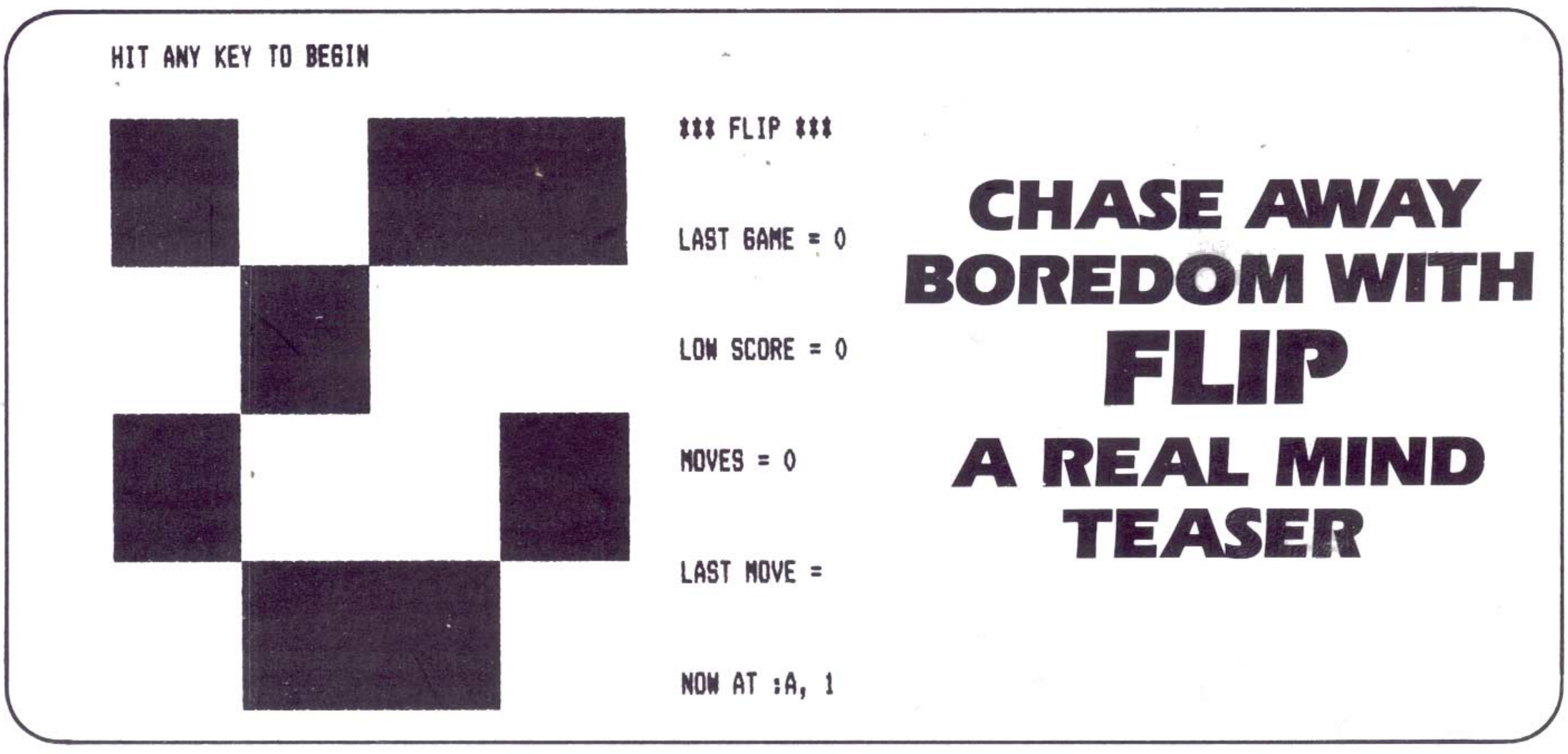
TRS-80 SYSTEM 80 VIDEO GENIE PMC-80

Vol. 3, Issue 5, April 1982

NOW CATERS FOR HITACHI PEACH AND TRS-80 COLOUR COMPUTER



CHASE AWAY BOREDOM WITH FLIP A REAL MIND TEASER

Also in this issue:

NEW RELEASES:

The Video Genie III and IV

PROGRAMMING:

Level I to Level II Program Conversion
Basic Basic
Basic Recursion

REVIEWS:

Some Educational Software
TRS-80 Colour Computer in Education

SOFTWARE:

- Space Commander — Level I
- Space Galaxy — Level I
- Atomic Tables Colour
- Keyword — Level II
- Read-a-Line — Level II
- Multiple Regression Analysis–Colour

MICRO-80

P.O. BOX 213, GOODWOOD, S.A. 5034. AUSTRALIA. TELEPHONE (08) 211 7244. PRICE: AUS. $2.50, N.Z. $4.00, U.K. £1.50

***** ABOUT MICRO-80 *****

EDITOR:	IAN VAGG
ASSOCIATE EDITORS:	
SOFTWARE LEVEL I :	MICHAEL SVENSDOTTER
SOFTWARE LEVEL II:	CHARLIE BARTLETT
HARDWARE :	EDWIN PAAY

MICRO-80 is an international magazine devoted entirely to the Tandy TRS-80 microcomputer and the Dick Smith System 80/Video Genie. It is available at the following prices:

	12 MONTH SUB.	SINGLE COPY
MAGAZINE ONLY	$ 26-00	$ 2-50
CASSETTE PLUS MAGAZINE	$ 65-00	$ 4-00 (cass. only)
DISK PLUS MAGAZINE	$ 125-00	$ 10-00 (disk only)

MICRO-80 is available in the United Kingdom from:
U.K. SUBSCRIPTION DEPT. 24 Woodhill Park, Pembury, Tunbridge Wells, KENT. TN2 4NW

Prices:			
	MAGAZINE ONLY	£ 16-00	£ 1-50
	CASSETTE PLUS MAGAZINE	£ 43-60	N|A
	DISK PLUS MAGAZINE	£ 75-00	N|A

MICRO-80 is available in New Zealand from:
MICRO PROCESSOR SERVICES, 940A Columbo Street, CHRISTCHURCH 1 N.Z. Ph. 62894

Prices:			
	MAGAZINE ONLY	NZ$ 43-00	NZ$ 4-00
	CASSETTE PLUS MAGAZINE	NZ$ 89-00	NZ$ 5-00
	DISK PLUS MAGAZINE	NZ$ 175-00	NZ$ 15-00

MICRO-80 is despatched from Australia by airmail to other countries at the following rates:

(12 MONTH SUB.)	MAGAZINE	CASS + MAG	DISK + MAG
PAPUA NEW GUINEA	Aus$ 40-00	Aus$ 83-00	Aus$ 143-00
HONG KONG/SINGAPORE	Aus$ 44-00	Aus$ 88-00	Aus$ 148-00
INDIA/JAPAN	Aus$ 49-00	Aus$ 95-00	Aus$ 155-00
USA/MIDDLE EAST/CANADA	Aus$ 55-00	Aus$ 102-00	Aus$ 162-00

Special bulk purchase rates are also available to computer shops etc. 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, System 80 or Video Genie and their peripherals. MICRO-80 is in no way connected with either the Tandy or Dick Smith organisations.

** WE WILL PAY YOU TO PUBLISH 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 or System 80 to earn some extra income is included in every issue.

** CONTENT **
Each month we publish at least one applications program in Level I BASIC, one in Level II BASIC and one in DISK BASIC (or disk compatible Level II). We also publish Utility programs in Level II BASIC and Machine Language. At least every second issue has an article on hardware modifications or a constructional article for a useful peripheral. In addition, we run articles on programming techniques both in Assembly Language and BASIC and we print letters to the Editor and new product reviews.

** LIABILITY **
The programs and other articles in MICRO-80 are published in good faith and we do our utmost to ensure that they function as described. However, no liability can be accepted for the failure of any program or other article to function satisfactorily or for any consequential damages arising from their use for any purpose whatsoever.

***** CONTENTS *****

MICRO-80 is registered by Australia Post - Publication SQB 2207 Category B

AUSTRALIAN OFFICE AND EDITOR: MICRO-80 P.O. BOX 213, GOODWOOD, SOUTH AUSTRALIA, 5034. TEL. (08) 211 7244

U.K. SUBSCRIPTION DEPT: 24 WOODHILL PARK, PEMBURY TUNBRIDGE WELLS, KENT TN2 4NW

Printed by: Shovel & Bull Printers, 379 South Road, MILE END 5031

Published in Australia by: MICRO-80, 433 Morphett Street, Adelaide.

***** EDITORIAL *****

In the first six months of 1982, 63 new microcomputers were introduced to Australia, each one claiming to be better than the others. No doubt those who had already made their decision and bought a computer in less confusing days are grateful that they do not have to try to make that decision today. The big question facing the prospective home computer buyer must be that of support. When the initial joy of ownership of the shiny new machine wears off, how will the new owner learn to make use of it? One of the unfortunate aspects of the plethora of computers now available is that the computer magazines are finding it more and more difficult to offer articles and programs for specific computers. To do so means that scarce editorial space is being given over to proportionately smaller special interest groups, whereas an increasingly larger group of readers is buying the magaines to assist them in making their decision of which computer to buy. Therefore, even the owners of 'traditional' computers find that information and support for their machines is dwindling too. It is an unfortunate situation which can only be cured either by an enormous increase in the sales of all microcomputers so that there are sufficient numbers of each to rate much larger magazines and/or more specialist magazines (unlikely) or there will be a shakeout such that many brands will disappear from sale, thus leaving a more manageable number of microcomputer models which can be satisfactorily supported (more likely). If, at this stage, you think we have ignored the most obvious avenue for support, the Australian distributors, let me assure you that we have not. With a few notable exceptions, most distributors of microcomputers do not have the geographic spread or the resources to provide the degree of support needed for the length of time required to ensure that their computers will be successful. The motor industry must have been very similar in the 1920's with hundreds of different manufacturers vying for trade - how many are left now?

Where does MICRO-80 stand in all this? We are essentially very different from other computer magazines. From the outset, we have concentrated on one model of computer and its derivatives or compatible alternatives. All our editorial material is aimed at that computer and the programs we publish are only of use in it. That computer is, of course, the TRS-80 Model I. In order to support a computer in the depth to which we do, we must invest considerable time, effort and funds but, above all, we must be interested in it. We will continue to support the '80 microcomputer until there is no more need to do so. From time to time, we will add other computers to the list we support. The major criterion in selecting these computers is that we are interested in them and that we can afford to make the very considerable investment needed to support them. As with the '80, when we first introduce a new computer, we will be learning too. It is in this spirit that we add two new computers to our magazine in this issue - the Hitachi "Peach" and the TRS-80 Colour Computer.

The Hitachi "Peach" is an interesting computer in its own right and of particular interest to Australasia, since this is the first (and so far the only) area outside Japan where the computer has been released. Hitachi is learning from its experiences here. Like all new computers, the Hitachi suffered from lack of software when first released. It is interesting to note that it took at least two years for the Apple and the TRS-80 to build up a significant library of software to the point where non-programmers would seriously consider buying one of these computers to do a specific task. The Hitachi "Peach" has achieved that in about six months. There is now available for the "Peach" high quality word processing, financial planning and accounting programs together with a small, but growing, number of games. Most of this software has been written in Australia. Delta, the distributors for the "Peach" and Hi Soft, recently appointed as the official Hitachi software publishers, are to be congratulated on a fine effort.

The TRS-80 Colour Computer represented something of a departure for Tandy. It was the first time this company moved away from using the Z80 microprocessor and was also its first acknowledgement of the value of colour and high resolution graphics. The TRS-80 Colour is a specialised computer in that it is primarily intended for home and educational use. It is also being supported with pre-packaged Tandy software much more than earlier machines. We have chosen to support the TRS-80 Colour Computer because we are interested in it, because we know many of our readers who already own Model I's are also interested in it and because its 6809 microprocessor and Microsoft colour BASIC interpreter make it reasonably compatible with the Hitachi "Peach". MICRO-80's style of support is a cooperative effort and we are eagerly awaiting receipt of articles and programs from our readers about the "Peach" and 80C (TRS-80 Colour Computer) for which we will, of course, pay publication fees. In the meantime, we have started our support modestly enough by converting two programs published in recent issues of MICRO-80, to run on the colour computers. We hope you enjoy them. We will also strive to keep you informed about developments in software in particular, for these machines.

