level 2 is not superior to Level 1 in all respects although, you will soon find that the lack of graphics wrap around is not a serious penalty to pay for all the other good features of Level 2.
We are surprised that you can't find out how to use the Level 1 to Level 2 conversion tape. It should have come with instructions. Briefly, you load it into your Level 2 machine under the SYSTEM command. You then type "/19190" for a 4K machine or "/31478" for a 16K machine. The program starts off by prompting you to read in the Level 1 tape that you want to convert with the message "LOAD TAPE AND PRESS ENTER". Before pressing ENTER, make sure you adjust the volume control on your recorder to the Level 1 range (7-8). Once the Level 1 tape is read in, the computer displays "PRESS ENTER TO BEGIN". Do that and wait for up to a minute while the Conversion program does its thing. On completion, the computer displays "CONVERSION COMPLETE - PRESS ENTER TO CONTINUE". On pressing ENTER you return to BASIC with the converted program in memory. It is a good move to CSAVE it immediately, before running it. Finally, you may need to make some changes to run the program in Level 2. If it uses an array, you may need to add a DIM statement. If it relies on graphics wrap around you will have to add that feature with some extra program lines. Incidentally, if you have a 4K machine, it is possible that some of your programs will be too long to convert and you will have to do the conversion on a 16K machine.
Afraid we can't help you with the Ready, Aim, Fire program since we've never used it. Perhaps some other reader has sorted it out and will write in. If not, we will have a look at it ourselves.
As far as programs for other machines are concerned, we have no plans to include them at present. Quite frankly, we've got our hands full with the TRS-80 for the moment. Maybe we will consider a separate magazine for Apple and Pet Commodore users at a later date, the problem being that there are not as many of these machines in use as there are TRS-80's.
You're right, TBUG is not included in the Editor/Assembler but it is available as a tape program from your Tandy Dealer. You are also right about using the mathematical functions in ROM whenever possible. However, some people like to write their own for the satisfaction it gives them. Incidentally the Editor/Assembler does not take up space when you are running your program.

You use it to make a system tape, clear the machine and then load in the system tape, by itself.
We will include your suggestion about connecting hardware directly to the output connector in our requests. You can count on an article about this in the next few months. Several readers have asked us to comment on printers and we intend to prepare an article on that subject for future publication.
Cassettes of individual copies are always available for $3.50. That would be the best way to buy the particular program you want on tape. From time to time we will make programs of particular interest, available on tape separately (like BASMON this month). We are also considering publishing collections of programs on tape, every few months. E.g. a games tape, a utilities tape, etc.
Program comments do not have to be typewritten but please make sure that they are written very legibly. Some of our contributors include their comments and descriptions as REM statements at the front of the program which we then delete for printing. The ideal, from our point of view would be to send your comments on an Electric Pencil Cassette (with lower case letters). Yes, we do LLIST programs for publication.
Your comments concerning the readability of the December issue versus that of January are noted. Unfortunately, it requires far more effort to lay out and correct the magazine using a typewriter. So, for the moment we must stick to the dot-matrix printer driven by a word processing program. Perhaps, as circulation increases, we will be able to afford a daisy-wheel printer and satisfy both requirements - Ed.).

From: RON KEHN, Korumburra, Vic.
If a particular month's software is of particular interest to me, is it possible to get a cassette just for that month, or must it be as part of an annual subscription to magazine and tape? I believe it is possible for about $20.00, to get a kit to modify the TRS-80 to lower case. I am sure that other readers and myself would be interested in further information about this. Also, let's have the article on the RS232 printer interface very soon.
(Yes Ron, you may purchase a single cassette of any month's prorams for $3.50. I guess our order form has caused some confusion here by not being specific enough on this point. We have changed it from this month on. As far as the lower case conversion is concerned, we use one ourselves and it costs much less than $20.00 for parts. However, it does not work with the ordinary Level 2 BASIC. We will describe how to do it in the near future. It will probably be the April or May issue before we can describe the RS232 printer interface but, if we can speed it up, we will. Ed.)

From: TONY MERITON, 85 Milton St., Elwood 3184
Tony is looking for a cheap expansion interface, new or used, with or without memory. (If anyone can help, please contact him direct - Ed)

From RONALD J. SULLY, Loganholme Qld.
Firstly, may I congratulate you on the concept of MICRO-80. I believe it to be the type of magazine which meets the much needed requirements of would-be programmers, (albeit adventurous), like myself, that up till now, were left in a void.
If I may, I would like to bring your attention to, what I believe to be, errors in some articles in both issues (Dec. '79 and Jan. '80.) If I am wrong, then I stand corrected but, if what I am about to say is true. then I believe I may save some of your readers heartache and frustration. Many of the errors I found are merely typing errors and any programmer worth his salt will discover them for himself and carry out the appropriate corrections so I

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542

have no intention of bringing these errors to your attention.
The first error that I believe to be significant is in Issue1, Page 17, in the illustration of the CTR-40 cassette modification. The relevant paragraph (7), directs that a piece of wire be soldered to point B (common with the yellow wire). The fact is that the yellow wire is at point C, not B. I carried out the modifications explained by you and read the illustration with point C and B reversed. I thank you for the advice and instructions as I find the CTR-41 is now much more convenient to use.
Secondly, as an enthusiastic but ignorant assembly language programmer without TBUG or an Editor/Assembler etc), on receipt of Issue 2, I eagerly jumped into Mr. Ian Vagg's MONITOR IN BASIC program, believing I could use it to load the m/c language subroutine of Mr. Peter Hartley's program, GAME OF LIFE However, this was not to be. Firstly, I encountered a lot of problems in getting the Monitor to work. The first problem involves the Command Codes in conjunction with the Test Input Command subroutine starting at line 1000. The essence of the problem was that when I attempted to modify memory in HEX by entering a number containing D (e.g. 4DH, CDH etc.) the Test Input Command subroutine would read it as D for DUMP and send me off to subroutine 8000 and ask the question 'TITLE'? I solved that by using "T" (for TAPE???) instead of D. To digress for a moment - the program I was using as a test case was LOADER as published in Issue 1. I will say now that I am most impressed with it and use it constantly even though I have the hassle of loading it. But, back to MONITOR in BASIC:
I was a bit slow in picking up the next problem which is by no means all that important but it did prove annoying. Both the DUMP and LOAD subroutines, commencing lines 8000 and 8000 respectively are entered by way of a conditional GOTO but each have the instruction RETURN at lines 8200 and 9200 respectively. It would not have been satisfactory to simply have each of the lines (8200 and 8200) read GOTO 20 because of the CLS at line 20 so I amended those two lines as follows:
(8200) 8200 INPUT "PRESS ENTER TO CONTINUE": X:GOTO 20
and that solved the problem. Another method of course could be a simple timing loop.
The next minor problem is that line 8005 will need to include a CLEAR 1000 (or whatever) to allow users to load a data tape as the first command after power-up. A better alternative would be to include the CLEAR statement in line 20 thereby covering the requirements of either subroutine 8000 - or 8000 -
The last, but by all means the most significant, problem with the program is the "LOAD MEMORY FROM TAPE" subroutine.
I found that not only does this subroutine NOT work but the present format produced some very disturbing effects (like freeze up). The problem lies in Line 8145, the POKE statement. Firstly, it is not possible to POKE into START, the start address as the least possible number POKEd into would be START + 2 (2 being the starting value of C) but in practice I found that the first memory address POKEd by that line was in fact START + 4 (it depends on how many characters make up the values to be POKEd - B$). Then, because of the condition laid down in line 8130 and 8140/8142, C would be incremented by at least 2 (the number of characters in B$) before line 8145 was encountered and something was POKEd again.

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542

To illustrate this, the following is an example of what occurred with me when I tried to load LOADER:

	Memory address		Value	
START =	31888	7C90H		
	31889	7C91		
START + C(2)=	31890	7C92		
	31891	7C93		
	31892	7C94	76	4C
	31893	7C95		
	31894	7C96		
	31895	7C97	79	4F
		etc		

31892 (dec) 7C94H was the first memory address POLEd by subroutine 8000 and thereafter it was every third (if there were two characters in the value POKEd), whereas START (31888 7C90H) should have been POKEd with the value 76 (4CH) and START + 1 (31889 7C91H) should have been POKEd with the value 79 (4FH) and so on). In the illustration above, where no value is shown against the memory address, the value was in fact whatever happened to be there!! So, under those conditions anything could happen (and it did!!) if the m/c language was called under the SYSTEM function. (It is rather tedious to enter a m/c language program via the keyboard three or four times).
I overcame the problem by adding or changing the following lines:
8035 ME=START
8145 D=D+S:.....POKE ME,VAL(B$):ME=ME+1:B$="".....(etc)
The above lines then only allowed the memory address to be incremented by 1 (and only 1) if and when a value was POKEd.
Onwards and upwards.!!
The write up (page 14) for the MONITOR IN BASIC claims the program is suitable for 4K. (I have a 16K machine). But, having sorted out all the previous problems I used LOADER to load MONITOR IN BASIC and was advised (by LOADER) that the end address was 51DBH. If you add another 1000 bytes on top of that to cater for the CLEAR 1000 statement it obviously exceeds the maximum address 4FFF of a 4K machine.
Which brings me to the next problem, THE GAME OF LIFE. The stat address of the m/c language subroutine is 5000H which means one cannot use the MONITOR IN BASIC program to load the GAME OF LIFE. GAME OF LIFE obviously needs at least 16k, so why would it not have been possible to locate the m/c language subroutine in high memory, as I believe is usually done, say from 7000H to 719AH? Perhaps you could consider that for publication in the March issue of MICRO-80.
Also, I was confused by the section that explains GAME OF LIFE (page 12) that stated that the entry point for the m/c language subroutine was 000H when the values POKEd into 16525 & 16527 (62 and 81 respectively) led me to believe that the entry address was 513EH. Am I right in believing the entry address is POKEd into 16526 and 16527 (LSB first)?
Apart from all I have said previously, my thanks go to you and Mr. Vagg for the BASIC IN MONITOR program. I hope to see many more m/c language programs in MICRO-80 (particularly those I can load using MONITOR IN BASIC).
Having pointed out some errors, I would now like to make some suggestions for the magazine which I feel will be a lot of help to some of the less experienced programmer. Have you considered including a HELPFUL HINTS section? By this I mean some programming method that will achieve either a quicker, more accurate or more economical method of what could be termed a consistent requirement within a program. For instance, there have been many occasions when I have needed a random number within certain values so I developed a formula. To illustrate: If the random number was to be between A

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542

and B.(A being the lower value and B being the greater value) then the simple statement:

R = INT ((B-A)*RND(0))+A would achieve this

If A = 12 and B = 45 then the line would read

R = INT (33*RND(0))+12

If you refer to line 3005 of Mr. C. Bartlett's RICOCHET program you will see that the above method is far more efficient.

Another requirement that programmers often need is a method or sorting numbers or names in ascending or descending order. I have encountered a few different methods, (the one I developed took far too much time to process). Perhaps you may be able to publish a method or methods to achieve this?

I think that these things I have been talking about could be considered "tricks of the trade" and until one has discovered a few programs are produced which are unnecessarily long, (4K is not very much) and slow (watching a "bullet" move across the screen at a snail's pace is not very realistic) so I believe a HELPFUL HINTS or TRICKS OF THE TRADE column will be appreciated by all readers of varying programming ability. And, of course, those same readers will gladly contribute any hints that they may have accumulated. It will also fill some empty spaces currently appearing in the magazine.