For the present, the cassette and disk editions of MICRO-80 will NOT be available for the "Peach" or CC. All new subscribers automatically receive a copy of the Software Library. This is for TRS-80 Model I compatible computers only, so will be of little use to "Peach" and CC users. We do not have firm plans at present to produce a software library for the colour computers.

Australia is not the only country where new computers are being released. Our U.K. Correspondent Tony Edwards, has followed up his earlier scoop on the Genie III with more details of this interesting computer which is being released in the U.K. on July 22nd. We naturally enquired of the Dick Smith organisation about their plans. They have had a Genie III in their possession for some time now but have not decided whether or not to release it in Australia. We will keep you posted. The other new Video Genie reported by Tony, the Genie IV due for release in September

1982 will pose some problems for the Dick Smith organisation too, since it would be in direct competition with the VIC-20 colour computer. Now, over to Tony to tell you all about the two new Genies.

- 0000000000 -

***** PEEKing (UK) by our U.K. Correspondent Tony Edwards *****

In this department of issue 23 (October 1981) the new Genie III computer was introduced. It is now available in the U.K. as per specification and on schedule (which must be some sort of record for the computer industry). It has the type No. EG3200 and is available via the usual outlets. As reported it only has 2K of bootstrap ROM with its BASIC interpreter on disk for transfer into its 64K of dynamic RAM. It is available with up to 700K bytes of storage on 2 80-track, single sided disk drives operating in double density, which will soon be doubled with the introduction of double sided drives. A further two drives can be added externally.

The Genie III looks exactly as shown in the photograph on the front cover of issue 23 of MICRO-80 with a few minor cosmetic improvements. What is not clear is the disk operating system used. We understand that it operates under either CPM or TRSDOS/Disk BASIC but are unsure how screen handling and the keyboard are arranged to make the latter possible. We will give you more news in a future issue. In the meantime, its specification is shown below.

GENIE III is an upgraded version of the existing GENIE I & II systems. It is designed for more professional applications such as office data processing. The system has faster processing speed and more disk storage. The system is composed of three cards which are mounted in a five slot card cage. The design is aimed at convenient maintenance and easy expansion. Following is the system specification.

* CPU	-	Z80A, clock frequency is 3.2 MHz.
* RAM	-	64K bytes dynamic memory.
* ROM	-	2K bytes bootstrap and diagnostic routines.
* Disk Storage	-	5.25" double track double density disk drives. Four drives can be daisy chain together to provide system storage capacity up to 1.4 Mbytes. Double-sided double track drives are coming which can double the capacity.
* Parallel Port	-	One parallel port for driving Centronics compatible printer.
* Serial Port	-	One RS232C interface port for connection to serial printer or modemm.
* Keyboard	-	Detachable 85 Key keyboard with full ASCII keyset and a numeric keypad. European keyset option provided.
* Display	-	Non-glaring,12" green display.
* Display Format	-	80x24,or64x16 format software selectable upper/lower case characters with limited graphics (160x72) or inverse video characters
* Cassette	-	provide TRS-80 Model I compatible interface.
* Real-time clock	-	a CMOS real-time clock with battery backup provides the computer with true time and date information.
* Power Supply	-	a switching power supply is used to minimise weight and heat generation. Input power 190V-250V or 100V-130V. Output power is 12V - 4A, 5V - 5A, -15V - 1A.
* Operating System	-	CP/M version 2.2 or GDOS version 2,0.
* Options:-		
Programmable graphics character interface		it is a circuit adapted on the video interface card which enables the user to define his own character set such as some mathematical symbols.
High Resolution graphics interface	-	graphics resolution is 640x288. It is adapted on the video interface card, therefore, it can not coexist with the programmable graphic commands will be added in the BASIC interpreter.
Hard Disk	-	5 Mbytes 5.25" hard disk can be installed to replace the floppy directly.
Multiuser System	-	192Kbytes bank switchable memory. One parallel port to serial junction box which communicates with the terminals. One printer port. Operating system is MPM/II.

Again, I am able to inform you of a new unit to be added to the Genie range, the GENIE IV Colorcomputer, the specification of which is just being released. This unit with type number EG2000, is based on the Z-80 processor at a clock speed of 2.2 MHz. It has a 16K RAM and an 8K standard BASIC interpreter which can be upgraded to 16K, and in addition it has provision for plug-in program cartridges up to 12K. It uses a PAL colour TV or a monitor and is compatible with black and white units. The display has 24 rows and 40 columns. Its display seems to be quite extensive with upper and lower case, 128 graphic characters and 64 patterns available. There are another 64 patterns available under program control and 128 user programmable graphic character patterns. The screen resolution is 160 x 80 pixels. The colour display is made up of 9 colours for character display and 4 for full graphic display. Sound is also provided via an internal speaker through three separate channels. The I/O facilities are provided by both serial and parallel I/O and

in addition to the disk arrangement, there is a single cassette port operating at 1200 baud. All in all, this looks to be a very interesting machine and at an estimated price of only £190 ($A380) it should prove popular when it is available in September of this year.

Specification of Color Genie EG2000 PAL Version

1) CPU: Z-80 microprocessor at 2.2 MHz clock.
2) Programmable memory: 16K RAM
   - 1K for normal alphanumeric display.
   - 4K for full color graphics but can be used for programs if graphics is not used
   - expandable to 32K through internal 16K RAM pad.
3) BASIC Interpreter: 8K standard BASIC, and can be upgraded to 16K extended BASIC on board.
4) Provison for plug-in program cartridges, max. 12K.
5) Keyboard:
   - typewriter style
   - 64 graphic pattern
   - 4 programmable function keys
6) Display: both monitor & home TV, color & B/W compatible.
   - RF modoulator built-in
   - 24 rows x 40 coluns screen format
   - Upper and lower character display.
   - 128 graphic characters: 64 patterns available from keyboard and extra 64 patterns from program control.
   - 128 user programmable graphic character patterns.
   - 160x80 pixels full graphic resolution
   - 9 colors for character display, 4 colors for full graphics display
7) Sound: software control
   - 3 channels
   - built-in speaker
8) Built-inI/O facilities
   - serial I/O e.g. for RS-232 devices
   - parallel I/O e.g. for keypad, printer, joysticks
9) Provision for light-pen
10) Audito cassette as mass storage, 1,200 baud.
11) Resident switching power supply

- 0000000000 -

***** INPUT/OUTPUT *****

From: Mr. P.R. Smith, Donvale, Victoria.

HOW TO MAKE A SYSTEM TAPE FROM THE TANDY MICROCHESS TAPE

I recently acquired a copy of Tandy's MICROCHESS tape and, being the suspicious sort of person that 20 years' of computing makes you, decided that I would like to make a backup copy of it. I had read in lots of places that MICROCHESS has a special loader, can't be copied to disk, in fact can't be copied at all. I dragged out my trusty copy of DUPLIK which can usually copy anything - but to no avail. I tried a routine given to me by a friend ("this will copy anything - even Louis Armstrong") but this would not work either. This is not to say that these programs wouldn't read the tape - no trouble there, but what they wrote wouldn't load.

What next? At least the darn thing starts off like a system tape - it's only the later part that seems to give trouble.

What about SYSCOPY? (See MICRO-80 Oct & Dec 1980). This gives some degree of success - the first part (the MICROCHESS loader) loads in and can be copied out. More importantly SYSCOPY told me where it was so that it could be dissassembled. (It resides from 4FA1H to 4FF0H, entry 4FA1H). The dissassembly shows that all the loader does is read one long string of data into memory, starting at 40C0H up to 4F99, then checks the next byte on tape (if it's not zero it's out of sync and jumps to 0000H to give MEM SIZE?), then it reads the balance into video memory (3C00 to 3CFF) and branches to 41FDH to start the program.

Having discovered this, I thought that it shouldn't be too hard to change the loader to leave the routine in memory, so that SYSCOPY could copy it out in genuine SYSTEM format.

There are only one or two remaining problems - the areas that are loaded. Part of the tape is loaded directly into video memory (a trick that Tandy have used to cram it into 4K). This means that as soon as you do anything using the screen (such as SYSCOPY) then the video memory is destroyed. Also the other area loads over a lot of reserved vectors and getting back to BMON (yes you'll need that too) becomes a little difficult.

The solution is to change the loader to load the two areas into higher memory, append block moves to them to move them back, and save the high memory area with the entry point being the block moves.

By carefully choosing the location that the routine will now load into it should also be possible to make it compatible with most DOS's.

The following is a step-by-step recipe for the creation of a real SYSTEM tape version of MICROCHESS.

1. Load both BMON and SYSCOPY into your machine. (Or the combined SYSMON if you created it - see Dec. 80 MICRO-80).

2. Put the MICROCHESS tape in the reader.

3. Enter SYSCOPY and read the loader (respond with C to the *? prompt). You should have the following responses:

   START = 4FA1
   END = 4FF0
   ENTRY = 4FA1

4. Press BREAK to exit from SYSCOPY and enter BMON.

5. Apply the following patches to the loader using the E function of BMON -

   Loc Hex
   4FA3 70
   4FB3 53
   4FCA 62
   4FDE 63
   4FE5 67
   4FEE 99
   4FEF 7B

   These last two are the entry points to BMON. If you have moved BMON then you will have to modify the address. This will cause the loader to load the two areas of MICROCHESS into memory starting at 53C0H (instead of 40C0H) and 6300H (instead of 3C00H).

6. Use the G function of BMON to branch to address 4FA1H. (You should have left the tape in the recorder positioned at the end of the loader). The modified loader will now load the two areas of MICROCHESS. If all works the tape will stop and you will have the BMON copyright notice at the top of the screen, followed by READY. If it didn't work you will probably be either locked up or (more likely) looking at MEM SIZE? If this happens power off and start again!

7. The next step is to append the block move routines. Enter BMON and select the E function. Apply the following string of bytes to consecutive memory locations, starting at 62A0H:

   21 FF 66 11 FF 3F 01 00 40 ED B8 21 9F 62 11 9F
   4F 01 E0 0E ED B8 31 00 50 C3 FD 41

8. Now use the C function of BMON to write a system tape. The parameters are: START = 53C0 END = 66FF ENTRY = 62A0 NAME = CHESS Make two or more copies to be on the safe side.

9. Power your machine off (to eliminate all the various bits of MICROCHESS floating around inside), turn it back on and load your new MICROCHESS system tape.

- 0000000000 -

***** LEVEL I TO LEVEL II PROGRAM CONVERSION - by P. Aubin *****

Included with a LEVEL II TRS-80 is a program called CONV which, as the documentation included with it explains, provides a translation from LEVEL I statements and abbreviations to their LEVEL II counterparts. I would suggest that before trying to convert programs, that you obtain a program such as CONV, because not only does it change the LEVEL I program to LEVEL II, but it has LEVEL I tape routines in it!

The CONV documentation continues:

"This does not mean the converted programmes will always run as is."

Minor changes are often necessary. Using the LEVEL II EDIT command assists in this. The documentation lists a few changes which may have to be made, but neglects to mention quite a few others. Here is a list of possible changes which may have to be made, with examples where appropriate:

PRINT STATEMENTS:

Change commas after TABs to semi-colons, or omit them altogether.

```
Change PRINT TAB(20),"HI THERE"
    to PRINT TAB(20)"HI THERE"
```

Some ATs may not be changed to @ during conversion, change any ';' to ',' following the PRINT@, and Level II won't support multiple @ statements following 1 PRINT (I'm told the new ROMs which display "MEM SIZE?" will allow this), or PRINT@ without arguments.

```
Change PRINT AT 470;"example" to PRINT@470,"example"
change PRINT AT 0,"NOTE";AT 30,"WELL" to
      PRINT@20,"NOTE":PRINT@30,"WELL"
change PRINT AT 25;: to PRINT@25,"";:
```

INPUT TECHNIQUES:

Some LEVEL I programs allow the input of variables and expressions, not just constants. This was often used for letter inputs and compares (such as input a Y for YES and N for NO). In LEVEL II, string variables must be used.

```
Change Y=1:N=Ø:INPUT"YES OR NO (Y/N)";Z
       IFZ=1THENPRINT"YES!"
    to INPUT"YES OR NO (Y/N)";Z$
       IFZ$="Y"PRINT"YES!"
```

INKEY$:

LEVEL I lacks this very handy feature, but advantage can be made of the LEVEL I keyboard routine, which prints every character that is typed, to simulate it. Converting a LEVEL I program incorporating this, may unfortunately require you to have a knowledge of the internal operation of the program. The replacement BASIC text may look something like this:

```
       IF INKEY$ = "A" THEN B=1    or
       IF INKEY$ = CHR$(13) THEN 20   etc.
```

DATA STATEMENT:

LEVEL II DATA statements must not contain variables or expressions, only alpha-numeric or numeric constants are allowable.

ARRAYS:

LEVEL I only allows the use of 1 array (A). LEVEL II supports many more, but before they can be used, a DIM statement must be used, specifying the maximum array element number

```
          DIM B(25)
```

GRAPHICS:

Although the graphics commands are identical in both LEVEL I & II, the operation of them is slightly different. LEVEL II graphics doesn't wrap around as in LEVEL I. An FC Error will result if the X values aren't in the range Ø to 127, and Y values aren't in the range Ø to 47. LEVEL I POINT returns different values to LEVEL II POINT. LEVEL I: POINT(x,y) returns Ø if off, 1 if on. LEVEL II: POINT(x,y) returns Ø if off, -1 if on

```
       change D=D+POINT(X,Y)
           to D=D-POINT(X,Y)
```

IF & THEN:

In LEVEL II, not all IF statements require THEN statements following, but not all can do without as LEVEL I can. A program line such as:

```
IFA=CPRINT"EQUAL"
```

is correct, but one such as:

```
IF A=C D=7       must be changed to read:
IF A=C THEN D=7
```

Multiple comparisons differ between LEVEL I & LEVEL II.

```
change IF (A=21)*(B=14)THEN STOP
    to IF (A=21) AND (B=14) THEN STOP
change IF (Z<4)+(X=5)THEN END
    to IF (Z<4) OR (X=5) THEN END
```

LOGICAL EXPRESSIONS:

A TRUE logical expression (condition met), in LEVEL I returns a 1. An equivalent true logical expression in LEVEL II will return a -1, so some logical evaluations may have to change.

```
change Z = (A> C)*5
    to Z = (A> C)*5
```

VARIABLES:

LEVEL I has 26 single precision variables (A to Z), 2 string variables (A$, B$), and 1 array (A). All these are set with Random values on power up. LEVEL II has variables A to Z, AØ to Z9, AA to ZZ, in single precision, double precision, integer precision, as string variables and arrays which are all cleared when RUN or CLEAR is executed. For additional speed, precision etc. variables may be used followed by %, #, etc., or preceding any variable use with a DEFINT, DEFDBL, etc. statement.

If you are using any string variables, only 50 characters are set aside in LEVEL II for this. To increase the string capacity, a CLEAR instruction must be included, preferably at the beginning of the program.

CLEAR 500

Two other types of conversion may be necessary to make ALL your LEVEL I programs LEVEL II compatible. One concerns the use of tape for DATA STORAGE. A program called DCONV comes with CONV, and the documentation for this covers all aspects necessary to convert LEVEL I tape files to LEVEL II.

The other conversion worth consideration is that of LEVEL I MACHINE LANGUAGE programs. This is rather difficult, particularly if the program calls LEVEL I ROM routines. For further information, consult the hints and tips in the Tandy TBUG program. This may be helpful, but don't be too surprised if a re-write of the program becomes necessary.

I hope this information will help you in converting some of your valuable programs from LEVEL I to LEVEL II. GOOD LUCK!

- 0000000000 -

***** A LOOK AT SOME MORE SOFTWARE FROM BRYANTS EDUCATIONAL by Dave Futcher *****

Some nine months ago Bryan Etherington launched his own software company to supply educational programs for the TANDY TRS-80 and VIDEO GENIE to schools. He started with just a list of two or three programs but, over the months, he has built up an extensive list of programs for primary and secondary schools and a few general games programs. In his early work, Bryan showed that he was determined to use the power of the micro, especially in its graphics mode. The work that this review will consider shows how far BRYANTS EDUCATIONAL have moved along in the last few months.

MONEY BOX.

Despite its name, Money Box is not a program about money. It is a program which offers practice in multiplication and subtraction, although the money box has a crucial role to play in the idea behind the program. Every time that the user responds correctly there is a reward of ten pence that drops into the user's money box.

The program opens with a very good visual sequence in which a dragon-like creature snaps his jaws and sets the tone for some of Bryant's visual rewards that are to come later.

It is so important when a program begins that the options available to the user are clearly established. This program has an excellent initialization routine. The screen is clearly laid out and good use is made of the TRS-80 double character mode. Firstly, you have to choose between subtraction and multiplication and then you move into a sequence where the user is able to set the limits of the numbers that are to be used in the practice. These have been carefully thought out and are presented well. Teachers, even those who are not frequent users, would have no trouble in following the sequence. Unfortunately, one of the options at this stage is the ending of the program. This seems harmless enough till you discover that it wipes the whole program from memory, leaving you to load it in again for the next child. I quickly altered my copy of the program to stop this happening!

Let's look at the two modes offered. First then, the subtraction option. Much play is made of the idea of subtraction and complementary addition. To get this over to the children the subtraction problem is set in two ways -

```
 47          12 and what
-12          make 47
```

At first, the idea proved a little difficult and confusing to our children, but they soon found the turning of subtraction problems into addition ones a great way to deal with them.

The subtraction option worked very smoothly and had much to offer, except it had nothing built in to help the child to split tens or hundreds up to solve the problem on screen. I felt that this was a serious omission. It is a great shame when the children sitting at a computer suddenly have to reach for paper and pencil to solve a problem when the computer can, and should, be taking the children through the necessary sequence.

The multiplication option also used the idea of setting out the problem that the child had to work on in two ways. However, this time the options were only between one, as a formal multiplication problem, and the other in words. By using a simple routine the answer immediately appeared both places at once.

I was most pleased with what was on offer in two of the options. These enabled the user to attempt quite difficult long multiplication -

- multiply numbers 10 to 50 by 10 to 20
- multiply numbers 50 to 100 by 10 to 50

Here the program had been cleverly designed, as the child is taken through the sequence of multiplying parts of the sums and building up the answer correctly. Bryants had carefully thought out the screen layout and the actual breaking down of the multiplication into parts and then the adding of the parts to find the actual answer. Staff at my school actually used the two long multiplication options more than any of the others available in this program.

Unfortunately, the version of Money Box that I was using did not help the child if they made a wrong response. In fact, they could go on making it time and time again until the teacher realized. I understand the version now on release limits the number of accepted mistakes to three.

Now is the time to think about the correct responses because Bryan Etherington has put a lot of effort in this direction. Every time that a problem is solved correctly the idea of the Money Box is used. For every correct response a ten pence piece tumbles down the screen and into a money box. In later versions sound, too, has been added to this. When one hundred pence has been collected the program switches into its graphics reward sequence.

Bryants has built up a reputation for graphics and Money Box follows in the way of the previous offerings. When the pound has been won, the user is allowed to choose or buy various graphical rewards. These range from

- Jets that scream across the screen.
- a sausage dog that walks and yaps.

But my favourite, and the kids' too, is the Dragon which snaps and flashes its teeth. The great reward is to use a few keys to control a little man and take on this Dragon and fight to the death.

Money Box is sold at the reasonable price of £4.50 which, considering the work that it offers in the two areas of subtraction and multiplication, is good value, although I feel that the multiplication option actually has the most to offer as it is better designed.

The children, of course, loved to use the program and, even after many months of use, they have not tired of the fun that the graphic rewards bring. Bryants certainly know what to aim at in this area.

DECIMAL DAN.

In a fun way DECIMAL DAN provides practice in multiplying and dividng numbers by 10 and 100. The program has been aimed at increasing the child's awareness of place value in decimals. I've seen a number of 'homemade' programs in this area, some of which have been very successfully used at my school. Bryants have tried to build on this important area of place value and to produce a commercial program.

Decimal Dan is full of some fine pieces of clever programming that certainly give it much that the 'home-made' ones do not have. I think the sequence that actually allows you to move the decimal point around on screen is excellent. Here the arrow keys are used to very good effect. The user can move the point, look at it, think and move again. The decimal point is finally fixed into place by tapping the space bar. For reinforcement you are then asked to type in the correct number.

The program starts by asking you to multiply or to divide a number by 10 or 100. Of course, there is the usual initialization routine where the teacher or the child sets the level of the work. The screen is set up in double character mode and the problem to be attempted is boxed, as are areas that tell you the score and whether you are right or wrong. The screen layout is very clear to read and it is quickly obvious to the child what is expected of him/her. I was a little worried how some of the decimal fractions were actually displayed, as I felt that they would lead to confusion. For instance, 5.54 might sometimes be displayed as 05.54.

After tackling one problem and getting it correct you move to where Decimal Dan shoots down decimal points with his 'all action bow and arrow'. This animation reward was appreciated by the children but I feel that Etherington has unfortunately overdone the options available for a simple reward package. He should simply program one shot at the decimal points for each correct response.

Decimal Dan was certainly liked by staff and children but it is in need of a little bit of further work to improve its performance. It must have some on-screen instructions at the beginning so that the user knows what to expect and how to control the program. Without this, much time is wasted by the new user. Also, the way some of the decimals are presented must be tidied up and presented the way that teacher and children expect to see them. Again, the price is £4.50 which is fair for the type of work that the program offers. It is certainly at the normal Bryant level of graphical display and animation to warrant addition to any school's software collection.

ALADDIN.

Aladdin is one of Bryan Etherington's most ambitious attempts at the use of graphics in an educational program. In most of his other work, the graphics are there as animated reward when the user has been successful but, in Aladdin, you are treated to excellent displays of graphics all the time while using the program.

As the name implies, the program is based on the story of Aladdin and his magic lamp. It is really a practise program that enables you to work on simple division problems. At the beginning, the user is asked which table the division problem is to be on. Once set, the Aladdin appears in front of an enormous mountain. After finding your way to the right cave entrance hidden in the mountainside, the children find themselves inside a cave with stalagmites and stalactites and treasure boxes with, yes you have guessed, division problems to be solved, in them. You move from box to box, answering the questions in turn. There are four boxes and if you are correct, you are treated to an excellent inverse video routine that alters the treasure cave before your eyes; in addition, the score is increased. However, if you are wrong, beware! as one of the stalactites falls from the roof and crashes down on you. While all this problem-solving is going on, a little graphic Aladdin and his lamp is there, scampering across the screen. After 12 problems have been worked, you are given a report on your progress which tells you your score so far, and also lists your errors and their corrections.

ALADDIN is a simple and very useful program which will have great use in many classrooms. It has much to commend it but has a few things which I do not like that could be easily improved. Firstly, I wish that the scoring was clearing explained to the children at the beginning of the program. They get confused when they have points deducted for not finding the cave entrance straight away, and children do like to know what to expect for correct response. You really cannot just leave them to find or to work it out. A simple but important point. I do hate programs that compromise aspects of computing and teaching, and there is one part of this program that I really feel lets it down. This is the use of the slash instead of the proper division sign. We all know that the proper sign is missing from the character set of the TRS-80 Model I, and we all recognize the difficulty of creating it from graphics: it takes more than the six pixels per line and so needs at least two lines.

Bryants are selling Aladdin for their normal price of £4.50 which is again good value. Despite some of my criticism, it is worthy of a place in a school software library, and will get heavy use.

TOWERS AND CROCODILE.

These two programs sell at £3.50 each and are designed to go together. They have both been written to cover the area of odd and even numbers. They both have a place at the top end of an infant school or in the early years of a junior school.