Another readers' service which I believe MICRO-80 could give is possibly a review or at least an opinion of various items of hardware being offered on the market. One that comes to mind is the Light Pen currently advertised by DICK SMITH. Advertisements very seldom tell the full story so perhaps MICRO-80 could give various items a "road test" and advise readers of the outcome. The review need not specifically refer to new products but, for instance, could possibly explain the workings of and the need for the Expansion Interface. What does it do beside providing the medium for expansion? Why is it needed to be able to operate the real-time clock? Are there various ways of adding on peripherals without it? (I read with interest your article in Issue 2 on the memory expansion without the interface). Is it possible to address the 255 I/O ports without the Expansion Interface?

I think I may have said enough for the time being so again I congratulate you on the concept of MICRO-80, thank you for giving me access to many other programs and I wish you success for the future. (If you succeed, I'll benefit).

(Thank you Ron both for the nice things you said about the magazine and, more importantly, for the very worthwhile contribution to the magazine which you have made. You have done a first class job debugging MONITOR IN BASIC. I am just sorry that you had to do it at all. Anyway, the article this month should put it right but if any reader wants to persevere with the original version, please make changes Ron has worked out.

You are also right about the error in Issue 1, page 17 concerning the CTR-41 cassette modification. The yellow wire is at point C, not D. In fact, we have heard of one recorder which had a different coloured wire so it might be best just to go by the diagram.

Your comments concerning GAME OF LIFE are noted. The article DEFUSRING GAME OF LIFE this month should answer the entry pointquestion. As far as relocating the m/c language subroutine at 7000H, we will print a HEX listing for that in the March Issue.

Your suggestion concerning a TRICKS OF THE TRADE column has been echoed by a number of other readers and starts this month. The "road testing" of hardware and software products offered for the TRS-80 is one of our original objectives and wehope to be able to start this in the near future. Once again, thanks for your contribution. - Ed.)

From: DAVID COUPE, Moonee Ponds, Vic.
You may get a guide for future articles or series of articles by listing, each month, suggestions made by readers. Feedback from other readers would clearly indicate to you which subjects your readers would like to have dealt with. I would welcome a series of articles fully exploring the file-creation, file-handling and file-manipulation capabilities of the TRS-80.
(No sooner said than done - David. The READERS REQUESTS LIST starts this month your request at the top of the Articles section. Thanks for the suggestion - Ed.)

From TREVOR JAMES, Kyabram, Vic.
I have received the last two Issues of MICRO-80 and have found the mag. very good except for lack of instructions on some progams (eg. MONITOR IN BASIC).
I also received Issue 2 on cassette and found it to be of very poor quality as the first program on side 2 had no index ("F") and, when loaded with CLOAD, loaded rubbish and not the program published. The only program which did run and load corectly was "H" with the others having various faults.
Also, could you print "RICOCHET" in L2/4K?
Could your mag. please advise me on the current "CHESS" programs available and their pro''s and con's?
(Your comments about lack of instructions are noted, Trevor. Now we have two Associate Editors, we should be able to overcome this problem completely. Sorry about your cassette. By now you should have received a replacement, free of charge. Copy cassettes from the first two Issues were made using entirely audio techniques. You may need to use a slightly higher volume setting than you are used to, to load thm satisfactorily. Future cassettes will be copied using digital techniques and should load at normal levels.
Charlie Bartlett, the author of Level 1 RICOCHET has just converted his machine to Level 2 and is developing a L2 version of RICOCHET which we hope to publish next month.
Your question concerning CHESS programs has been put in our READERS' REQUESTS column, we will try to get to it as soon as possible. - Ed.)

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542


```
1 REM HANGMAN - LEVEL ONE - APPROX 8K NEEDED IN ORIGINAL FORM
5 REM**************************************************
10 REM FOR 4K VERSION ONE - AUTOMATIC WORD SELECTION
11 REM DELETE ALL REM LINES, LINES 1410,1440
12 REM LINES 2150 TO 2175, LINES 3000 TO 3030
13 REM AND LINES 1800 TO 1890
14 REM FOR 4K VERSION TWO - USER INPUT WORDS
15 REM DELETE ALL REM LINES, LINES 1410,1440
16 REM LINES 2160 TO 2175, LINES 3000 TO 3030
17 REM LINES 1445 TO 1490 AND DATA LINES 19999 TO 20009
18 REM
20 REM**********************************************
50 REM COPYRIGHT PETER G. HARTLEY
60 REM (C) 1980  57 MAIN AVENUE, FREWVILLE, S.A. 5063.
70 REM PHONE (08)-79-4061 FOR CUSTOM PROGRAMME DEVELOPMENT
80 REM ANY-TIME            AT REASONABLE RATES.
90 REM THIS PROGRAMME SPECIALLY COMMISSIONED BY 'MICRO-80'
95 REM>>><<<>>><<<>>><<<>>><<<>>><<<>>><<<>>><<<>>><<<>>><<<>>>
100 GOSUB30000:GOSUB1400
110 GOSUB200:GOTO100
180 A(3)=A(0)-1:RETURN
200 A(3)=INT((63-(A(8)-11)*2)/2)+63:A(4)=0
205 FORA(0)=11TOA(8):A(5)=(A(0)-11)*2+A(3)
210 IF((A(A(0))>0)*(A(A(0))<27))THENPRINT@A(5),"? ";:A(4)=A(4)+1
220 IFA(A(0))=27THENPRINT@A(5),"- ";
230 NEXTA(0):A(5)=251:PRINT@116,"WRONG 'UNS":A(3)=0
240 PRINT@0,"";:INPUT"GUESS";A(6),PRINT@0,"          ";:GOSUB500
245 IFA(A(6)+30)<>0GOTO240
250 A(7)=0:A(A(6)+30)=1:FORA(0)=11TOA(8):IFA(A(0))=A(6)GOSUB400
260 NEXTA(0):IFA(7)=0GOSUB500:X=24:Y=25
265 IFA(3)=9THENRETURN
270 IFA(4)<>0GOTO240
280 PRINT@312,"C O N G R A T U L A T I O N S !":FORA=0TO2000:NEXTA:RETURN
400 A(2)=(A(0)-11)*2+A(3):PRINT@A(2),A$;:A(7)=1:A(4)=A(4)-1:PRINT@0,:RETURN
500 PRINT@A(5),A$;:A(5)=A(5)+64:A(3)=A(3)+1
510 ONA(3)GOSUB2000,2010,2030,2040,2060,2080,2090
520 IFA(3)<>8THENRETURN
530 GOTO2150
600 RESTORE:FORA(1)=0TOA(6):READA$:NEXTA(1):RETURN
610 DATA" ",A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z,"-"
1400 CLS:PRINT@278,"H A N G M A N"
1410 PRINT@512,"";:INPUT"WHO SETS THE WORD? (1) YOU OR (0) ME";A(0)
1420 FORA(9)=11TO57:A(A(9))=0:NEXTA(9)
1440 ONA(0)GOTO1800
1445 CLS:RESTORE:FORA(0)=0TO27:READA$:NEXTA(0)
1450 A(0)=RND(6):REM CHANGE NUMBER FOR MORE WORDS IN DATA
1460 READA(1):IFA(1)=99THENA(0)=A(0)-1
1475 IFA(0)<>0GOTO1460
1480 FORA(0)=11TO30:READA(A(0)):IFA(A(0))<28THENNEXTA(0)
1490 GOTO180
1800 CLS:PRINT"INPUT THE WORD NOW - MAKE SURE NO-ONE'S PEEKING!"
1805 PRINT@832,"  "
1810 FORA(0)=11TO30:A(5)=832+A(0)
1820 PRINT@192,"INDICATE THE END OF THE WORD WITH >>> 99 <<<"
1830 PRINT"IF YOU MAKE A MISTAKE - ENTER >>> 77 <<<"
1840 PRINT"DON'T ENTER THE NEXT LETTER UNTIL THE LAST IS DISPLAYED"
1845 PRINT"TO ENTER A SPACE ENTER >>> 0 <<<"
```

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542


```
1847 PRINT"TO ENTER A HYPHEN       >>> 27 <<<"
1849 PRINT@A(5),"";:INPUTA(6)
1850 IFA(6)=77GOTO1800
1860 IFA(6)=99THENCLS:GOTO180
1870 IFA(6)>27GOTO1800
1880 GOSUB600:PRINT@A(5),A$;:A(A(0))=A(6):NEXTA(0)
1885 PRINT:PRINT" SORRY - WORD BUFFER FILLED - SOMETHING SHORTER PERHAPS?"
1890 GOTO1805
2000 FORX=25TO80:SET(X,38):NEXTX:RETURN
2003 DATA3,15,13,16,21,20,5,18,0,15,16
2010 FORY=38TO32STEP-1:SET(25,Y):SET(50,Y):NEXTY:FORX=25TO50:SET(X,32):NEXTX:RET
URN
2030 FORY=38TO8STEP-1:SET(65,Y):NEXTY:RETURN
2040 Y=39:FORX=58TO65:Y=Y-1:SET(X,Y):NEXTX
2050 Y=39:FORX=72TO65STEP-1:Y=Y-1:SET(X,Y):NEXTX:RETURN
2060 FORX=65TO37STEP-1:SET(X,8):NEXTX
2070 Y=16:FORX=65TO58STEP-1:Y=Y-1:SET(X,Y):NEXTX:RETURN
2080 FORY=8TO14STEP2:SET(37,Y):NEXTY:RETURN
2090 FORX=35TO39:SET(X,14):SET(X,19):NEXTX:SET(34,15):SET(34,18):SET(40,18)
2100 FORY=16TO17:SET(33,Y):SET(41,Y):NEXTY:SET(36,16):SET(38,16):FORY=20TO26
2110 SET(37,Y):NEXTY:X=37:FORY=26TO31:X=X-1:SET(X,Y):NEXTY:X=37:FORY=26TO31
2120 X=X+1:SET(X,Y):NEXTY:SET(30,31):SET(44,31):FORX=35TO39:SET(X,22):NEXTX
2130 X=35:FORY=23TO25:X=X-1:SET(X,Y):NEXTY:X=39:FORY=23TO25:X=X+1:SET(X,Y)
2140 NEXTY:PRINT@484,"LAST CHANCE !";:RETURN
2150 FORX=25TO50:RESET(X,32):NEXTX:PRINT@484,"GASP, MOAN, THUD";
2155 A(3)=9:FORA(10)=11TOA(8):A(6)=A(A(10)):GOSUB600:GOSUB250:NEXTA(10)
2160 FORA(9)=1TO4:FORY=35TO14STEP-1:FORX=25TO44
2165 IFPOINT(X,Y-2)=1THENSET(X,Y):RESET(X,Y-2)
2170  SET(37,14):SET(37,12):NEXTX:NEXTY
2175 NEXTA(9)
2180 RETURN
3000 REM THIS ROUTINE IS TO HELP YOU CREATE YOUR OWN DATA LINES
3005 CLS
3010 GOSUB30000
3020 A(1)=512
3030 PRINT@0,"";:INPUTA(0):PRINT@A(1),A(0);:A(1)=A(1)+4:GOTO3030
19999 DATA99
20000 DATA20,1,14,4,25,0,13,1,14,1,7,5,18,99,2,15,24,0,15,6,0,20,18,1,14
20001 DATA19,9,19,20,15,18,18,99,13,9,3,18,15,0,5,9,7,8,20,25
20002 DATA99,20,8,5,0,3,8,9,16,19,0,1,18,5,0,4,15,23,14,99,23,1,14,20,5,4
20003 DATA0,27,0,16,18,15,7,18,1,13,13,5,19,99
20004 DATA1,18,5,0,25,15,21,0,8,1,22,9,14,7,0,6,21,14,99
20009 DATA999
30000 A=1:B=2:C=3:D=4:E=5:F=6:G=7:H=8:I=9:J=10:K=11:L=12:M=13:N=14:O=15
30010 P=16:Q=17:R=18:S=19:T=20:U=21:V=22:W=23:X=24:Y=25:Z=26
30020 RETURN
```