TOWERS is all about using cubes and building them into towers, and if those towers are of equal height, the number is an even number. If the towers are not equal in height, the number is odd. The program recreates in animation the sorting and the stacking up of cubes just as children and teacher would do in a practical situation. The aim behind TOWERS is to let the child practise odd and even number recognition and to reinforce his success. I think there is only one area in which this program could be improved, and that is in the size of the cubes presented on the screen.

At the beginning of the program you have the chance to define the limits of the problems set. Two levels are on offer - up to ten, or up to twenty five.

CROCODILE really does complement TOWERS. It builds on to the practical sorting of cubes and instead, offers numbers to be assessed as either odd or even. CROCODILE is based on a bit of fun where a tame crocodile eats up the number if you respond correctly to the question asked. But if you answer incorrectly a whole screenful of crocodiles appear on screen and they advance across the screen and....After that, so that the crocodiles don't get you again there is a sequence that offers corrections.

TOWERS and CROCODILE cover a small area of mathematics but they do provide a different way to practise it. The animation has been cleverly done and is well worth using to complement work in this area. Both programs run smoothly and were enjoyed by children and staff.

Well, this is the second batch of programs that BRYANTS EDUCATIONAL have on offer that I have reviewed. Viewed as a group, they are a good range of programs that must be of interest to the TRS-80 and Video Genie educational users in schools. Bryants have tried very hard to harness the power of the computer and let it be used in an educational setting. Unfortunately, there are still not many people or companies around who are producing educational software. If Bryants carry on producing material like these on offer, they will do very well.

All of the programs ran very smoothly and were mostly bug free. From my previous dealings with Bryan Etherington I know that you can rely on him for sending out updates and corrections to the minor errors that become obvious only after long term validation in school.

A whole range of language development programs is also available from Bryants. These include Spellsort, Speechparts, Target Sounds, Storybuilder and Jumbles. At the moment the reviewed programs are, I understand, being updated to include sound routines that will add much to them. Also versions are being made available for the BBC MICRO and for the SINCLAIR ZX81. Programs can be ordered from Bryants Educational, 1 The Hollies, Chalcraft Lane, North Berstead, Bognor Regis.

- 0000000000 -

***** BASIC RECURSION by Michael Byrne *****

The topic of recursion occurs quite frequently in the field of programming, although mainly in the areas of mathematics, compiling and artificial intelligence. This describes the fundamentals of recursion and how it may be used by BASIC programs.

What is recursion? Essentially it is the ability of a function or procedure to call itself. (The difference between a function and a procedure is that a function can return a value to the calling program via the function name). Now, why would a function call itself? Well, there are some things which may be done much more efficiently and easily using recursive, as opposed to iterative methods. For instance many mathematical functions are recursive and most Pascal compilers are written recursively. In fact, some BASIC interpreters use it to evaluate arithmetic expressions. Further, most BASIC's allow the use of recursion in BASIC programs although this requires extra work on the part of the programmer as we shall see later.

To illustrate how recursion works, let's consider the example of calculating a factorial. Doing this recursively is very inefficient but the example was chosen because it is an easy one to understand as almost everyone will have calculated a factorial at one time or another.

The factorial of some number, N, is given by the expression

N! = N*(N-1)*(N-2)*.....*3*2*1

with 0! and 1! both defined as being equal to one. An alternative way of expressing this is

N! = N*(N-1)!

or

FACTORIAL(N) = N*FACTORIAL(N-1)

This is known as a recursive definition because the factorial is defined in terms of itself, i.e. it appears on both sides of the equation. At first glance it appears never-ending but no matter what N is initially eventually 1! will have to be evaluated and so the recursion will ultimately end. Think about it for a while until you feel comfortable with it.

What we will do now is develop a function which calculates factorials - let's call it FACTORIAL(N). To describe the algorithm, I will use a language which is similar to Pascal and convert it to BASIC later. As a first step we can take the definition of factorials given above, and write (the line numbers are just for reference purposes):

```
1  FUNCTION FACTORIAL(N: INTEGER)
2  BEGIN
3  FACTORIAL := N*FACTORIAL(N-1)
4  END
```

Line 1 declares the function and its name. It also indicates that the function has one parameter, an integer, which is called N. (Parameters allow values to be passed to the function for it to work on - generally values are not returned by the function parameters but by the function name). Lines 2 and 4 enclose the body of code which comprises the function. Line 3 does the work.

Notice that within the function the name of the function is used to return a value. The ':=' performs the same operation as the '=' in BASIC. However, before a value can be given to this instance of FACTORIAL, in line 3, it is necessary to evaluate FACTORIAL(N-1). At this point I will explain a bit more about N.

The variable N is said to be local to the function. It exists only while the function is active or executing. It cannot be accessed outside the function. Each time the function is called it reserves space for its own local copy of N, with an initial value corresponding to whatever was passed to it as a parameter when it was called.

As you may have noticed there is a problem with the function as it stands. We have not told it when to stop calling itself. We know that 1! is 1 so the function becomes:

```
1  FUNCTION FACTORIAL( N : INTEGER)
2  BEGIN
3          IF N = 1
4          THEN
5               FACTORIAL := 1
6          ELSE
7               FACTORIAL := N*FACTORIAL(N-1)
8  END
```

For all values of N greater than 1 the function will call itself; however when the function is called with a value of 1 then line 5 will be executed and the function will return with a value of 1. Program execution will then continue from line 7 in the calling function. So 1 will be multiplied by the current value of N for that function activation (i.e. 2) and the function will return to the level which called it. Eventually a return will be made to the main program and the factorial will have been calculated. The main program might be something simple such as

```
INPUT"NUMBER?";X
Y:= FACTORIAL(X)
```

Let's now trace the execution of the function when it is called with an initial value of 3. As yet the value of FACTORIAL is undefined so currently we have

```
N = 3                    FACTORIAL = ?
```

The "else" clause is executed and FACTORIAL is called again with a parameter value of 2, thus

```
N = 3                    FACTORIAL = ?
     N' = 2                          FACTORIAL = ?
```

(The prime ' is just a notational convenience to distinguish between the different calls of FACTORIAL). Remember that N is local to the function and cannot be modified outside it. It is a distinct variable in each invocation of the function, i.e. the N resulting from the second call of FACTORIAL is entirely separate from the N resulting from the first call, and the third, and the fourth...

Again the "else" clause executed as N still isn't 1 and so we have the following values:

```
N = 3                    FACTORIAL = ?
     N' = 2                          FACTORIAL' = ?
          N" = 1                               FACTORIAL" = ?
```

Now the current value of N is 1 so the "Then" clause is executed and the function returns with a value of 1.

```
N = 3                    FACTORIAL = ?
     N' = 2                          FACTORIAL' = ?
          N" = 1                               FACTORIAL" = ?
          N" = 1                               FACTORIAL" = 1
```

On return to the calling function, there is now a value for FACTORIAL(N-1) in line 7 so the function returns again after doing the multiply. Thus the complete sequence is

```
N = 3                    FACTORIAL = ?
     N' = 2                          FACTORIAL' = ?
          N" = 1                               FACTORIAL" = ?
          N" = 1                               FACTORIAL" = 1
     N' = 2                          FACTORIAL' = 2
N = 3                    FACTORIAL = 6
```

Thus in the main program Y becomes 6.

Now let's take a peek at what goes on behind the scenes when the function is called. One of the prerequisites for recursion is the stack. This allows a procedure or function to have local variables and also enables it to remember where it was called from so it can return to the right spot.

When the function is called, enough room is allocated on the stack for local variables, the return address and a few links of which we will only consider the base pointer. This information comprises what is known as the Stack Frame. Figure 1 shows the state of the stack when FACTORIAL has called itself once.

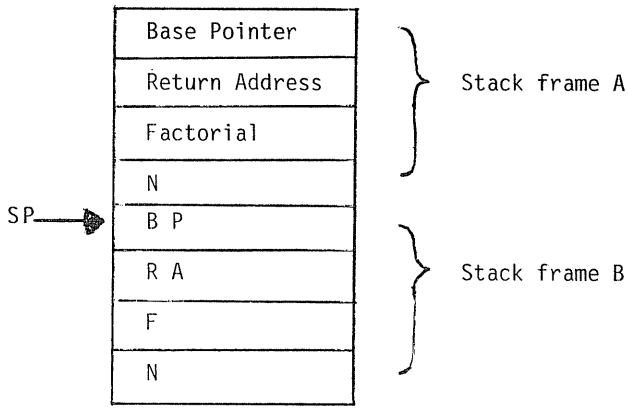


Figure 1

The base pointer merely points to the start of the previous stack frame so that the stack pointer can be reset when the current function returns. After the return, stack frame A becomes current and the space allocated to stack frame B is released for re-use.

All of this stack manipulation is done automatically in language such as Pascal, Simula and ALGOL etc. In BASIC the only autmatic stacking operations concern the return address. Also, since there are no local variables in BASIC it is up to the programmer to explicitly set up and maintain a stack if recursion is to be used.

The hard part now begins - converting the function to BASIC. I presume here that most of you do not have disk BASIC and hence it will be necessary to use a subroutine to perform the work of a function. Our main program will then be something like the following :

```
10  DIM SK(100)  'THE STACK
20  ' SP IS THE STACK POINTER
30  ' THE INFO CONTAINED IN THE STACK IS -
40  ' SK(SP) - THE BASE POINTER
50  ' SK(SP+1) - THE VALUE RETURNED BY THE FUNCTION
60  ' SK(SP+2) - THE VARIABLE
70  '
80  'MAIN PROGRAM
90  '
100 INPUT "NUMBER";X
110 IF X=0 THEN END
120 SP=1
130 SK(SP)=0
140 SK(SP+2)=X
150 GOSUB 210: 'CALL FACTORIAL
160 Y=SK(SP+1)
170 PRINT Y
180 GOTO 100
190 END
```

Note that lines 120 to 140 set up the stack prior to the call. The location SK(SP+1) is not initialised because its value is as yet undefined. Line 150 is the call and line 160 extracts the value returned from the call. The stack structure is shown in Figure 2.

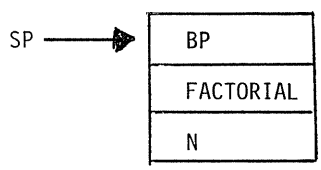


Figure 2

Now let's write the function.

```
200  '
210  ' FUNCTION FACTORIAL
220  '
230  IF SK(SP+2) = !
     THEN
          SK(SP+1) = 1 :
          GOTO 310
240  'ELSE
250       SK(SP+3) = SP
260       SP = SP+3
```

```
270       SK(SP+2) = SK(SP-1) - 1
280       GOSUB 210 ;'RECURSIVE CALL
290       SP = SK(SP)
300       SK(SP+1) = SK(SP+2) * SK(SP+4)
310       RETURN
```

Once again, notice that lines 250 to 270 set up the stack and adjust the stack pointer. Line 280 is the recursive call and line 290 resets the stack pointer. Line 300 performs the multiply i.e.,

N*FACTORIAL(N-1).

Type the program in and run it. The stack is big enough to let you calculate up to about 33! If you are still not too sure about how it all works then calculate a small factorial such as 4! and then hit BREAK. Now, while in command mode type

```
FOR I = 1 TO 12: ?I;": ";SK(I):NEXT I  ENTER/NEWLINE
```

You will see the contents of the stack displayed on the screen. The base pointers for each of the calls are stored at locations 1,4,7....The values returned at each stage are at locations 2,5,8.....while the local values for N are at locations 3,6,9...

There are no return addresses on this stack as BASIC maintains its own stack for that.

I should point out that no-one would seriously write a recursive function to calculate factorials - the overhead in terms of memory requirement and execution time is somewhat excessive. The example was chosen merely to illustrate the ideas behind recursion and one way in which it may be implemented in BASIC.

- 0000000000 -

***** BASIC BASIC - AN OCCASIONAL SERIES OF ARTICLES BY KEN B. SMITH *****

Ken Smith is a pilot in the R.A.F. stationed at Muscat, Oman. In between flights, Ken is obviously a keen and knowledgeable computer hobbyist. There is a very active computer club in Oman namely the Muscat Computer Group, of which Ken is a member. Given his experience, he is often called upon to teach other members much newer to the hobby, the rudiments of computing. Ken has submitted this article to us for publication and, hopefully, will follow it up with others just as informative and appealing.