***** CORRECTION *****

PLEASE MAKE THE FOLLOWING CORRECTION TO THE "FILES" PROGRAM IN THIS ISSUE:-

LINE 680, AFTER - DIM B$(LE): ADD - DIM C$(LE):

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542


```
*** AMAZIN ***
10 'ORIGINAL CONCEPT BY JACK HAUBER, LOOMIS SCHOOL, WINDSOR,
   CT., U.S.A.
20 'THIS VERSION FOR '80 (C)OPYRIGHT 1975, PETER G. HARTLEY,
   MAIN AVENUE, FREWVILLE, SOUTH AUSTRALIA.
100 CLS:PRINT"IT MAY TAKE A FEW MOMENTS - YOU'LL HAVE TO PRESS A KEY FOR ME."
110 CLEAR:RANDOM:DEFINTA-Z:DIMW(33,7),V(33,7)
120 H=30:V=6
130 Q=0:Z=0:X=RND(H):FORI=1TOH
140 FORI=1TOH:IFI=XTHENPRINTCHR$(149);" ";:NEXT:ELSEPRINTCHR$(157);CHR$(140);:NE
XT
150 PRINTCHR$(149):C=1:W(X,1)=C:C=C+1:R=X:S=1:GOTO200
160 IFR<>HTHENGOTO180ELSEIFS<>VTHENGOTO170ELSES=1:R=1:GOTO190
170 R=1:S=S+1:GOTO190
180 R=R+1
190 IFW(R,S)=0GOTO180
200 IFR-1=0ORW(R-1,S)<>0THENGOTO330ELSEIFS-1=0ORW(R,S-1)<>0THENGOTO250ELSEIFR=HO
RW(R+1,S)<>0THENGOTO210ELSEX=RND(3):ONXGOTO480,490,510
210 IFS<>VTHENGOTO220ELSEIFZ=1THENGOTO240ELSEQ=1:GOTO230
220 IFW(R,S+1)<>0GOTO240
230 X=RND(3):ONXGOTO480,490,550
240 X=RND(2):ONXGOTO480,490
250 IFR=HORW(R+1,S)<>0THENGOTO290ELSEIFS<>VTHENGOTO260ELSEIFZ=1THENGOTO280ELSEQ=
1:GOTO270
260 IFW(R,S+1)<>0GOTO280
270 X=RND(3):ONXGOTO480,510,550
280 X=RND(2):ONXGOTO480,510
290 IFS<>VTHENGOTO300ELSEIFZ=1THENGOTO320ELSEQ=1:GOTO310
300 IFW(R,S+1)<>0GOTO320
310 X=RND(2):ONXGOTO480,550
320 GOTO480
330 IFS-1=0ORW(R,S-1)<>0THENGOTO410ELSEIFR=HORW(R+1,S)<>0THENGOTO370ELSEIFS<>VTH
ENGOTO340ELSEIFZ=1THENGOTO360ELSEQ=1:GOTO350
340 IFW(R,S+1)<>0GOTO360
350 X=RND(3):ONXGOTO490,510,550
360 X=RND(2):ONXGOTO490,510
370 IFS<>VTHENGOTO380ELSEIFZ=1THENGOTO400ELSEQ=1:GOTO390
380 IFW(R,S+1)<>0GOTO400
390 X=RND(2):ONXGOTO490,550
400 GOTO490
410 IFR=HORW(R+1,S)<>0THENGOTO440ELSEIFS<>VTHENGOTO420ELSEIFZ=1THENGOTO430ELSEQ=
1:GOTO500
420 IFW(R,S+1)<>0THENGOTO430ELSEX=RND(2):ONXGOTO510,550
430 GOTO510
440 IFS<>VTHENGOTO450ELSEIFZ=1THENGOTO470ELSEQ=1:GOTO460
450 IFW(R,S+1)<>0GOTO470
460 GOTO550
470 GOTO600
480 W(R-1,S)=C:C=C+1:V(R-1,S)=2:R=R-1:IFC=H*V+1THENGOTO610ELSEQ=0:GOTO200
490 W(R,S+1)=C
500 C=C+1:V(R,S+1)=1:S=S-1:IFC=H*V+1THENGOTO610ELSEQ=0:GOTO200
510 W(R,S)=C:C=C+1:IFV(R,S)=0THENGOTO520ELSEV(R,S)=3:GOTO530
520 V(R,S)=2
530 R=R+1
540 IFC=H*V+1THENGOTO610ELSEGOTO330
550 IFQ=1THENGOTO580ELSEW(R,S+1)=C:C=C+1:IFV(R,S)=0THENGOTO560ELSEV(R,S)=2:GOTO5
```

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542


```
560 V(R,S)=1
570 S=S+1:IFC=H*V+1THENGOTO610ELSEGOTO200
580 Z=1:IFV(R,S)=0THENGOTO590ELSEV(R,S)=3:Q=0:GOTO600
590 V(R,S)=1:Q=0:R=1:S=1:GOTO190
600 GOTO160
610 D$=""
620 D$=INKEY$:IFD$=""GOTO620
630 FORJ=1TOV:PRINTCHR$(149);:FORI=1TOH:IFV(I,J)<2THENPRINT" ";CHR$(149);:NEXT:E
LSEPRINT"  ";:NEXT
640 PRINT:FORI=1TOH:IFV(I,J)=3ORV(I,J)=0THENPRINTCHR$(157);CHR$(140);:NEXT:ELSEP
RINTCHR$(149);" ";:NEXT
650 PRINTCHR$(149):NEXT:PRINT"ANOTHER AMAZIN MAZE WILL FOLLOW SHORTLY: PRESS A K
EY WHEN READY.";:GOTO110
5

'***  BIORHYTHM CURVE PLOTTER. ***
7 '  AUTHOR:  BERNIE SIMSON,  18 BULLER TCE, CHELTENHAM
8 '                           STH AUST.   5014.   PH 47 7528.
9 '  NOTE: MINIMUM HARDWARE REQUIREMENTS: LEVEL ii, 4K RAM.
10 CLS:PRINT@80,'THE  THEORY  OF  BIO-RHYTHMS":PRINT@144,"-----------------------
-------"
20 PRINT:PRINT"RESEARCHERS BELIEVE THAT FROM BIRTH, 3 DIFFERENT CYCLES START TOO
PERATE IN YOUR BODY, CALLED BIO-RHYTHMS, WHICH INFLUENCE YOUR"
30 PRINT"THOUGHTS, FEELINGS AND ACTIONS.  KNOWING  THE STATE OF THE 3    CYCLES
(PHYSICAL, EMOTIONAL, INTELLECTUAL) AT GIVEN TIMES CAN"
40 PRINT"PREPARE YOU FOR YOUR CHANGING MOODS."
50 PRINT:PRINT" THE TRS-80 WILL PLOT YOUR 3 BIO-RHYTHM CURVES FOR YOU TO STUDY."
:PRINT:INPUT"PRESS 'ENTER' TO CONTINUE";X
80 CLS:INPUT"NAME OF PERSON TO BE ANALYZED";P$
100 INPUT"TODAY'S DATE (DD,MM,YY) ";TD,TM,TY
110 IFTM<1ORTM>12GOTO160
120 FORL=1TOTM:READMT:NEXT:RESTORE
130 DATA 31,28,31,30,31,30,31,31,30,31,30,31,31
135 IFTM=2IFTY/4=INT(TY/4)MT=29
140 IFTD>0ANDTD<=MTGOTO170
160 PRINT"  HA HA, VERY FUNNY. TRY AGAIN":GOTO100
170 INPUT"YOUR BIRTH DATE (DD,MM,YY) ";BD,BM,BY
180 IFBM<1ORBM>12GOTO240
182 D1=TD+(TM*100)+(TY*10000):D2=BD+(BM*100)+(BY*10000)
184 IFD2>D1GOTO240
190 FORL=1TOBM:READMT:NEXT:RESTORE
195 IFBM=2IFBY/4=INT(BY/4)MT=29
200 IFBD>0ANDBD<=MTGOTO250
240 PRINT"  HA HA, VERY PECULIAR. TRY AGAIN":GOTO170
250 D=TD:M=TM:Y=TY:GOSUB2000:D1=DS
260 D=BD:M=BM:Y=BY:GOSUB2000:D2=DS
270 D3=D1-D2+1
280 P=D3-(INT(D3/23)*23)
290 E=D3-(INT(D3/28)*28)
300 I=D3-(INT(D3/33)*33)
390 CLS
400 FORL=3TO34:SET(0,L):NEXT
410 FORL=385TO435STEP2:PRINT@L,"-";:NEXT
415 PRINT@12,"*  *  *  ";P$;"'S  BIO-RHYTHMS  *  *  *";
416 L1=12:FORZ=1TO11:L1=L1+64
```

```
420 FORL=L1TOL1+40STEP13:PRINT@L,"!";:NEXTL,Z
460 FORL=780TO830STEP13:PRINT@L,CHR$(91);:NEXT
490 PRINT@768,CHR$(91);:PRINT@832,"TODAY";
500 FORL=1TOTM:READMT:NEXT
520 FORL=841TO889STEP13:TD=TD+7
540 IFTD>MTTHENTD=TD-MT:TM=TM+1
560 IFTM>12TM=1
580 PRINT@L,TD;"/";TM;
600 NEXT
620 PRINT@961,"PHYSICAL";:FORL=20TO32
640 SET(L,46):NEXT
660 PRINT@982,"EMOTIONAL";:FORL=64TO76STEP2
680 SET(L,46):NEXT
700 PRINT@1004,"INTELLECTUAL";:FORL=114TO126STEP4
720 SET(L,46):NEXT
740 C=1:F=23:N=P:GOSUB2500
760 C=2:F=28:N=E:GOSUB2500
780 C=3:F=33:N=I:GOSUB2500
800 PRINT@184,"ACTIVE";:PRINT@248,"STAGE";:PRINT@121,CHR$(91);:PRINT@313,CHR$(92
);
820 PRINT@632,"PASSIVE";:PRINT@696,"STAGE";:PRINT@569,CHR$(91);:PRINT@761,CHR$(9
2);
840 PRINT@437,CHR$(93);" CRITICAL";
860 GOTO860
2000 ' CALC "DAYS FROM 1900" ROUTINE
2020 Y1=Y:IFM<3Y1=Y1-1
2030 DS=Y*365+INT(Y1/4):IFM=1GOTO2050
2040 FORL=1TOM-1:READMT:DS=DS+MT:NEXT:RESTORE
2050 DS=DS+D:RETURN
2450 ' SINE CURVE ROUTINE
2500 IN=.28:DC=360*.0174533:FORL=1TO102
2520 IFC=3IFL/5=INT(L/5)GOTO2560ELSE2800
2540 IFC=2IFL/2=INT(L/2)GOTO2560ELSE2800
2560 YC=SIN(N/F*DC)*15
2580 SET(L,19-YC)
2800 N=N+IN:NEXT:RETURN
```

APPLICATION FORM

To MICRO-80, POBOX 213, GOODWOOD, SA 5034. Phone: (08) 381 8542

Please send me:

- CURRENT ISSUE .....$ 2.50
- CASSETTE ONLY .....$ 3.50
- 12 MONTH SUBSCRIPTION, MAGAZINE ONLY $24.00
- 12 MONTH SUBSCRIPTION,MAGAZINE PLUS CASSETTE $60.00
- START SUBSCRIPTION FROM F.....(Dec.79 Issue 1)

(Make cheques payable to MICRO-80)

NAME : .....

ADDRESS: .....

.....POST CODE .....

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542

```
1 REM MKS$sSQR~1.~VALINSTR=PEEK+m~*GET@↑#V$N#FLOF~#DELETE+ FIX:EXP~PEEKEND_
ENDLOFCOSs$r#q$p#MKS$CTAN~:EXP~OENDLOF!IsA~MKS$DEFLOF"EXP~'6E'"RIGH'@".@COS[USRC
DBLMKS$(S@R~ERRSTEP!IsA~~#!85@ RIGHT$x2EXP~INKEY$ABS>ERR23@!TRON~"4@LPRINT2ABS~
2 CLS:PRINTCHR$(23);"



   *** INITIALIZING ***"
3 Y=32256:Z=PEEK(16548)+PEEK(16549)*256+6
5 FORX=ZTOZ+220:A=PEEK(X):B=B+A:IFA=128THENA=0:POKEY,A:Y=Y+1:NEXTX:ELSEPOKEY,A:Y
=Y+1:NEXTX
7 IFB(>27340THENCLS:PRINTCHR$(23);"



* * * CHECKSUM ERROR !!! * * *


":END:ELSEPOKE16526,126:POKE16527,126:X=USR(0)
20 REM *** DATA PROCESSING AND FILING SYSTEM ***
30 '  *** BY EDWIN R. PAAY 39 FAIRVIEW GRV. HACKHAM WEST 5163 ***
40 CLS:PRINT"

            --- SELECT REQUIRED FUNCTION ---
50 PRINT"

              1 ==>   ENTER NEW DATA TO FILE.
              2 ==>   SEARCH FILE FOR DATA.
              3 ==>   EXIT FROM PROGRAM.
              4 ==>   EDIT FILE.
              5 ==>   LIST ALL DATA.
60 PRINT"

":A=0:FG=0:H=0:PRINT@780,"";
70 A$=INKEY$:A=A+1:IFA$<>""GOTO110
80 GOSUB250  :GOTO70
110 IFA$>"6"ORA$<"1"THEN70   ELSEB=VAL(A$):ONBGOTO120  ,310  ,540  ,560  ,670
120 '  ** ENTER NEW DATA **
130 CLS:PRINT"TYPE IN DATA SEPARATED WITH COMMA'S , HIT  <ENTER>  AFTER EACH GR
OUP OF DATA, THEN TYPE <ENTER> AGAIN WHEN ";CHR$(34);"READY";CHR$(34);" APPEARS
ON  THE SCREEN.

140 PRINT"EXAMPLE : DATA , DATA , DATA , DATA   <ENTER> - <ENTER>

TYPE  ";CHR$(34);CHR$(92);CHR$(34);" TO STOP ANYTIME WHILE CURSOR IS BLINKING !

              * * * ";MEM;" BYTES FREE. * * *

          ---=== READY TO ACCEPT DATA ! ! ! ===---

150 A=2:BUFFER=32511:POKEBU,136:BU=BU+1:POKEBU,34:PRINTCHR$(34);:BU=BU+1:E=0
160 GOSUB250  :A$=INKEY$:IFA$=""THEN160
170 IFA=>250THENCLS:PRINT"          *** ERROR *** ERROR *** ERROR *** ERROR ***
```

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542


```
---=== DATA FIELD TO LONG CANNOT ENTER LAST CHARACTER ! ===---":ELSEGOTO150
180 FORC=0TO3000:NEXT:GOSUB280  :GOTO130
190 PRINTA$;:GOSUB250  :A=A+1:IFA$=",THENPOKEBU,34:BU=BU+1:POKEBU,44:BU=BU+1:PO
KEBU,34:BU=BU+1:PRINTCHR$(8);CHR$(34);",";CHR$(34);:A=A+3:GOTO160
200 IFA$=CHR$(10)GOTO40
210 IFA$=CHR$(13)THENPOKEBU,34:BU=BU+1:POKEBU,44:BU=BU+1:POKEBU,34:BU=BU+1:POKEB
U,ASC("@"):BU=BU+1:POKEBU,34:BU=BU+1:POKEBU,44:A=A+5:GOTO280
220 IFA$=CHR$(9)THENBU=BU-1:PRINTCHR$(30);:GOTO160
230 POKEBU,ASC(A$):BU=BU+1:GOTO160
240 '  *CURSOR*
250 D=PEEK(16416)+PEEK(16417)*255:E=E+1
260 IFE>6AND(PEEK(D)=95ORPEEK(D)=32)THENPOKED,136:E=0:RETURN
270 IFE>6POKED,32:E=0:RETURN
275 RETURN
280 '  *END OF LINE*
290 POKE16405,0:POKE16526,0:POKE16527,128:X=USR(0)
300 POKE16405,1:NU=PEEK(32483)+PEEK(32484)*256:NU$=STR$(NU)
302 X=PEEK(32480)+PEEK(32481)*255:FORG=2TOLEN(NU$):POKEX+G-2,ASC(MID$(NU$,G,1)):
NEXT
305 GOTO130
310 ' ** SEARCH **
320 RESTORE:CLS:READLE$:LE=VAL(LE$):DIMB$(LE):DIMC$(LE):ONERRORGOTO432  :B$="":F
L=0:Z$=""
325 FORJ=1TOLE:READC$(J):NEXT:GOSUB460  :READH
330 PRINT"
                TYPE ";CHR$(34);CHR$(92);CHR$(34);" TO STOP ! !

  ----=== ENTER DATA TO BE SEARCHED FOR. ===----

340 GOSUB250  :A$=INKEY$:IFA$=""GOTO340
342 PRINTA$;:IFA$=CHR$(10)THEN40  ELSEIFA$=CHR$(8)THEN344  ELSEIFA$<>CHR$(13)TH
ENB$=B$+A$:GOTO340
343 GOTO350
344 GOSUB680  :PRINTCHR$(30);:GOTO340
350 M=0:FORJ%=1TOLE:READB$(J%):GOSUB800:M=M+SS:NEXTJ%:READH$:READH
355 IFM=0THEN350  ELSEFL=1
410 CLS:FORJ=1TOLE:PRINTC$(J);"  ";B$(J):NEXT:PRINT"DATA IS IN LINE :";H;
412 IFFG=1THENRETURN
420 PRINT" <<< >>> CONTINUE SEARCH OR STOP (C/S)";
421 A$=INKEY$:GOSUB250  :IFA$=""THEN421  ELSEIFA$="C"THEN350
422 GOTO40
432 IF(ERR/2)+1=10RESUMENEXT
435 IF(ERR/2)+1<>4THEN450
440 IFERL=350  THENRESUME508
450 PRINT"*** ERROR ";(ERR/2)+1;" IN LINE ";ERL;" ***":END
460 READC$:IFC$="@"THENRETURNELSE460
500 CLS:PRINTCHR$(23)
520 IFFL=1THENPRINT"



** ALL DATA HAS BEEN SEARCHED **":FORJ=0TO1500:NEXT:GOTO320
530 PRINT"
```

```
* * *  DATA NOT FOUND ! * * *":FORJ=0TO1500:NEXT:GOTO320
540 CLS:END
550 ' *** EDIT FILE ***
560 CLS:PRINT"TO DELETE OR EDIT FILES STOP THE PROGRAM AND EDIT DATA LINES    US
ING THE EDIT COMMAND PROVIDED BY BASIC.

IF YOU WISH YOU CAN ENTER DATA AND I WILL TELL YOU WHICH LINES
570 PRINT"CONTAIN THE DATA IN QUESTION.


        ---------- ENTER DATA OR TYPE <  ";CHR$(92);" > ==========

.
580 B$="":ONERRORGOTO630
590 GOSUB250  :A$=INKEY$:PRINTA$;:IFA$=CHR$(10)THEN40  ELSEIFA$=CHR$(13)THEN600
  ELSEIFA$=CHR$(8)THENGOSUB660  :PRINTCHR$(30);:GOTO590  ELSEB$=B$+A$:GOTO590
600 RESTORE:GOSUB460  :READX
610 READC$:IFC$=B$THEN620  ELSE610
620 GOSUB460  :READX:PRINTX;:GOTO610
630 IFERL=610  AND(ERR/2)+1=4THENRESUME650
640 GOTO450
650 PRINT"
                *** SEARCH COMPLETED ***":INPUT"                    TYPE <ENTER> T
O CONT.";A$:GOTO560
660 B$=LEFT$(B$,(LEN(B$)-1)):RETURN
670 ' *** LIST ALL ***
680 RESTORE:ONERRORGOTO730  :READLE$:LE=VAL(LE$):DIMB$(LE):FG=1
690 FORJ=1TOLE:READC$(J):NEXT:GOSUB460  :READX
700 CLS:GOSUB751  :READA$:READX:PRINTCHR$(8);CHR$(8);X;" CONTINUE (Y/N) ?";
710 A$=INKEY$:GOSUB250  :IFA$=""GOTO710
720 IFA$="Y"THEN700  ELSE40
730 IFERL=751  AND(ERR/2)+1=4RESUME40
740 IF(ERR/2)+1=10RESUMENEXT
750 GOTO450
751 FORJ=1TOLE:READB$(J):NEXTJ:GOTO410
800 FORSS=1TOLEN(B$(J%))-LEN(B$)+1
805 IFB$=MID$(B$(J%),SS,LEN(B$))RETURN
810 NEXT:SS=0:RETURN
998 REM*** DATA STARTS HERE ***
1000 DATA"4","MEMORY LOCATION :","FUNCTION :        ","CALL SEQUENCE :  ","COMMEN
TS :        ","@",1000
1010 DATA"0000-01D8","SYSTEM INIT. I/O SUBR.","N.A.","","@",1010
1020 DATA"01D9-03E2","CASSETTE SUBR.","N.A.","","@",1020
1030 DATA"03E3-0457","KEYBOARD DRIVER","N.A.","","@",1030
1040 DATA"0458-058C","VIDEO DRIVER","N.A.","","@",1040
1050 DATA"058D-673","LPRINT DRIVER","N.A.","","@",1050
1060 DATA"0674-070A","INITIALIZE","N.A.","","@",1060
1070 DATA"070B-1607","FLOATING POINT MATH","N.A.","","@",1070
1080 DATA"1608-164F","TABLE LEVII ENTRY POINTS","N.A.","","@",1080
1090 DATA"1650-1820","TABLE BASIC COMMANDS","","","@",1090
1100 DATA"1821-191C","TABLE OF JUMP ADRS. FOR BASIC COMMANDS.","N.A.","","@",110
0
1110 DATA"3000-37DD","RESERVED FOR DMA DEVICES","N.A.","THIS SPACE IS FREE AND C
AN BE USED","@",1110
1120 DATA"37DE","DOS COMUNICATION ADRS","","","@",1120
```