O.K. all you big wheels - relax! This is a teaching article and as you don't need any tuition in BASIC, there is no point going any further. But before you go - format the numbers 1 to 100 on the screen in lines of 10 with all the hundreds, tens and units lined up vertically - without using USING or TAB. Tricky - not at all. Rest the pride a moment and read on.

What I will try and do in this article is introduce a few teaching topics that have proved very valuable in the past. The theme for this month - how about output formatting. Very few people get that totally correct and the reasons why are not always obvious. So let's take a couple of relatively straightforward problems and pull them apart as we go...

PRINT THE NUMBERS 1 TO 100 ON THE SCREEN IN ORDERED COLUMNS.

The first question must be - "What's an ordered column?"

A very good question which will probably answer itself as we progress, but I would assume for a start that all the hundreds, tens and units should be lined up. Now this problem has more answers than the one about the dimensions of a piece of string. The neatest solution that uses the full power of Level II BASIC is:

```
10 CLS : FOR X = 1 TO 100 : PRINT USING "###";X, : NEXT
```

This produces an orderly display but there is something missing. I would like to see ten numbers a line, 1 to 10 on the first line, 11 to 20 on the second and so on. To do this with the above program needs some fairly fancy calculations and the use of the TAB function. It then starts to look a little ragged and is also rather difficult to grasp. I won't include my version so that you more able programmers can get your OWN version running.

Well, for those still with me, let's take a very simplistic approach and start from basics...

To make a start we will use the one French instruction in the Level II vocabulary. (C)LEAR (L)E (S)CREEN - this will be a good beginning for neatness ake. Now add the start of a loop to give us the numbers 1 to 100 in the variable X.

```
10 CLS :REM Clear Le Screen
20 FOR X = 1 to 100 :REM Start a loop from 1 to 100 in X
50 PRINT X; :REM Print the value of X and stop the cursor moving
70 NEXT X :REM Loop between 20 and 60 until X = 100 then carry on.
```

Type this in and see if you get the same mess as myself. Untidy lines of figures all over the screen. But this is only a beginning. If you would use the same line numbers as above it will stop any grief in a moment when we add some more lines. The REMs are for those who aren't reading the text!!

Looking at my display it now appears that I need to add a carriage return and line feed to the program after the 10,20,30,40 and so on. The number of ways I have seen this done are legion, and some rather strange. For instance, I have had the following method of line feeding after 10,20 .... handed in as homework.

```
60 IF X = 10 OR X = 20 OR X = 30 OR X = 40 OR X = 50 OR X = 60 OR X = 70 OR X = 80 OR X = 90
OR X = 100 THEN PRINT
```

Now this was from quite an experienced programmer. However, he could obviously not see the wood for the trees, because disregarding the fact that this method does work - it is hardly elegant. Elegance is a valid goal because elegance = efficiency in memory usage and speed of execution.

Well, what method should we employ here? Look at the idea again. A PRINT after 10, 20, 30 and so on. What are these numbers but the 10 times table. Great but there are no BASIC statements to say:

```
60 IF X = PART OF THE 10 TIMES TABLE THEN PRINT
```

Well there is, but the words are slightly different. Try this in the command mode.

```
X = 7 : PRINT X/10, INT (X/10)
```

Now do that again with X equal to 10, 20 or 30. Notice anything? Of course. Much banging of the forehad and shouting at the dog later. The way that BASIC identifies its tables is:

```
60 IF X/10 = INT(X/10) THEN PRINT :REM Identify 10,20 etc. and PRINT
```

Add that to your program and RUN It. Looking a little neater, but why is that first line so short? The first line is so short because it only has little numbers in it!! What can we do about that? One should always try it in English, Arabic or Double Dutch first - because if you can't express a problem in your own language the odds are very high that you won't get your computer to understand you in BASIC. In English it comes out as:

```
30 IF THE NUMBER IS A SINGLE DIGIT THEN PRINT A SPACE IN FRONT OF IT.
```

Or put a slightly different way -

```
30 IF THE NUMBER IS LESS THAN 10 THEN PUT A SPACE IN FRONT OF IT.
```

The BASIC is a natural progression from the English - a valuable lesson. GET IT WORKING IN ENGLISH FIRST.

```
30 IF X<10 THEN PRINT " "; :REM ADD A SPACE BEFORE SINGLE DIGITS
```

Add that and sit back and think awhile. There is still one more task to fully tidy this program. The Hundreds don't line up. I have left you line 40 to add this finishing touch for yourself.

TOPIC 2 - HOW CAN WE CONTROL A PROGRAM WITH THE CURSORS?

This problem came up - yet again - only the other week. Essentially some of our club members are writing a game which involves moving a dot around the screen with the cursor arrows. (What game doesn't???) Anyway, they had used the INKEY$ function and were concerned that their keyboard would not last the week. To explain to those who are not familiar with INKEY$. INKEY$ is a method of extracting the result of a single key input from the keyboard into a STRING, which can then be decoded to find which key has been pressed. The biggest problem in this particular case is that INKEY$ is a single latch function, i.e. it only returns a key's value once, even if it is held down. To get any results in the application mentioned, the key must be repeatedly pressed - hence the concern. (N.B. Certain types of repeat key routines allow a multiple return in INKEY$, but it is very slow and not suited to game work).

It proved valuable to go through what is generally a standard method of accessing the keyboard for this type of application and you might benefit also...

The TRS-80 keyboard is a memory mapped device which is scanned and decoded from within the ROM. However, using the PEEK function directly into the Keyboard Memory we can access the information directly. This has advantages both in speed and elegance. There are quite a few locations that people use for this method. I always use 14426 decimal, not because it is particularly the best, but because I know what's there.

The first step in examining this particular feature is to write a simple program:

```
10 PRINT PEEK (14426); : GOTO 10 :REM PRINT DECIMAL CONTENTS OF 14426
```

Running this will show that there is a continuous return of decimal values ranging from 0 to 255 (a single BYTE's worth) which change whenever certain keys are pressed. This particular PEEK gives a different number for each cursor and combinations of cursors with a few extra keys recognised as well. If you care to use other values to PEEK from 14420 to 14430 you will find some other possibilities.

What you are actually seeing is the decimal representation of a BIT pattern generated by the decoding of the pressed key (s). Unless you have an I.Q. in the high 200's then the pattern will not be obvious. The following program uses the TRS-80 to help. (A noble use).

```
10 CLS                :REM CLEAR LE SCREEN
20 DEFINT A-Z         :REM INTEGER ONLY TODAY PLEASE
30 A = PEEK (14426)   :REM DECIMAL VALUE OF 14426 INTO A
40 FOR X = 0 TO 7     :REM LOOP X FROM 0 TO 7
50 IF 2  X AND A THEN PRINT "1";ELSE PRINT ".";
60 REM LINE 50 CHECKS BIT X TO SEE IF IT'S SET.
70 NEXT X             :REM LOOP BETWEEN 40 AND 70 UNTIL X =7
80 PRINT"",           :REM FORCE CURSOR TO THE NEXT TAB POSITION
90 GOTO 30            :REM START AGAIN AT 30
```

The above, hopefully, illustrates the principle although line 50 would be the topic of a whole article in itself. The TRS manual is rather sketchy on Logical Operators but see if you can manage. (Perhaps I will cover it another month if demand is there). The following is a condensed version that runs rather faster and is quicker to type in.

```
10 CLS:DEFINTX
20 FORX=0TO7:IFX 2ANDPEEK(14426)THENPRINT"1";ELSEPRINT".";
30 NEXT:PRINT"",:GOTO20
```

Well, that's about it for this month. The program 'DRAW' is a very early one of mine that illustrates the above techniques but does not get overly complex. Next time we will enhance the draw routines and get the 'pictures' stored away into strings. But that's another problem!

```
40 CLS:PRINT:PRINT:PRINTCHR$(23);TAB(12);"DRAW"
50 PRINTTAB(12);"====="
60 PRINT:PRINT:PRINT
70 PRINTTAB(5);"BY KEN B SMITH"
80 PRINT:PRINT:PRINT:PRINT"   DO YOU WANT INSTRUCTIONS";:INPUTZ$
90 IFLEFT$(Z$,1)="N"THEN300
100 CLS:PRINTTAB(15)"DRAW INSTRUCTIONS
110 PRINTTAB(15)"==================
120 PRINT"   THIS PROGRAM WILL LET YOU DRAW ANY DESIGN OF YOUR CHOICE
130 PRINT"ON THE SCREEN. YOUR CONTROLS ARE AS FOLLOWS:-"
140 PRINT"THE CURSOR ARROWS WILL MOVE THE LINE IN THE NATURAL SENSE.
150 PRINT"ALSO COMBINATIONS WILL PRODUCE A DIAGONAL LINE.
160 PRINT"  YOU CAN HAVE EITHER A POSITIVE OR NEGATIVE DRAWING. TO
170 PRINT"CHANGE THE FORMAT PRESS '1' AT ANY TIME IN THE RUN.
180 PRINT"   HITTING THE 'ENTER' WILL ERASE THE SCREEN AND LEAVE THE
190 PRINT"START POINT AT THE LAST POSITION. SO ALLOWING YOU TO
200 PRINT"BEGIN AT ANY POINT ON THE SCREEN.
210 PRINT"   HITTING THE SPACE BAR WILL FLASH THE CURSOR AND ALLOW YOU
220 PRINT"TO MOVE IT ANYWHERE WITHOUT A PRINT. TO EXIT FROM THIS
230 PRINT"PRESS THE SPACE BAR AND UP ARROW TOGETHER.
240 PRINT"  PRESS ENTER TO CONTINUE.....";:INPUTX:CLS
250 PRINT:PRINT:PRINT:PRINT"   YOU MAY ALSO LET THE 'TRS-80' DO THE DRAWING
    , EITHER BY
260 PRINT"ENTERING '2' NOW OR BY PRESSING A '2' AT ANY TIME IN THE RUN.
270 PRINT"  YOU MAY PAUSE THE PROGRAM BY HITTING 'SHIFT @ '.
280 PRINT"PRESS ENTER TO CONTINUE.....";:INPUTX
290 IFX=2GOTO890
300 X=63:Y=23
310 CLS
320 SET(X,Y)
330 RESET(X,Y):SET(X,Y):A=PEEK(14426)/8:IFA=16THENZ=1
340 IFA=17THENZ=0
350 IFZ=1THENSET(X,Y):RESET(X,Y):GOTO420
360 A=PEEK(14426)/8
370 IFA=0GOTO330
380 IFA=.125GOTO 310
390 IFA=.25GOTO600
400 IFA=.5GOTO890
```

```
420 ONAGOTO440,460,30,480,500,520,30,540,560,580
430 GOTO330
440 IFY>0THENY=Y-1:GOTO330
450 RESET(X,Y):GOTO330
460 IFY<47THENY=Y+1:GOTO330
470 GOTO450
480 IFX>0THENX=X-1:GOTO330
490 GOTO450
500 IFX>0ANDY>0THENX=X-1:Y=Y-1:GOTO330
510 GOTO450
520 IFX>0ANDY<47THENX=X-1:Y=Y+1:GOTO330
530 GOTO450
540 IFX<127THENX=X+1:GOTO330
550 GOTO450
560 IFX<127ANDY>0THENX=X+1:Y=Y-1:GOTO330
570 GOTO450
580 IFX<127ANDY<47THENX=X+1:Y=Y+1:GOTO330
590 GOTO450
600 FORZ=15360TO16383:POKEZ,191:NEXT
610 RESET(X,Y)
620 SET(X,Y):RESET(X,Y):A=PEEK(14426)/8:IFA=16THENZ=1
630 IFA=17THENZ=0
640 IFZ=1THENRESET(X,Y):SET(X,Y):GOTO710
650 A=PEEK(14426)/8
660 IFA=0GOTO620
670 IFA=.125GOTO600
680 IFA=.5GOTO890
690 IFA=.25GOTO300
710 ONAGOTO730,750,600,770,790,810,600,830,850,870
720 GOTO620
730 IFY>0THENY=Y-1:GOTO620
740 SET(X,Y):GOTO620
750 IFY<47THENY=Y+1:GOTO620
760 GOTO740
770 IFX>0THENX=X-1:GOTO620
780 GOTO740
790 IFX>0ANDY>0THENX=X-1:Y=Y-1:GOTO620
800 GOTO740
810 IFX>0ANDY<47THENX=X-1:Y=Y+1:GOTO620
820 GOTO740
830 IFX<127THENX=X+1:GOTO620
840 GOTO740
850 IFX<127ANDY>0THENX=X+1:Y=Y-1:GOTO620
860 GOTO740
870 IFX<127ANDY<47THENX=X+1:Y=Y+1:GOTO620
880 GOTO740
890 CLS:PRINT"THIS IS A 'TSR 80' PICTURE. WOULD YOU LIKE A NEGATIVE OR A
900 PRINT"POSITIVE SCREEN?
910 PRINT:PRINT:
920 PRINT"TYPE 'P' OR 'N'
930 Y$=INKEY$
940 IFY$="N"GOTO1040
950 IFY$="P"GOTO970
960 GOTO930
970 CLS
980 X=63:Y=23
990 B=RND(3)-2:C=RND(3)-2:X=X+B:Y=Y+C
1000 IFX<1ORX>126GOSUB1120
1010 IFY<1ORY>46GOSUB1120
1020 RESET(X,Y):SET(X,Y):Y$=INKEY$:IFY$="1"GOTO330
1030 GOTO990
1040 FORZ=15360TO16383:POKEZ,191:NEXT
1050 X=63:Y=23
1060 B=RND(3)-2:C=RND(3)-2:X=X+B:Y=Y+C
1070 IFX<1ORX>126GOSUB1120
1080 IFY<1ORY>46GOSUB1120
1090 SET(X,Y):RESET(X,Y):Y$=INKEY$:IFY$="1"GOTO330
1100 GOTO1060
1110 END
1120 IFX<1THENX=1
1130 IFX>126THENX=126
1140 IFY<1THENY=1
1150 IFY>46THENY=46
1160 RETURN
```