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542

*** SET2 MACHINE LANGUAGE FOR LEVEL2 BASIC ***

```
42EF:  ED 73 DD 7E 31 FD 7E F5 C5 D5 E5 CD 6D 7E 2A A4
42FF:  40 5E 23 56 23 4E 23 46 EB 7E 23 B6 2B 20 F2 03
430F:  03 3A E0 7E C5 16 80 5F 19 11 0A 80 19 EB E1 73
431F:  23 72 23 71 23 70 23 ED 43 E3 7E 3A E0 7E 4F 06
432F:  80 EB 21 FF 7E ED B0 EB 06 05 22 E0 7E 36 20 23
433F:  10 FB 06 03 36 80 23 10 FB 22 F9 40 22 F8 40 22
434F:  FD 40 E1 D1 C1 F1 ED 7B DD 7E C3 CC 06 06 02 21
435F:  FF 7E 7E 04 23 FE 40 20 F9 78 32 E0 7E C9 D9 3E
436F:  C3 32 33 40 21 96 7E 22 34 40 AF 32 D9 7E 32 DA
437F:  7E 32 D8 7E D9 C9 CD E3 03 F5 3A D8 7E B7 20 0A
438F:  F1 FE 0D C0 3E 01 32 D8 7E F5 F1 E5 D5 21 D1 7E
439F:  ED 5B D9 7E 19 13 7B FE 07 D4 C4 7E 7E ED 53 D9
43AF:  7E D1 E1 C9 E5 21 D8 7E AF 77 23 77 23 77 5F E1
43BF:  C9 52 55 4E 33 30 30 0D 80 80 80 80
```

*** SET2 MACHINE LANGUAGE FOR DISK BASIC ***

```
6A2A:  ED 73 DD BE 31 FD BE F5 C5 D5 E5 CD 6D BE 2A A4
6A3A:  40 5E 23 56 23 4E 23 46 EB 7E 23 B6 2B 20 F2 03
6A4A:  03 3A E0 BE E5 16 80 5F 19 11 0A 80 19 EB E1 73
6A5A:  23 72 23 71 23 70 23 ED 43 E3 BE 3A E0 BE 4F 06
6A6A:  80 EB 21 FF BE ED B0 EB 06 05 22 E0 BE 36 20 23
6A7A:  10 FB 06 03 36 80 23 10 FB 22 F9 40 22 F8 40 22
6A8A:  FD 40 E1 D1 C1 F1 ED 7B DD BE C3 CC 06 06 02 21
6A9A:  FF BE 7E 04 23 FE 40 20 F9 78 32 E0 BE C9 D9 3E
6AAA:  C3 32 33 40 21 96 BE 22 34 40 AF 32 D9 BE 32 DA
6ABA:  BE 32 D6 BE D9 C9 CD E3 03 F5 3A D8 BE B7 20 0A
6ACA:  F1 FE 0D C0 3E 01 32 D8 BE F5 F1 E5 D5 21 D1 BE
6ADA:  ED 5B D9 BE 19 13 7B FE 07 D4 C4 BE 7E ED 53 D9
6AEA:  BE D1 E1 C9 E5 21 D8 BE AF 77 23 77 23 77 5F E1
6AFA:  C9 52 55 4E 33 30 30 0D 80 80 80 80
```

```
                00100 ;SET2 BY E.R.PAAY VER 2.0
                00110 :THE FOLLOWING LINES WILL INITIALIZE SET2 FOR USE WITH BASIC

7DDE            00120         ORG     7DDEH
7DDE 21FA7D     00130 INIT    LD      HL,START
7DE1 3AC441     00132         LD      A,(41C4H)
7DE4 32697F     00134         LD      (VECTR0),A
7DE7 3EC3       00136         LD      A,0C3H
7DE9 32C441     00138         LD      (41C4H),A
7DEC ED5BC541   00140         LD      DE,(41C5H)
7DF0 22C541     00150         LD      (41C5H),HL
7DF3 ED536A7F   00160         LD      (VECTR1),DE
7DF7 C3CC06     00170         JP      6CCH
                00180 ;THIS PROGRAM ALLOWS THE USER TO DRAW GRAPHIC LINES
                00190 ;BETWEEN TWO POINTS ON THE SCREEN POINTED TO BY X1,Y1 AND
                00200 ;X2,Y2 THROUGH THE USE OF A NEW COMMAND.
                00210 ;THE SYNTAX OF THIS COMMAND IS : SET<VAR1,VAR2,VAR3,VAR4>
                00220 ;OR RESET<VAR1,VAR2,VAR3,VAR4>.
                00230 ;WHERE VAR IS ANY INTEGER VARIABLE ALLOWED UNDER BASIC IT
                00240 ;IS PREFFERED TO USE A DEFINT STATEMENT FOR ANY VARIABLES
                00250 ;USED BY THE NEW SET/RESET COMMAND.
                00260 ;OTHER THINGS TO BE CONSIDERED ARE:
                00270 ;(1)-PUT THIS COMMAND IN A LINE BY ITSELF AND NOT IN A
                00280 :MULTI-STATEMENT LINE.
                00290 ;(2)-DO NOT PUT ANY BLANKS BEFORE OR IN THIS LINE.
                00300 ;(3)-VAR1,VAR2 MUST BE ONE X,Y PAIR AND VAR3,VAR4 THE OTHER
                00310 ;(4)-THIS COMMAND DOES NOT WORK AS A DIRECT STATEMENT.
                00320 ;(5)-IF THE ABOVE RULES ARE NOT ADHERED TO A SN ERROR WILL
                00330 ;BE GENERATED.
                00340 ;TO USE THIS PROGRAM FROM DISC TYPE "LOAD SET2" THEN
                00350 ;"BASIC" ANSWER THE MEM SIZE ? WITH :32222 (16K) AND THEN
                00360 ;USE THE SYSTEM COMMAND TO INITIALIZE "SET2" BY ANSWERING
                00370 ;THE " *? " WITH " /32222.
                00380 ;HAPPY DRAWING !
7DFA ED73EA7F   00390 START   LD      (BUF+9),SP      ;SAVE SP
7DFE 31D47D     00400         LD      SP,INIT-10      ;SET NEW SP
7E01 F5         00410         PUSH    AF
7E02 7E         00420         LD      A,(HL)
7E03 B7         00430         OR      A
7E04 C2647F     00440         JP      NZ,RETRN        ;A=0 IF PROGRAM IS RUNNING
7E07 C5         00450         PUSH    BC
7E08 D5         00460         PUSH    DE
7E09 E5         00470 LP4     PUSH    HL              ;SAVE REG'S
7E0A 23         00480         INC     HL
7E0B 23         00490         INC     HL              ;BYPASS LNUM & LPOINT
7E0C 23         00500         INC     HL
7E0D 23         00510         INC     HL
7E0E 23         00520         INC     HL
7E0F 7E         00530         LD      A,(HL)
7E10 FE83       00540         CP      83H             ;IS IT "SET" ?
7E12 2814       00550         JR      Z,SETPR         ;IF IT IS GOTO SETPR
7E14 FE82       00560         CP      82H             ;IS IT "RESET" ?
7E16 2803       00570         JR      Z,RSETPR        ;IF IT IS GOTO RSETPR
7E18 C3617F     00580         JP      RETN            ;IF NOT RETURN TO BASIC
7E1B 23         00590 RSETPR  INC     HL
7E1C 7E         00600         LD      A,(HL)          ;GET NEXT BYTE
7E1D FED6       00610         CP      0D6H            ;IS IT '<' ?
7E1F C2617F     00620         JP      NZ,RETN         ;IF NOT RETURN TO BASIC
```

```
7E22 AF         00630          XOR     A
7E23 32E57F     00640          LD      (BUF+8),A       ;"SET/RESET" FLAG = OFF
7E26 180C       00650          JR      BP4
7E28 23         00660 SETPR    INC     HL
7E29 7E         00670          LD      A,(HL)
7E2A FED5       00680          CP      0D5H
7E2C C2617F     00690          JP      NZ,RETN         ;SAME AS ABOVE
7E2F 3E80       00700          LD      A,128
7E31 32E57F     00710          LD      (BUF+8),A       ;"SET/RESET" FLAG = ON
7E34 23         00720 BP4      INC     HL
7E35 CD0D26     00730          CALL    260DH           ;GET X1
7E38 23         00740          INC     HL              ;ADVANCE PAST COMMA
7E39 1A         00750          LD      A,(DE)          ;PUT X1 IN A REG.
7E3A 32E27F     00760          LD      (BUF+1),A       ;LOAD X1 INTO BUFFER
7E3D CD0D26     00770          CALL    260DH           ;GET Y1
7E40 23         00780          INC     HL
7E41 1A         00790          LD      A,(DE)
7E42 32E47F     00800          LD      (BUF+3),A
7E45 CD0D26     00810          CALL    260DH           ;GET X2
7E48 23         00820          INC     HL
7E49 1A         00830          LD      A,(DE)
7E4A 32ED7F     00840          LD      (BUF+12),A
7E4D CD0D26     00850          CALL    260DH           ;GET Y2
7E50 1A         00860          LD      A,(DE)
7E51 32EF7F     00870          LD      (BUF+14),A
7E54 7E         00880          LD      A,(HL)
7E55 FED4       00890          CP      0D4H            ;IS LAST BYTE '>'
7E57 C2617F     00900          JP      NZ,RETN         ;IF NOT PRINT SN ERROR
7E5A 23         00910 LP3      INC     HL
7E5B 7E         00920          LD      A,(HL)
7E5C B7         00930          OR      A
7E5D 20FB       00940          JR      NZ,LP3          ;FIND END OF LINE
7E5F F1         00950          POP     AF              ;REMOVE OLD HL FROM STACK
7E60 E5         00960          PUSH    HL              ;PUSH NEW HL ON STACK
7E61 3E00       00970          LD      A,0
7E63 32E17F     00980          LD      (BUF),A
7E66 32E37F     00990          LD      (BUF+2),A       ;CLEAR BYTES IN BUFFER
7E69 32EC7F     01000          LD      (BUF+11),A
7E6C 32EE7F     01010          LD      (BUF+13),A
7E6F 2AE17F     01020          LD      HL,(BUF)        ;GET X1
7E72 ED5BEC7F   01030          LD      DE,(BUF+11)     ;GET X2
7E76 B7         01040          OR      A               ;RESET CY FLAG
7E77 ED52       01050          SBC     HL,DE
7E79 3818       01060          JR      C,CNT0          ;IF X1<X2 GOTO CNT0
7E7B 2AE17F     01070          LD      HL,(BUF)        ;EXCHANGE X1,Y1
7E7E ED53E17F   01080          LD      (BUF),DE        ;WITH X2,Y2
7E82 22EC7F     01090          LD      (BUF+11),HL     ;SOTHAT X1<X2
7E85 2AE37F     01100          LD      HL,(BUF+2)
7E88 ED5BEE7F   01110          LD      DE,(BUF+13)
7E8C 22EE7F     01120          LD      (BUF+13),HL
7E8F ED53E37F   01130          LD      (BUF+2),DE
7E93 2AEC7F     01140 CNT0     LD      HL,(BUF+11)
7E96 ED5BE17F   01150          LD      DE,(BUF)
7E9A B7         01160          OR      A
7E9B ED52       01170          SBC     HL,DE           ;DX=X2-X1
7E9D 23         01180          INC     HL              ;CORRECT DX
7E9E E5         01190          PUSH    HL              ;STORE DX
7E9F 2AE37F     01200          LD      HL,(BUF+2)      ;GET Y1
```

```
7EA2 ED5BEE7F 01210          LD      DE,(BUF+13)     ;GET Y2
7EA6 B7       01220          OR      A
7EA7 ED52     01230          SBC     HL,DE           ;DY=ABS(Y2-Y1)
7EA9 3005     01240          JR      NC,CNT2
7EAB 19       01250          ADD     HL,DE
7EAC EB       01260          EX      DE,HL
7EAD B7       01270          OR      A
7EAE ED52     01280          SBC     HL,DE
7EB0 23       01290 CNT2     INC     HL              ;CORRECT DY
7EB1 D1       01300          POP     DE              ;GET DX
7EB2 B7       01310          OR      A
7EB3 ED52     01320          SBC     HL,DE
7EB5 3819     01330          JR      C,CNT3          ;IF DX>DY GOTOCNT3
7EB7 19       01340          ADD     HL,DE           ;RESTORE DY IN HL
7EB9 3AE97F   01350          LD      A,(BUF+8)
7EBC CBB7     01360          RES     6,A
7EBE 32E97F   01370          LD      (BUF+8),A       ;RESET X/Y FLAG (Y COUNT)
7EC0 22E57F   01380          LD      (BUF+4),HL      ;LOAD DY IN BUFFER
7EC3 6C       01390          LD      L,H
7EC4 2600     01400          LD      H,0             ;DEFSNG DY
7EC6 EB       01410          EX      DE,HL           ;PREP. FOR DIVISION
7EC7 CD6C7F   01420          CALL    DIV             ;CALCULATE DX
7ECA ED43E77F 01430          LD      (BUF+6),BC      ;LOAD DX IN BUFFER
7ECE 1817     01440          JR      CNT4
7ED0 19       01450 CNT3     ADD     HL,DE
7ED1 3AE97F   01460          LD      A,(BUF+8)       ;GET FLAG BYTE
7ED4 CBF7     01470          SET     6,A             ;SET X/Y FLAG (X COUNT)
7ED6 32E97F   01480          LD      (BUF+8),A
7ED9 ED53E77F 01490          LD      (BUF+6),DE      ;LOAD DX IN BUFFER
7EDD 5A       01500          LD      E,D
7EDE 1600     01510          LD      D,0             ;DEFSNG DX
7EE0 CD6C7F   01520          CALL    DIV
7EE3 ED43E57F 01530          LD      (BUF+4),BC      ;LOAD DY IN BUFFER
7EE7 3AE97F   01540 CNT4     LD      A,(BUF+8)
7EEA CB77     01550          BIT     6,A
7EEC 281D     01560          JR      Z,CNT5          ;BYPASS IF COUNT=Y (DY>DX)
7EEE 2AE37F   01570          LD      HL,(BUF+2)      ;GET Y1
7EF1 ED5BEE7F 01580          LD      DE,(BUF+13)     ;GET Y2
7EF5 B7       01590          OR      A
7EF6 ED52     01600          SBC     HL,DE
7EF8 3811     01610          JR      C,CNT5
7EFA 280F     01620          JR      Z,CNT5
7EFC 3AE57F   01630          LD      A,(BUF+4)
7EFF ED44     01640          NEG
7F01 32E57F   01650          LD      (BUF+4),A       ;2'S COMP. DY
7F04 3AE67F   01660          LD      A,(BUF+5)       ;AS Y1>Y2 AND
7F07 2F       01670          CPL                     ;Y MUST TRAVEL
7F08 32E67F   01680          LD      (BUF+5),A       ;IN NEGATIVE DIRECTION
7F0B 3AE97F   01690 CNT5     LD      A,(BUF+8)
7F0E CB77     01700          BIT     6,A
7F10 2022     01710          JR      NZ,XSTEP
7F12 ED4BE57F 01720          LD      BC,(BUF+4)      ;SET UP COUNT
7F16 2AE37F   01730          LD      HL,(BUF+2)      ;GET Y1
7F19 ED5BEE7F 01740          LD      DE,(BUF+13)     ;GET Y2
7F1D B7       01750          OR      A
7F1E ED52     01760          SBC     HL,DE
7F20 3009     01770          JR      NC,CNT6         ;IF Y2<Y1, Y=NEG. GOING
7F22 110001   01780          LD      DE,100H         ;Y STEP =1
```