***** THE TRS-80 COLOUR COMPUTER - AN EXAMINATION OF ITS DESIGN AND CAPABILITIES, AND ITS POSSIBILITIES AS AN EDUCATIONAL COMPUTER

by Dave Futcher *****

The TRS-80 has been long accepted as one of the four main educational computers in the UK and its Model I has been at the forefront of educational computing for the past three years or so, the Model I was joined by the Model III about a year ago. Now the TRS-80 has a colour computer on the market. It was launched in December 1981 in Britain and is now becoming available in volume.

Two models are currently available but, for this review, we used a machine with 16K RAM and EXTENDED COLOUR BASIC which is sold for £449 at this time. A model with 4K RAM is the other alternative but with a less powerful BASIC.

WHAT COMES WITH IT.

For £449 you get the following:

16K Colour computer with Extended Basic
Built in RS-232 I/O port and real time clock
Cable to connect computer to TV
Operating manual
Colour BASIC instruction manual
Extended Colour BASIC instruction manual
Card listing statements, functions, and operations.

You must supply your own colour TV set (or black and white) and tape recorder.

The computer uses a 6809E Motorola eight bit microprocessor chip with a clock speed of 0.894 MHz.

THE COMPUTER CASE AND ALL.

As in other models, the Colour computer comes in the now traditional grey and silver plastic casing. It looks as sturdy as ever and I am in no doubt as to the reliability of the strength of this material, having seen a few TRS-80's bounce in recent years. Unlike the Model I, the TRS-80 Colour computer has its power supply inside the case so that reduces the number of leads trailing. In fact, the power supply unit seems to take up a large proportion of the case but there appears to be no problems with overheating. The case measures 37cm wide and 34cm deep, and is well mounted on rubber feet that certainly do not let the computer slide about on desktops.

At the back of the computer there are DIN connectors for a tape recorder, serial printer or modem, and dual joystick controllers. In addition this is where you connect the computer to the TV. Here, too, is where you find the on/off switch and well tucked away is the reset button.

Near the back on the right hand side is the slot for ROMPAKS suitably protected with a flap.

THE KEYBOARD.

I was more than a little concerned when I first saw the keyboard. Although it had calculator style keys, they were certainly much better than those of the early PET. They were well spaced out and quite suitable for typing and entering programs, but why did Tandy move away from the excellent keyboard that they had provided on their other machines? In school the keyboard has presented no problems whatsoever. Children have used quite happily the keyboard for Computer Aided Learning (CAL) packages. In fact, there is one great asset of this keyboard. It is really easy to produce masks that slot over the keys; the use of such a mask has tremendous potential in education for labelling keys to represent responses.

In the package ART GALLERY from Tandy the overlay keyboard is used to very good effect and the children who have worked with this program have appreciated the overlay. I understand that the Colour computer word processing package will have a similar overlay for remembering defined functions.

SCREEN DISPLAY.

The letters and symbols displayed on the screen are of good size and are extremely clear for even the youngest children to read. Unfortunately, the machine that we have at present is only able to display on screen in upper-case letters, but I understand that a lower-case modification is shortly to be released and will be available in the UK. The screen is sixteen lines deep and 32 upper-case characters across. As I have already said, this provides for real clarity in an educational setting, and within any normal CAL work the length of line should present no problems, although I must admit that when programming, I found the 32 character lines a little cramped.

The clarity and colour of your screen will depend upon the condition of your TV set. For the purpose of this review, I have used a 14 inch SONY and a 16 inch PHILLIPS very successfully. In addition, some use has been made of the large school TV. We found no real problems with colours not being as sharp as they should be.

The screen display is normally black on a green background, but this can easily be changed to red on yellow within a program.

THE GRAPHICS CAPABILITIES.

In the past, those familiar with Tandy computers have had to deal with blocks of white graphics in screens of 48 x 128, but now, with the colour computer, you have as many as eight colours, nine if you include black, and five modes of resolution ranging from 32 x 64 to 192 x 256. The colours available are -

green
yellow
blue
red
buff
cran
magenta
orange

When working at maximum resolution of 192 x 256, up to four colours are available. As you would expect, there is a trade-off between memory consumption, colour choice and the resolution.

When you switch on, four pages of graphics memory are reserved for use. Each page uses 1.5K and it is possible to reserve up to eight pages.

PROGRAMMING IN COLOUR GRAPHICS.

There are three commands that must be used in sequence to program in colour graphics. These are -

PCLEAR This will reserve up to eight 1.5K pages of memory for graphics. If you don't use the command, four pages are automatically reserved.

PMODE This command sets up the resolution to be used and the starting page of the screen memory.

SCREEN This instructs the computer as to the screen type (graphics or text) and the colour set, depending on the colour mode.

Two additional commands are needed but these can appear anywhere in the program and in any sequence following the PCLEAR statement.

PCLS This will clear the screen to a specified colour.

COLOUR This sets up the colour of the fore and backgrounds used by the other graphics commands.

Let's look now at some of the graphics commands that back up the hardware and really make use of the power of the colour computer.

The LINE command, as you would expect, draws lines between two specified points. If only one pair of coordinates is specified, the line is drawn between the point and the previous point used. The line command can also be used to erase a line too, as well as creating one. To do this you merely draw in the background colour that you are using.

The CIRCLE command, as you would expect, allows you to draw circles but it is also possible to draw ellipses and arcs as well. Drawing circles on the screen is surprisingly easy. You only have to specify the centrepoint and the radius and there is a further feature that enables you to specify the colour it is drawn in.

Perhaps the most wonderful of the graphics commands is DRAW. This command presents an easier way of doing some of the things that you may wish to attempt with the LINE and CIRCLE commands. DRAW enables you to draw a line or a series of lines by specifying their direction, angle, and colour, all in the same program line. This command takes the form of a string expression. It is possible, within DRAW, to move in eight different directions and there is also the amazing potential to scale up and down the object that you are drawing.

Once you have produced an outline shape, whether it be by LINE, CIRCLE or DRAW, it is possible to use another Colour BASIC word to fill it in. The command here is PAINT. You merely have to specify a starting point and the shape is filled in.

MICRO-80 PRODUCTS - CATALOGUE

HIGH QUALITY PRODUCTS FOR YOUR COMPUTER AT UNBEATABLE PRICES.

ABOUT MICRO-80 PRODUCTS

MICRO-80 PRODUCTS was started at the request of MICRO-80 readers who wanted to obtain good quality peripherals and software for their computers at reasonable prices. In the past 2 1/2 years literally thousands of satisfied customers can attest to the fact that MICRO-80 PRODUCTS has achieved this objective. We have removed much of the mystique which surrounds the interfacing of such useful peripherals as disk drives and printers and have become the major Australian source of supply for such software products as NEWDOS and DOSPLUS which have increased the power and speed of TRS-80 micro-computers enormously. More recently, we have saved Hitachi owners considerable sums by interfacing MPI disk drives to the Hitachi Peach. We were the first in Australia (in the world?) to successfully interface the range of Olivetti electronic typewriters to be printers and have designed and produced a number of useful modifications for the TRS-80/System 80.

As the interest of micro-computer users broaden, so do our own. We now actively sell and support the TRS-80 Model 3, the Osborne 1, the Hitachi Peach, the Olivetti M20 microcomputer, the North Star Advantage and the Altos multi-user system. We would be happy to assist you in upgrading your present computer with new peripherals or even exchanging it for a more modern machine.

MAIL ORDER POLICY

Much of our business is carried out by Mail Order and our customers find it a simple and efficient way to do business. You may place your order by telephone or by mailing in the order form from any issue of MICRO-80 magazine. Generally, it takes about 1 week from receipt of order until dispatch. You should allow 2-3 days for your letter to reach us and 7-10 days for the parcel to reach you making a total turnaround time of about 3 weeks. If we are temporarily out of stock of an item, we will send you a notification of back order giving our best estimate of when it will be back in stock. Payment, which should accompany the order, may be by Cheque, Money Order, Bankcard or Access. If we are unable to supply an order immediately, we apply the following rules:

-If payment is by cheque and none of the order is in stock, the cheque is not presented until the order can be fulfilled.

-If payment is by cheque and some items are in stock, the cheque is presented and the items back ordered are shown on the invoice which accompanies the goods.

-If payment is by Bankcard or Access, only these items which can be supplied are charged. Back ordered items are not charged until available.

If you wish to speed up delivery, you may pay a special delviery fee to have the item sent by road freight or even air express. Ring for prices.

WARRANTY AND SERVICE

All hardware products carry a 90 day parts and labour warranty either from the manufacturer/distributor or from MICRO-80 PTY. LTD. In many cases, warranty servicing can be arranged in your own city, otherwise goods should be returned to MICRO-80 PTY. LTD. the cost of freight to MICRO-80 is at customer's expense. Return freight on goods which require repair or adjustment, either by road or post at MICRO-80's discretion, will be paid for by MICRO-80 PTY. LTD. Customers should obtain a return authorisation from MICRO-80 before despatching goods for warranty repair, post warranty servicing can also be carried out at very reasonable rates.

-0000000000-

TRADE-INs, EASY PAYMENT TERMS

MICRO-80 BRINGS COMMONSENSE TO COMPUTER BUYING

If you wish to buy a new car, you are able to trade-in your existing vehicle and arrange finance for your new purchase, all under the one roof. Not so with microcomputers. If you want to dispose of an existing machine, you are on your own and, in most cases, you must make your own arrangements about finance, too. Here at MICRO-80 we think this is ridiculous, so we have done something about it. We are now able to accept TRADE-INs on new COMPUTERS and PERIPHERALS and to arrange CONSUMER MORTGAGE terms to approved customers. This offer applies to our customers ALL OVER AUSTRALIA, not just in South Australia.

Here is what you do.
If you are interested in trading-in existing equipment:-

1) Write to us or phone us, describing the equipment you wish to trade-in. Make sure you tell us its age and any distinguishing features. Eg.: TRS-80 Model 1, early style keyboard with "square' monitor, L2/16K, 3 years old, good condition.