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542


```
7F25 ED53E57F 01790          LD      (BUF+4),DE
7F29 1814     01800          JR      CNT1
7F2B 1100FF   01810 CNT6     LD      DE,0FF00H       ;Y STEP = -1
7F2E ED53E57F 01820          LD      (BUF+4),DE      ;COMES HERE IF DY>DX AND
7F32 180B     01830          JR      CNT1            ;Y1>Y2 (NEG. GOING Y)
7F34 ED4BE77F 01840 XSTEP    LD      BC,(BUF+6)      ;SET UP COUNT
7F38 110001   01850          LD      DE,100H         ;X STEP =1
7F3B ED53E77F 01850          LD      (BUF+6),DE
7F3F 04       01900 CNT1     INC     B
7F40 D9       01910 CNT1A    EXX
7F41 CD7C7F   01920          CALL    SET
7F44 D9       01930          EXX
7F45 2AE17F   01940          LD      HL,(BUF)        ;GET X
7F48 ED5BE77F 01950          LD      DE,(BUF+6)      ;GET DX
7F4C 19       01960          ADD     HL,DE           ;NEXT X
7F4D 22E17F   01970          LD      (BUF),HL
7F50 2AE37F   01980          LD      HL,(BUF+2)      ;GET Y
7F53 ED5BE57F 01990          LD      DE,(BUF+4)      ;GET DY
7F57 19       02000          ADD     HL,DE           ;NEXT Y
7F58 22E37F   02010          LD      (BUF+2),HL
7F5B 10E3     02020          DJNZ    CNT1A           ;LOOP IF NOT DONE
7F5D E1       02030          POP     HL
7F5E C3097E   02040          JP      LP4             ;CHECK NEXT LINE BEFORE
              02050                                  ;RETURNING
7F61 E1       02060 RETN     POP     HL
7F62 D1       02070          POP     DE
7F63 C1       02080          POP     BC
7F64 F1       02090 RETRN    POP     AF              ;RESTORE REG'S
7F65 ED7BEA7F 02100          LD      SP,(BUF+9)      ;RESTORE SP
7F69 00       02110 VECTR0   DEFB    0               ;CARRY ON WITH BASIC PROG.
7F6A 0000     02120 VECTR1   DEFW    0               ;TO BE FILLED WITH BASIC
              02130                                  ;RETURN ADDRESS
7F6C 010100   02140 DIV      LD      BC,1
7F6F 7A       02150          LD      A,D
7F70 B3       02160          OR      E
7F71 C8       02170          RET     Z
7F72 010000   02180          LD      BC,0            ;HL = VALUE
7F75 B7       02190 LP2      OR      A
7F76 ED52     02200          SBC     HL,DE           ;DE = DIVISOR
7F78 D8       02210          RET     C               ;DONE
7F79 03       02220          INC     BC              ;BC = QUOTIENT
7F7A 18F9     02230          JR      LP2             ;LOOP UNTILL DONE
              02240 ;THE FOLLOWING SUBROUTINE DOES THE SAME AS A SET(X,Y)
              02250 ;COMMAND DOES UNDER BASIC
7F7C 3AE47F   02260 SET      LD      A,(BUF+3)       ;GET Y
7F7F 0600     02270          LD      B,0             ;RESET COUNTER
7F81 0E03     02280          LD      C,3             ;C=DIVISOR
7F83 CDDC7F   02290          CALL    SUBTR
7F86 B1       02300          ADD     A,C             ;GET REMAINDER
7F87 57       02310          LD      D,A             ;D=Y REMAINDER
7F88 21003C   02320          LD      HL,3C00H        ;SET HL TO START OF VID. MEM

7F8B 78       02330          LD      A,B
7F8C B7       02340          OR      A
7F8D 2808     02350          JR      Z,BP0
7F8F D5       02360          PUSH    DE
7F90 114000   02370          LD      DE,64           ;64 BYTES PER LINE
7F93 19       02380 LP0      ADD     HL,DE           ;64 * HL
```

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542


```
7F94 10FD      02390          DJNZ    LP0
7F96 D1        02400          POP     DE
7F97 0600      02410 BP0      LD      B,0
7F99 0E02      02420          LD      C,2
7F9B 3AE27F    02430          LD      A,(BUF+1)
7F9E CDDC7F    02440          CALL    SUBTR
7FA1 81        02450          ADD     A,C
7FA2 5F        02460          LD      E,A             ;E=X REMAINDER
7FA3 48        02470          LD      C,B             ;PREPARE FOR -
7FA4 0600      02480          LD      B,0             ;16 BIT ADD
7FA6 09        02490          ADD     HL,BC           ;HL POINTS TO VIDEO BYTE
7FA7 3E01      02500          LD      A,1
7FA9 42        02510          LD      B,D             ;GET Y REMAINDER
7FAA 04        02520          INC     B               ;SET FLAGS
7FAB 05        02530          DEC     B
7FAC 2806      02540          JR      Z,BP1           ;SKIP IF ZERO
7FAE CB27      02550 LP1      SLA     A
7FB0 CB27      02560          SLA     A
7FB2 10FA      02570          DJNZ    LP1
7FB4 1C        02580 BP1      INC     E
7FB5 1D        02590          DEC     E
7FB6 2802      02600          JR      Z,BP2           ;SKIP IF X REMAINDER = 0
7FB8 CB27      02610          SLA     A
7FBA F680      02620 BP2      OR      80H
7FBC 4F        02630          LD      C,A             ;STORE GRAPHICS BYTE
7FBD 7C        02640          LD      A,H             ;MAKE SURE THAT HL WILL
7FBE F63C      02650          OR      3CH             ;ALWAYS STAY WITHIN
7FC0 E63F      02660          AND     3FH             ;VIDEO MEMORY
7FC2 67        02670          LD      H,A
7FC3 7E        02680          LD      A,(HL)
7FC4 FE80      02690          CP      80H
7FC6 3002      02700          JR      NC,BP3
7FC8 3680      02710          LD      (HL),80H
7FCA 3AE97F    02720 BP3      LD      A,(BUF+8)
7FCD CB7F      02730          BIT     7,A
7FCF 2804      02740          JR      Z,RESET
7FD1 79        02750          LD      A,C
7FD2 B6        02760          OR      (HL)
7FD3 77        02770          LD      (HL),A
7FD4 C9        02780          RET
7FD5 79        02790 RESET    LD      A,C
7FD6 2F        02800          CPL
7FD7 A6        02810          AND     (HL)
7FD8 F680      02820          OR      80H
7FDA 77        02830          LD      (HL),A
7FDB C9        02840          RET
7FDC 91        02850 SUBTR    SUB     C
7FDD D8        02860          RET     C
7FDE 04        02870          INC     B
7FDF 18FB      02880          JR      SUBTR
000F           02890 BUF      DEFS    15
7DDE           02900          END     INIT
00000 TOTAL ERRORS
```