2) Tell us too, what computer you wish to purchase from our range of Hitachi, TRS-80 Model 3, Osborne, Olivetti and North Star.

3) We will write, offerring you a trade-in valuation and quoting the price of the equipment requested. Our trade-in offer will be subject to inspection of the equipment at our premises. Our letter will also include instructions for sending the equipment to us in the most cost effective manner.

4) If you are satisfied with our offer and quotation, together with payment for the balance (or if you wish to purchase on terms, see 6 below) and we will send your new computer to you.

If you would like to take advantage of consumer mortgage or leasing finance, with or without a trade-in:-

5) Write or 'phone telling us the equipment you wish to purchase.

6) We will send you a written quotation, an order form and an application form for the appropriate finance.

7) Complete the application and order forms and return them, together with a cheque made out to the finance company indicated, for the first monthly payment. The order is conditional upon you obtaining finance of the required amount at the quoted rate. If this is not available at the time the order is received, we will contact you for further instructions. South Australia has some of the most stringent regulations in Australia affecting consumer financing and you may rest assured that your interests will be well protected.

8) When authorised to do so by the finance company, (generally 3-7 days) we will despatch the new equipment to you.

EASY PAYMENT TERMS ALSO AVAILABLE ON PERIPHERALS

The same consumer finance is also available on hardware peripherals selling for more than $250. For example, if you require a disk drive costing $499, you could purchase it on 10% deposit and payments of only $4.17 over a period of 36 months.

Even software can be included in the overall purchase to a limited extent. Eg.; If you purchase a new computer system then you could also finance a Disk Operating System and application programs up to about 10% of the total value of the purchase.

WE HAVE CUSTOMERS WAITING FOR USED COMPUTER SYSTEMS

In high demand are TRS-80 Model 1 systems with one or more disk drives. If you have such a system, why not trade it in on a new computer?

Finance and leasing facilities to approved clients is available through "ESANDA" Adelaide.

-0000000000-

***** BOOKS *****

THE CUSTOM TRS-80 & OTHER MYSTERIES
$32.50 + $1.20 p.&p.

The complete guide to interfacing your TRS-80 to the outside world, covering both hardware and software.

TRS-80 DISK & OTHER MYSTERIES
$27.00 + $1.20 p.&p.

A must for the serious disk user. Disk file structures revealed. DOS's compared and explained, how to recover lost files, how to rebuild crashed directories.

LEARNING LEVEL 2 NOW ONLY $7.95 +$1.20 p.&p.

Written by David Lien, the author of the TRS-80 Level 1 Handbook, this book teaches you, step-by-step, how to get the most from your Level 2 machine. Invaluable supplement to either the TRS-80 Level 2 manual or the System 80 manuals.

Level 2 ROM ASSEMBLY LANGUAGE TOOLKIT
$29.95 + $1.20 p.&p.

The definitive work on using Level 2 ROM routines in your own programs. Covers TRS-80 Model 1 and 3 and System 80. Comes complete with DBUG, a machine language debugging monitor distributed on cassette. This package is a must for machine language programmers and BASIC programmers.

BASIC BETTER AND FASTER
$32.50 + $1.20 p.&p.

Fast becoming the "bible" on the TRS-80 for BASIC programmers , this book is packed full of useful routines and techniques all fully explained, which you can use in your own programs. If you are serious about learning to program, then this is a must.

LOOKING FOR A NEW COMPUTER?

MICRO-80 PRODUCTS HAS A COMPLETE RANGE OF
HIGH QUALITY MICRO-COMPUTERS FOR EVERY REQUIREMENT.

HITACHI PEACH - Add Some Colour To Your Life.
The Hitachi Peach is built to the highest quality standards we have seen in a microcomputer. Not surprising really when it comes from Japan's leading computer company. The Peach can be a versatile, cassette based system for home use or a fully expanded business system carrying out word processing, accounting, financial planning etc. etc. When you see a Peach, you will quickly be convinced, as we are, that it is the shape of things to come. Within the next few years high resolution colour graphics will become a standard feature on business computers. Fortunately, you do not have to wait. The Hitachi Peach is here now and so is high quality, applications software for business use.

Standard features of the "Peach" include 40/80 character display, Upper/Lower case with descenders, high density graphics (640 x 200), 8 colours, numeric keypad, ten programmable function keys, speaker with volume control, RS232 Port, Centronics parallel printer port, Red/Green/Blue video output, light pen port, 32K RAM, 24K ROM, six expansion slots, RAM expandable to 64K in cabinet, 6809 Microprocessor, and Microsoft basic.

		PRICE	*WEEKLY PAYMENTS
32K CASSETTE SYSTEM Consists of Hitachi Peach with 32K RAM, 24K ROM, Mono Cassette deck and High Resolution Monitor	Green Screen Colour	$2,100 $3,000	$17.57 $25.10
48K SINGLE DENSITY DISK SYSTEM Hitachi Peach with 48K RAM, 24K ROM, Disk Controller two 5 1/4 inch single-sided single density MPI disk drives (80 Kbyte capacity each) and disk operating system	Green Screen Colour	$3,300 $4,200	$27.79 $35.13
Two 5 1/4 inch double-sided single density disk drives, controller, and disk operating system in lieu of single-sided drives.	Add	$ 250	$2.09
48K DOUBLE DENSITY DISK SYSTEM Hitachi Peach with 48K RAM, 24K ROM disk controller two 5 1/4 inch double-sided, double desnity disk drives (280 Kbyte capacity each) and Disk operating system.	Green Screen Colour	$4,400 $5,300	$36.81 44.33
COMPLETE PROFESSIONAL WORD PROCESSOR IN COLOUR! 48K single density disk system with colour monitor ITOH 40 character per second daisywheel printer and HiWriter word processing program.		$6,450	$53.95
COLOUR ACCOUNTING SYSTEM 48K double-density disk system with colour monitor, Star dot matrix printer and any three modules from the HiFinance Accounting Software e.g. Debtors, Inventory and Invoicing. This System is capable of handling up to: - 1200 inventory items - 100 creditors - 100 general ledger accounts - 300 transactions per month	From	$6,900	$57.72
The capacity can be more than trebled by using DS/DD 8 inch drives.	Add	$ 700	$5.85

Finance to approved customers is available through "ESANDA" Adelaide. Weekly payment figures shown are approximate only and are calculated after deducting 10% deposit from the selling price, using an effective rate of interest of 25.98% pa over a period of 36 months.

-0000000000-

ANNOUNCING THE OLIVETTI M20

FULL 16 BIT PROCESSING POWER IN AN OFFICE MACHINE.

The Olivetti M20 represents a breakthrough in computer design - both technically and aesthetically. From its latest full 16 bit technology using a Z8000 microprocessor to its attractive styling, it is as modern as tommorow. Most important however, are the human interfaces. Olivetti has drawn on its years of expereince in office equipment design to produce a computer which is easy and comfortable to use. The keyboard is well laid out with a nice "feel", the disk drives are quiet and unobtrusive whilst the all important display is without equal. We have not seen such an attractive, easy to read display anywhere. The special anti-glare treatment makes text readable under the most adverse lighting conditions whilst characters are of high definition and clearly formed. The high resolution graphics display (512 x 256) is sharp and rock-steady. If you think we are laying it on a bit thick, then you haven't seen the M20!

The Olivetti M20 is primarily designed to be used in a commercial or technical office although its low price could also make it attractive to the serious hobbyist or programmer. The basic specification is: Full 16 bit Z8001 microprocessor, 128K RAM expendable in the cabinet to 512K, one 5 1/4 inch, double-sided, double density 35 track disk drive (320K unformatted capacity) RS232 serial selectable port, Centronics parallel port, software 80 x 24 or 64 x 16 display with high resolution characters - lower case with full descenders, fully programmable keyboard, PCOS Disk Operating System and MICROSOFT Version 5.2 BASIC interpreter on Disk and high resolution graphics (512 x 256).

Options include, second disk drive in cabinet, winchester hard disk drive, IBM 3870 terminal emulation, videotext, high resolution colour graphics.

The operating software and BASIC interpreter support graphics commands such as DRAW, BOX, PAINT, 16 INDEPENDENT WINDOWS, LABELS, WHILE AND WEND AND MANY OTHER POWERFUL STATEMENTS. The operating system and interpreter reside in RAM so no system disk is required after boot-up thus making a single disk system viable. Olivetti is fully supporting the M20 with a wide and ever increasing range of applications software. Available immediately is the OLIBIZ suite of accounting programs - written in Australia, MULTIPLAN financial planning, ISAM file handling utility and a wide range of scientific and engineering sub-routines. which have been thoroughly tested on earlier Olivetti systems. Releases scheduled over the remainder of 1982 include OLIWORD word processor, OLIENTRY data handler, OLISORT high speed sorting routine and an Editor/Assembler. If you require an up-to-date powerful super-fast and well supported computer for your business, then you should seriously consider the OLIVETTI M20.

SINGLE DISK DRIVE SYSTEM consists of Olivetti M20 with 128K RAM, one DS/DD disk drive (320K unformatted) PCOS disk operating system and Microsoft BASIC 5.2	$4,400 inc. ST. $37 per week
ACCOUNTING SYSTEM consists of Olivetti M20 with 128K RAM two DS/DD disk drives (320K unformatted) complete suite of OLIBIZ programs including Debtors, Creditors, Inventory, Invoicing and General Ledger, 80 Column dot matrix printer.	$7,500 incl. ST. $63 per week

Add $30 for road freight to anywhere in Australia.

Finance to approved customers is available through "ESANDA" Adelaide. weekly payment figures shown are approximate only and are calculated after deducting 10% deposit from the selling price, using an effective interest rate of 25.98% over a period of 36 months.

-0000000000-

THE OSBORNE 1 BUSINESS SYSTEM
$2,595
THE BEST VALUE FOR MONEY IN COMPUTING

Without any shadow of a doubt, the Osborne 1 represents the best value for money available in a computer system anywhere. Not only does $2,595 buy you a 4MHZ, 64K CP/M computer with two disk drives but also a well thought out selection of applications and utility software. Osborne pioneered the complete package aproach in microcomputers and no one else has been able to equal it. The software supplied with the Osborne includes:

- WORSTAR word processing program. WORDSTAR is undoubtedly one of the leading word processors available on a microcomputer.	Retail Value	$ 495.
- MAILMERGE, used in conjunction with WORDSTAR, MAILMERGE enables you to carry out mass mailings of personalised letters.	Retail Value	$ 150.
- SUPERCALC, sophisticated electronic spread sheet for budgeting, cash flow predictions engineering calculations, prices lists etc.	Retail Value	$ 295.
- CBASIC compiler for developing your own application programs.	Retail Value	$ 150.
- MBASIC BASIC interpreter for writing interactive programs.	Retail Value	$ 390
- CP/M Disk Operating System	Retail Value	$ 150.
	TOTAL RETAIL VALUE OF SOFTWARE	$ 1630

As if all that is not enough, the Osborne 1 is portable. You can take your computing power with you, wherever you need it. Do your budgetting in the quiet of your own home, enter your sales results and orders in the motel room as you travel, write your reports while they are still fresh in your mind. Wherever you go your Osborne computer can go to increase your productivity and automate those dreary administrative tasks.

OSBORNE PORTABLE WORDPROCESSING SYSTEM CONSISTS OF OSBORNE 1, PRAXIS 35 TYPEWRTER/PRINTER AND CABLE.	$3,590 $33 per week
OSBORNE WORDPROCESSING STATION CONSISTS OF OSBORNE 1, ITOH 40CPS DAISYWHEEL PRINTER AND CABLE 12 INCH GREEN MONITOR.	$4,850 $41 per week

Add $20 for road freight to anywhere in Australia.

Finance to approved customers is available through "ESANDA" Adelaide. Weekly payment figures shown are approximate only and are calculated after deducting 10% deposit from the selling price, using an effective interest rate of 25.98% over a period of 36 months.