MICRO-80

***** BMON - PART 1 *****

```
75C4:  0D 48 45 58 2E 20 41 44 44 52 45 53 53 20 3F 20
75D4:  00 CD C9 01 21 C4 75 CD 76 72 CD 13 7A 3E 0D D5
75E4:  CD 33 00 7C E6 F0 CD 4F 77 7C E6 0F CD 4F 77 7D
75F4:  E6 F0 CD 4F 77 7D E6 0F CD 4F 77 3E 20 CD 33 00
7604:  3E 3C CD 33 00 7E E6 F0 CD 4F 77 7E E6 0F CD 4F
7614:  77 3E 3E CD 33 00 3E 20 CD 33 00 D1 06 02 11 3D
7624:  72 D5 CD 49 00 D1 FE 01 CA 63 72 FE 08 28 AE FE
7634:  0A 20 03 23 18 A7 FE 30 38 E7 FE 58 28 96 FE 5B
7644:  20 03 2B 18 98 FE 47 30 D8 D5 CD 33 00 D1 12 13
7654:  10 CF 11 3D 72 1A 13 D6 30 FE 10 38 02 D6 07 CD
7664:  9A 76 4F 1A D6 30 FE 10 38 02 D6 07 81 77 23 C3
7674:  E1 75 21 C4 75 CD 76 72 06 04 3E 20 CD 33 00 21
7684:  E6 41 CD F9 79 06 02 21 E6 41 56 23 5E 7A 18 13
7694:  FE 3A 30 0B D6 30 CB 27 CB 27 CB 27 CB 27 C9 D6
76A4:  37 18 F3 CD 94 76 57 7B FE 3A 30 0D D6 30 82 57
76B4:  23 D5 10 D6 E1 6C D1 62 E9 D6 37 18 F1

774D:  C4 76 FE 10 D4 68 77 FE 0A 30 08 C6 30 D5 CD 33
775D:  00 D1 C9 C6 37 D5 CD 33 00 D1 C9 CB 3F CB 3F CB
776D:  3F CB 3F C9 2A

797S:  AC 78 53 54 41 52 54 20 45 4E 44 20 20 45 4E 54
7989:  52 59 20 4E 41 4D 45 20 3F 0D 00 54 59 50 45 20
7999:  3C 20 42 52 45 41 4B 20 3E 20 54 4F 20 53 54 4F
79A9:  50 20 44 55 52 49 4E 47 20 43 41 53 53 45 54 54
79B9:  45 20 49 2F 4F 2E 0D 00 CD C9 01 21 7B 79 CD 76
79C9:  72 18 57 CD 49 00 FE 01 CA 63 72 FE 08 28 0E FE
79D9:  30 38 F0 FE 47 30 EC CD 33 00 77 23 C9 5F 78 FE
79E9:  04 7B 30 DF 2B 04 CD 33 00 18 D8 06 04 21 32 72
79F9:  CD CC 79 10 FB C9 7E 23 CD 94 76 4F 7E 23 FE 3A
7A09:  30 04 D6 30 18 02 D6 37 81 C9 CD F4 79 21 32 72
7A19:  CD FF 79 57 CD FF 79 5F EB C9 CD 13 7A 22 3F 72
7A29:  3E 20 CD 33 00 CD 33 00 CD 13 7A 22 41 72 3E 20
7A39:  CD 33 00 CD 13 7A 22 43 72 3E 20 CD 33 00 CD 33
7A49:  00 06 06 21 45 72 CD 49 00 FE 01 CA 63 72 FE 0D
7A59:  28 0D FE 08 28 32 77 23 CD 33 00 10 E9 18 06 3E
7A69:  20 77 23 10 FC 3E 0D CD 33 00 21 94 79 CD 76 72
7A79:  2A 41 72 ED 5B 3F 72 B7 ED 52 23 11 80 00 06 00
7A89:  B7 ED 52 38 11 04 18 F8 5F 78 FE 06 7B 30 B7 CD
7A99:  33 00 2B 04 18 B0 19 7D 32 26 72 21 45 72 AF C5
7AA9:  CD 12 02 CD 87 02 C1 3E 55 CD 64 02 C5 06 06 7E
7AB9:  23 CD 64 02 10 F9 2A 3F 72 C1 78 B7 28 3B CD 5B
7AC9:  72 3E 3C CD 64 02 7B CD 64 02 7D 57 CD 64 02 7C
7AD9:  CD 64 02 82 57 3A 40 38 E6 04 20 2A 7E 23 CD 64
7AE9:  02 1D 20 EF 82 CD 64 02 1E 80 16 00 10 D0 E5 ED
7AF9:  5B 41 72 B7 ED 52 30 0E E1 16 00 3A 26 72 06 01
7B09:  B7 28 03 5F 18 B8 2A 43 72 3E 78 CD 64 02 7D CD
7B19:  64 02 7C CD 64 02 CD F8 01 C3 63 72 42 41 53 49
7B29:  43 20 4D 4F 4E 49 54 4F 52 20 42 59 20 45 2E 52
7B39:  2E 50 41 41 59 20 28 43 29 20 31 39 37 39 20 28
7B49:  48 49 54 20 3C 53 48 49 46 54 3E 20 5C 20 54 4F
7B59:  20 41 43 43 45 53 53 2E 29 42 41 53 49 43 20 50
```

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542


```
7B69:  52 4F 47 52 41 4D 20 4C 4F 43 41 54 49 4F 4E 53
7B79:  20 3A 20 53 54 41 52 54 20 3D 20 58 58 58 58 48
7B89:  20 2C 20 45 4E 44 20 3D 20 58 58 58 58 48 20 20
7B99:  CD 9F 7B C3 CC 06 CD C9 01 21 40 3C 22 20 40 11
7BA9:  00 3C 21 25 7B 01 3D 00 ED B0 21 C5 7B 3E C3 32
7BB9:  33 40 22 34 40 AF 32 15 40 21 99 7B C9 CD E3 03
7BC9:  B7 C8 FE 1A C2 85 7D ED 73 24 72 31 22 72 E5 21
7BD9:  00 00 22 28 72 D5 C5 F5 DD E5 FD E5 C3 63 72


7D58:  13 C9 CD C3 01 21 15 40 36 00 FD E1 DD E1 F1 C1
7D68:  D1 E1 ED 7B 24 72 C3 CC 06


7D83:  F4 1F C5 01 0F 0D 10 FE 0D 20 FB C1 C9 42 41 53
7D93:  49 43 20 4D 4F 4E 49 54 4F 52 20 56 45 52 2E 20
7DA3:  31 2E 32 20 20 43 4F 4D 4D 41 4E 44 20 20 4C 49
7DB3:  53 54 20 3A 0D 2D 2D 2D 20 53 45 4C 45 43 54 20
7DC3:  46 55 4E 43 54 49 4F 4E 20 21 21 21 20 2D 2D 2D
7DD3:  0D 42 20 2D 20 52 45 54 55 52 4E 20 54 4F 20 42
7DE3:  41 53 49 43 0D 52 20 2D 20 52 45 4E 55 4D 42 45
7DF3:  52 0D 4C 20 2D 20 4C 4F 41 44 20 42 41 53 49 43
7E03:  20 50 52 4F 47 52 41 4D 0D 4D 20 2D 20 4D 45 52
7E13:  47 45 20 42 41 53 49 43 20 50 52 4F 47 52 41 4D
7E23:  0D 50 20 2D 20 50 52 4F 54 45 43 54 20 50 52 4F
7E33:  47 52 41 4D 0D 56 20 2D 20 4C 49 53 54 20 56 41
7E43:  52 49 41 42 4C 45 53 20 55 53 45 44 20 49 4E 20
7E53:  50 52 4F 47 52 41 4D 0D 53 20 2D 20 52 45 53 54
7E63:  4F 52 45 20 50 52 4F 47 52 41 4D 0D 47 20 2D 20
7E73:  47 4F 54 4F 20 48 45 58 20 4C 4F 43 41 54 49 4F
7E83:  4E 0D 44 20 2D 20 44 45 43 2E 20 54 4F 20 48 45
7E93:  58 2E 20 43 4F 4E 56 45 52 53 49 4F 4E 0D 48 20
7EA3:  2D 20 48 45 58 2E 20 54 4F 20 44 45 43 2E 20 43
7EB3:  4F 4E 56 45 52 53 49 4F 4E 0D 46 20 2D 20 43 41
7EC3:  4E 43 45 4C 20 50 52 4F 54 45 43 54 49 4F 4E 0D
7ED3:  45 20 2D 20 45 44 49 54 20 4D 45 4D 4F 52 59 0D
7EE3:  43 20 2D 20 43 4F 50 59 20 4D 45 4D 4F 52 59 20
7EF3:  54 4F
```

```
7345    C3  63  72
73F8    C3  63  72
78F8    C3  63  72
7588    C3  63  72
752A    C3  63  72
789A    C3  63  72
75E7    C3  63  72
7601    C3  63  72
7816    C3  63  72
7546    C3  63  72
```

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542


```
100 ' HEX MONITOR IN BASIC - FULLY DEBUGGED!
110 ' AUTHOR PETER G. HARTLEY
120 ' COPYRIGHT (C) 1980, PETER G. HARTLEY AND "MICRO-80"
130 CLEAR550:DEFINTA-X:DEFDBLS:DEFSTRZ:POKE16553,255:ONERRORGOTO500
140 CLS:PRINT"                    M E N U
                  -------

X - EXAMINE MEMORY BLOCK

E - EDIT     MEMORY BLOCK

P - PUNCH    MEMORY BLOCK TO TAPE

L - LOAD     MEMORY BLOCK FROM TAPE


PLEASE SELECT OPTION";
150 GOSUB480  :IFZD$="X"THENGOSUB220  ELSEIFZD$="E"THENGOSUB260  ELSEIFZD$="P"TH
ENGOSUB300  ELSEIFZD$="L"GOSUB340
160 GOTO140
170 DATA0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
180 RESTORE:V=0
190 READZR$:IFZR$<>ZV$ THEN V=V+1:GOTO190  :ELSERETURN
200 RESTORE:FORKV=0TOV:READZV$:NEXT:IFTAPE=77THENZZ$=ZZ$+ZV$:KC=KC+1:RETURN:ELSE
RETURN
210 CLS:PRINT"STARTING ADDRESS";:GOSUB450  :INPUTZS$:GOSUB460  :RETURN
220 GOSUB210  :CLS:PRINT"AUTOMATIC HEX LISTING OF MEMORY FROM";S;"
TO HALT LISTING PRESS >> @ <<
TO RESTART      PRESS >> @ <<
TO ABORT        PRESS >ENTER<

";
230 GOSUB400  :GOSUB470  :IFZD$="@"THENGOSUB250  ELSEIFZD$=CHR$(13)THENRETURN
240 GOTO230
250 ZD$="":GOSUB470  :IFZD$="@"THEN230  ELSE250
260 GOSUB210  :CLS:PRINT"MEMORY EDITING FROM ADDRESS";S;"
TO END EDITING ENTER A  >> @ <<
TO LEAVE LOCATION UNCHANGED HIT >> ENTER <<
TO ENTER NEW STARTING ADDRESS ENTER A  >> N <<

";
270 GOSUB380  :GOSUB410  :ZV$="":INPUTZV$:IFZV$="@"THENRETURNELSEIFZV$=""THENS=S
+1:GOTO270  :ELSEIFZV$="N"THENGOTO260
280 L=LEN(ZV$):IFL<2 THEN 270  ELSE IF L>2 THEN ZV$=RIGHT$(ZV$,2)
290 ZZ$=ZV$:ZV$=LEFT$(ZZ$,1):GOSUB180  :A=V:ZV$=RIGHT$(ZZ$,1):GOSUB180  :A=A*16+
V:POKES,A:S=S+1:GOTO270
300 CLS:INPUT"PROGRAM NAME";ZN$:GOSUB210  :SS=S:CLS:PRINT"MEMORY DUMP TO TAPE OF
 ";ZN$;" FROM ADDRESS";S"
TO ";:PRINT"FINISHING ADDRESS";:GOSUB450  :INPUTZS$:GOSUB460
310 SF=S:CLS:PRINT"MEMORY DUMP TO TAPE OF ";ZN$;" FROM";SS;"TO
";SF;:PRINT"ENTRY ADDRESS";:INPUTZS$:GOSUB460
```

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542


```
320 CLS:PRINT"TAPE DUMP OF ";ZN$;"

FROM ";SS;"

TO   ";SF;"

     ";S;"ENTRY POINT

";:INPUT"HIT >> ENTER << WHEN CASSETTE READY TO RECORD";ZC$:ST=S:S=SS:PRINT#-1,S
S,SF,ST,ZN$:CLS:PRINT"TAPE DUMP IN PROGRESS":PRINT:ZZ$="":KC=0:KK=0:ZN$="":X=FRE
(0)
330 TAPE=0:GOSUB380 :TAPE=77:FORKL=0TO15:GOSUB410 :S=S+1:IFKC=248 OR S=SF+1 TH
ENGOSUB430 :NEXT:GOTO330 :ELSENEXT:GOTO330
340 CLS:INPUT"IF YOU HAVE NOT PROTECTED HIGH MEMORY, THIS WILL BE A WASTE OF
EVERYONE'S TIME. HIT >> ENTER << WHEN CASSETTE READY TO PLAY";ZZ$:Z$="":INPUT#-
1,SS,SF,ST,ZN$:POKE16553,255:CLS:PRINT"STARTING ADDRESS";SS.:S=SS:GOSUB380 :PRI
NT
350 PRINT"FINISHING ADDRESS";SF,:S=SF:GOSUB380 :PRINT:PRINT"SYSTEM ENTRY ADDRES
S /";ST,:S=ST:GOSUB380 :PRINT:PRINT"PROGRAM NAME ";ZN$:ZN$="":S=SS:KC=16
360 INPUT#-1,ZZ$:POKE16553,255:L=LEN(ZZ$):FORKZ=1TOL STEP2:IFKC=16GOSUB380 :KC=
0
370 GOSUB420 :ZV$=MID$(ZZ$,KZ,1):PRINTZV$;:GOSUB180 :DD=V:ZV$=MID$(ZZ$,KZ+1,1)
:PRINTZV$;" ";:GOSUB180 :DD=DD*16+V:POKESX,DD:S=S+1:KC=KC+1:IFS<>SF+1 THENNEXTK
Z:GOTO360 :ELSEGOTO440
380 GOSUB420 :A1=INT(SD/4096):SD=SD-(A1*4096):A2=INT(SD/256):SD=SD-(A2*256):A3=
INT(SD/16):A4=SD-A3*16
390 KL=0:V=A1:GOSUB200 :PRINTZV$;:V=A2:GOSUB200 :PRINTZV$;:V=A3:GOSUB200 :PRI
NTZV$;:V=A4:GOSUB200 :PRINTZV$,:RETURN
400 GOSUB380 :FORKL=0TO15:GOSUB410 :S=S+1:NEXT:RETURN
410 GOSUB420 :VV=PEEK(SX):V=INT(VV/16):VV=VV-V*16:GOSUB200 :PRINTZV$;:V=VV:GOS
UB200 :PRINTZV$;" ";:RETURN
420 IFS<32768THENSX=S:SD=S:RETURN:ELSESX=(S-32767)*-1:SD=32767-SX:RETURN
430 IFTAPE<>77THENRETURNELSEPRINT#-1,ZZ$:ZZ$="":KC=0:IFS=SF+1 THENGOTO440 ELSER
ETURN
440 PRINT:PRINT"TAPE DUMP COMPLETED - HIT >> ENTER << FOR MENU":GOSUB480 :RUN
450 PRINT"

        ####      DECIMAL

        ####H     HEX

";:RETURN
460 ZH$=RIGHT$(ZS$,1):L=LEN(ZS$):IFZH$<>"H"THENS=VAL(ZS$):RETURN:ELSES=0:FORKL=1
TOL-1:ZV$=MID$(ZS$,KL,1):GOSUB180 :S=S*16+V:NEXTKL:RETURN
470 ZD$=INKEY$:RETURN
480 ZD$=INKEY$:ZD$=""
490 ZD$=INKEY$:IFZD$=""  THEN 490  ELSERETURN
500 IFERR/2+1=4THENCLS:PRINT"OUT OF DATA ERROR
PROBABLY CAUSED BY A TAPE-READING ERROR

STRING DATA READ FROM CASSETTE WAS

";ZZ$:PRINT"LAST BYTE ENTERED WAS";S-1;"

HIT >> ENTER << TO RESUME WITH BYTE";S;" FAULTY":INPUTZQ$:ZR$="X":ZV$="X":V=0:RE
SUME190
510 RESUME130
```

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542

***** MINI MACHINE LANGUAGE SAVE ROUTINE IN BASIC - L2/4K *****

```
20 'AUTHOR PETER G. HARTLEY
30 'COPYRIGHT PETER G. HARTLEY AND "MICRO-80"
100 CLS:CLEAR480:DEFINTA-Y:DEFSTRZ:INPUT"STARTING ADDRESS
ENDING ADDRESS
ENTRY  ADDRESS
NAME";SS,SF,ST,ZN$
110 CLS:INPUT"HIT >> ENTER << WHEN CASSETTE READY TO RECORD";ZQ$:PRINT#-1,SS,SF,
ST,ZN$:ZN$="":KS=0:P1=192:P2=62
120 CLS:KC=16:PRINT@128,"PLEASE ENTER THE LISTING IN HEX

IF YOU MAKE A MISTAKE, DO NOT HIT ENTER,
JUST ADD EXTRA DIGITS UNTIL THE LAST TWO
ARE CORRECT - THESE ARE ALL THE PROGRAM
WILL RECOGNISE."
130 FORS=SSTOSF
140 IFKC=16 THEN KC=0:POKE16416,P1:POKE16417,P2:PRINTS.:P1=PEEK(16416):P2=PEEK(1
6417)
150 IFKS=124 THEN KS=0:PRINT@74,"T A P E   D U M P   I N   P R O G R E S S ";:PR
INT#-1,ZZ$:ZZ$="":PRINT@74,"                                                  ";:IFS=>S
F PRINT@74,"T A P E   D U M P   I S   C O M P L E T E D ":END
160 PRINT@0,"                                                                  ";:P
RINT@0,"";:INPUTZB$:L=LEN(ZB$):IFL<2GOTO160
170 ZB$=RIGHT$(ZB$,2):ZZ$=ZZ$+ZB$:ZB$=ZB$+" ":POKE16416,P1:POKE16417,P2:PRINTZB$
;:P1=PEEK(16416):P2=PEEK(16417):KC=KC+1:KS=KS+1:IFS=SFTHENKS=124:GOTO150ELSENEXT
S
```

***** MINI MACHINE LANGUAGE LOADER IN BASIC *****

```
10 CLEAR260:DEFINTA-X:DEFDBLS:DEFSTRZ:POKE16553,255::GOSUB60   :GOTO10
20 DATA0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
30 RESTORE:V=0
40 READZR$:IFZR$<>ZV$ THEN V=V+1:GOTO40   :ELSERETURN
50 RESTORE:FORKV=0TOV:READZV$:NEXT:RETURN
60 CLS:INPUT"IF YOU HAVE NOT PROTECTED HIGH MEMORY, THIS WILL BE A WASTE OF
EVERYONE'S TIME.  HIT >> ENTER << WHEN CASSETTE READY TO PLAY";ZZ$:Z$="":INPUT#-
1,SS,SF,ST,ZN$:POKE16553,255:CLS:PRINT"STARTING ADDRESS";SS,:S=SS:GOSUB100  :PRI
NT
70 PRINT"FINISHING ADDRESS";SF,:S=SF:GOSUB100  :PRINT:PRINT"SYSTEM ENTRY ADDRESS
 /";ST,:S=ST:GOSUB100  :PRINT:PRINT"PROGRAM NAME ";ZN$:ZN$="":S=SS:KC=16
80 INPUT#-1,ZZ$:POKE16553,255:L=LEN(ZZ$):FORKZ=1TOL STEP2:IFKC=16GOSUB100  :KC=0

90 GOSUB130  :ZV$=MID$(ZZ$,KZ,1):PRINTZV$;:GOSUB30   :DD=V:ZV$=MID$(ZZ$,KZ+1,1):
PRINTZV$;" ";:GOSUB30   :DD=DD*16+V:POKESX,DD:S=S+1:KC=KC+1:IFS<>SF+1THENNEXTKZ:
GOTO80   :ELSEGOTO140
100 GOSUB130  :A1=INT(SD/4096):SD=SD-(A1*4096):A2=INT(SD/256):SD=SD-(A2*256):A3=
INT(SD/16):A4=SD-A3*16
110 KL=0:V=A1:GOSUB50   :PRINTZV$;:V=A2:GOSUB50   :PRINTZV$;:V=A3:GOSUB50   :PRI
NTZV$;:V=A4:GOSUB50   :PRINTZV$,:RETURN
120 STOP:GOSUB100  :FORKL=0TO15:GOSUB      :S=S+1:NEXT:RETURN
130 IFS<32768THENSX=S:SD=S:RETURN:ELSESX=(S-32767)*-1:SD=32767-SX:RETURN
140 PRINT:PRINT"TAPE DUMP COMPLETED.":STOP
150 ZD$=INKEY$:RETURN
```

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034 AUSTRALIA
TELEPHONE (08) 381 8542

APPLICATION FORM

FOR PUBLICATION OF A PROGRAM

To: MICRO-80

Please consider the enclosed program for publication:

IN MICRO-80

ON CASSETTE OR DISK ONLY

☐ Tick one only

☐

NAME .....

ADDRESS.....

.....POST CODE .....

CHECK LIST

Make sure you have PRINTED the following information clearly on your cassette or disk:

YOUR NAME AND ADDRESS
PROGRAM NAME
MEMORY SIZE
LEVEL 1, LEVEL 2, BASIC 2.2, SYSTEM 1 or 2, EDTASM
HOW TO START PROGRAM (Eg. Enter/32ØØØ)

Make sure you include the following information with your cassette or disk:

EXPLANATION OF WHAT THE PROGRAM DOES AND HOW TO USE IT

DESCRIPTION OF HOW TO CHANGE IT FOR DIFFERENT MEMORY SIZES AND DIFFERENT LEVELS OF BASIC

THE SIZE OF MEMORY REQUIRED AND TYPE OF SYSTEM NEEDED TO RUN IT (Eg. LEVEL 1 4K, 32K WITH 2 DISK DRIVES ETC.)

START, END AND ENTRY POINT FOR MACHINE LANGUAGE PROGRAMS

ANY SIMPLE CHANGES WHICH CAN BE MADE TO MAKE IT MORE FLEXIBLE OR USEFUL

DON'T FORGET A STAMPED SELF-ADDRESSED ENVELOPE IF YOU WANT YOUR CASSETTE OR DISK RETURNED.