-0000000000-

BUY YOUR MODEL 3 FROM MICRO-80 AND SAVE $000's

MICRO-80 fits reliable MPI disk drives to the TRS-80 Model 3 to give system capacities and capabilities far in excess of those available elsewhere. All our conversions utilise low dissipation, switching-mode power supplies to avoid screen jitter and overheating. The disk controller boards used incorporate special compensation circuitry for 80 track disk drives and may also be used to run 8 inch disk drives with an appropriate cable and DOS.

MODEL 340
2 40 TRACK SINGLE-HEAD DISK DRIVES GIVING
350K FORMATTED STORAGE, 48K RAM **$3130**

MODEL 340+
2 40 TRACK DUAL-HEAD DRIVES GIVING
700K FORMATTED STORAGE, 48K RAM **$3350**

MODEL 380+
2 80 TRACK DUAL-HEAD DRIVES GIVING
1.4 MEGABYTE FORMATTED STORAGE, 48K RAM **$3800**

★ *NEW* ★ ★ *NEW* ★ ★ *NEW* ★

MODEL 500 — 5+ MEGABYTE MODEL 3
1 40 TRACK DUAL-HEAD DRIVE GIVING
350K OF FLOPPY DISK STORAGE FOR TRANSFERRING
PROGRAMS AND BACKUP, 48K RAM, EXTERNAL
5 MEGABYTE WINCHESTER SUB-SYSTEM,
CP/M (ORG 4200N) DISK OPERATING SYSTEM **$5895**

The MODEL 500 offers the high speed, mass storage capacity and reliability of a Winchester drive for thousands of dollars less than you would pay for any comparable system. Model 500 is a serious business computer able to tackle the most demanding tasks.

All prices are in Australian dollars, include Sales Tax and are subject to change without notice. Prices are FOB Adelaide. Add $20 road freight anywhere in Australia. All computers and systems carry MICRO-80's 90-day Warranty covering parts and labour.

PRINTERS GALORE AT UNBEATABLE PRICES

MICRO-80 has a range of printers to suit every requirement from dot-matrix to correspondence quality daisywheel. Chose from the table below:

BRAND	MODEL	TYPE	SPECIFICATIONS									
			COL	SPEED CPS	BI-DIR	LOWER CASE	PAPER FEED	GRAPHICS	INTER FACES	FREIGHT	PRICE	WEEKLY PAY-MENTS*
STAR	DP	DM	80	80	Y	ND	F/T	BLOCK	P	1	$ 575	$4.81
EPSON	MX-80	DM	80	80	Y	FULL	F	BLOCK	P	1	$ 899	$7.53
EPSON	MX-80II	DM	80	80	Y	FULL	F/T	HI-RES	P	1	$ 999	$8.35
EPSON	MX-100	DM	132	100	Y	FULL	F/T	HI-RES	P	1	$1500	$12.55
MICROLINE	83A	DM	132	120	Y	FULL	F/T	BLOCK	P/S	1	$1599	$13.37
MICROLINE	84	DM	132	200	Y	FULL	F/T	HI-RES	P	1	$2220	$18.57
MICROLINE	84	DM	132	200	Y	FULL	F/T	HI-RES	S	1	$2340	$19.57
C ITOH	8510	DM	80	112	Y	FULL	F/T	HI-RES	P	1	$1099	$9.19
C ITOH	M1550	DM	132	120	Y	FULL	F/T	HI-RES	P	1	$1499	$12.54
DATA SOUTH	DS-180	DM	132	180	Y	FULL	T	OPT.	P/S	1	$2590	$21.66
OLIVETTI	PRAXIS30	DW	100	6	N	FULL	F	NO	P	1	$ 895	$7.49
OLIVETTI	PRAXIS35	DW	100	6	N	FULL	F	NO	P	1	$ 995	$8.33
OLIVETTI	ET121	DW	132	12	N	FULL	F	NO	P	2	$1500	$12.55
OLIVETTI	ET221	DW	132	16	N	FULL	F	NO	P	2	$2650	$22.17
ITOH	F10-40P	DW	132	40	Y	FULL	F	NO	P	2	$1950	$16.31
ITOH	F10-40S	DW	132	40	Y	FULL	F	NO	S	2	$2190	$18.32

NOTES: The following symbols are used:

-	TYPE+	DM = DOT MATRIX. DW = DAISYWHEEL
-	BI DIRECTIONAL	Y = YES. N = NO
-	LOWER CASE	FULL - means Lowercase descenders go below line ND - means Lowercase descenders do not go below line
-	PAPER FEED	F - means Friction Feed T - means Tractor Feed F/T - means both Friction and Tractor feed included in the price
-	INTERFACES	P = PARALLEL (Centronics). S = SERIAL (RS232)
-	FREIGHT	1 - Add $10 for road freight anywhere in Australia 2 - Add $20 for road freight anywhere in Australia

Note Prices subject to change without notice. Prices quoted include Sales Tax at the 17.5% rate.

Call or write for more details.

NOTE!!! NEW LOW PRICES FOR PRAXIS 30 AND 35 PRINTER/TYPEWRITERS

Finance to approved customers is available through "ESANDA" Adelaide. Weekly payment figures shown are approximate only and are calculated after deducting 10% deposit from the selling price, using an effective interest rate of 25.98% over a period of 36 months.

-0000000000-

EDITOR ASSEMBLER PLUS NOW ON DISK!!!!!!

The disk version of Editor/Assembler Plus does everything that the cassette version does with the added speed and convenience of disk drives. the best news is the price, only $39.50 plus $1.20 p.&p.

-0000000000-

DISK OPERATING SYSTEMS FOR TRS-80/SYSTEM 80 COMPUTERS

You can increase your programming productivity, the execution speed and 'user friendliness" of your programs by using an enhanced Disk Operating System (DOS). MICRO-80 recommends DOSPLUS and NEWDOS 80 according to your requirements and experience.

USERS REQUIREMENTS	RECOMMENDED DOS	PRICE	ORDERING INFORMATION
Single-sided Disk Drives, Economy, First-Time User (requires TRSDOS & DISK BASIC MANUAL to supplement DOSPLUS MANUAL.	DOSPLUS 3.3	$ 99.95	Specify Model 1 or Model 3. If Model 1 whether single or double density.
Single or Double-sided Disk Drives, any track count 5 inch or 8 inch. First-time or experienced user wanting Fuss-Free, Bug-Free easy to understand, but very powerful DOS which support variable length records up to 255 Bytes long with stand alone manual. High degree of compatability with TRSDOS.	DOSPLUS 3.4 Highly Recommended	$149.95	Specify Model 1 or Model 3. If Model 1 whether single or double density
Single or Double-sided single or double density disk drives, any track count. 5 inch or 8 inch. Experienced user who has already used TRSDOS and understands the manual. Requires the most powerful DOS available and is prepared to learn the somewhat complicated Syntax. Requires flexible and powerful file handing in BASIC including variable length records up to 4096 Bytes long. Definitely not for the Beginner.	NEWDOS 80 Version 2.0	$169.00	Specify Model 1 or Model 3

NEWBASIC $99.95 PLUS $1.20 P.&P.

BASIC is the programming language used on most microcomputers. One of its main limitations is its unstructured nature which not only leads to untidy and complicated code but also allows very little portability of code from one program to another. NEWBASIC overcomes this limitation by adding PROCEDURE CALLS and enabling you to define BLOCKS thus localising parts of your program yet enabling you to pass parameters to the remainder of the program. With NEWBASIC loaded on top of your BASIC interpreter, you have the familiarity and interactive nature of BASIC with many of the advantages of PASCAL. NEWBASIC adds the following facilities to your interpreter.

COMMANDS & FUNCTIONS

BREAK	lets you program commands for breakpoints.	0.3K
CALL	now you have procedures and sub-programs in BASIC.	*
CONT	continue after a break by just pressing enter.	0.1K
DEF BLOCK	localise parts of your programs yet pass parameters.	*
DEF END	end of a BLOCK, FUNCTION, or PROCEDURE.	*
DEF FUNCTION	start of a multi-line function.	**
DEF PROCEDURE	start of a CALLed procedure.	*
FIELD @	point strings at any part of memory.	0.1K
&FIND	find strings very quickly, anywhere in memory.	0.5K
&FN	access to multi-line functions	**
MERGE	Very speedy loading of programs	***
MOVE	copy memory anywhere, fill it with anything, fast	0.3K
PLUG	chain + pack parts of your program, keep running.	***
RESERVE	reserve and release protected memory as you run.	0.3K
STRINGS	extend and reduce string space when you want to.	0.2K
TIME	measure the time taken by any lines in your program.	1.0K

NewBasic has a 2.9K mandatory root.
* 5.0K in total for blocks.
** 0.5K for functions in addition to blocks.
*** 1.0K for segmented overlaying.

NEWBASIC requires a single disk drive TRS-80 1 or 3 with at least 32K of RAM using TRSDOS, NEWDOS, or NEWDOS 80 Versions 1 or 2.

-0000000000-

SOUNDS.

I was most impressed at the sound possibilities that the machine had at its disposal. At the heart of the sound lay the idea to add audio information to the TV signal, thus enabling the TV speaker to be used for the generation of sounds.

Two distinct methods of using sound are possible - they can be generated from BASIC by the computer, or they may come from the cassette recorder. The latter idea allows instruction on tape to be called into use. The command AUDIO is used to output the cassette to the television speaker; on and off are used as control words. In addition, MOTOR controls the tape motion of the tape recorder.

Musical sounds and tones are generated in BASIC by the commands SOUND and PLAY. Sound is very easy to control and master by these commands. SOUND has considerable potential in providing simple effects within an educational program, whereas PLAY is far more powerful and whole sound sequences can be built up by strings.

COLOUR BASIC.

Tandy Colour BASIC, and Extended BASIC are both written by MICROSOFT and therefore have a lot in common with the many MICROSOFT BASICs available. Earlier in the review we have looked particularly at the Colour and graphics features, and it would not be right to spend too much time and space on the basic BASIC of micro-computers that are now so well known.

PROGRAM LOADING AND SAVING.

I have only used the normal Tandy computer cassette recorder during the period that I used the Colour computer. I found it reliable and effective as long as one remembered to keep it clean. The cassette interface transfers data and programs at 1500 baud. As with other TRS-80 computers, motor control was provided. As you can imagine, I found the 1500 baud rate fast compared with the 500 of the Model I, and it was a real improvement as far as program loading was concerned.

Programs can be saved to tape with names of up to eight characters in length. Normal TRS-80 CLOAD and CSAVE are present for loading and saving programs, but there is no CLOAD? for verifying programs saved to tape. Machine language programs are loaded by the CLOADM instead of SYSTEM.

One of the features that make the TANDY colour computer stand out is the facility to use ROMPAKS. We are all aware of the use of games packs on machines like the Atari. They have been acclaimed as a most successful means of inputting programs. The Sorcerer has proved again the advantage of this method of entering programs. So now the TRS-80 Colour machine has this possibility. ROMPAKS certainly make for ease of loading programs. It is difficult to imagine anyone not being able to manage this loading method.

INSTRUCTIONS AND THE MANUALS.

The operation manual is a clear, easy to read and follow, guide to connecting the system together. It includes a useful section on what to do if the computer fails to operate correctly.

The two manuals - GETTING STARTED WITH COLOR BASIC and GOING AHEAD WITH EXTENDED COLOR BASIC are written in the well-known and loved (or hated) Tandy style. I personally like the humorous style used and other colleagues who came to grips with Colour BASIC with the TRS-80 Colour computer and these manuals thought a lot of them. The Getting Started manual assumes that you know next to nothing about computing and takes you through a simple and effective course. I felt that both manuals contained plenty of examples, guides, and tasks to be undertaken. Both manuals also contained, for the more experienced users, the necessary appendices summarizing keywords, ASCII codes etc.

CONCLUSION.

There is no doubt that the TRS-80 Colour computer is a powerful machine that has tremendous potential. It makes the most of its colour and its graphics and certainly functions well. I believe that it is still reasonably priced when compared with the other colour machines on the market. Already, in the USA, a full range of accessories is becoming available, in the same way as they have done for other TANDY machines. A disk system is already in the UK and official memory expansion is coming.

What potential then has the Colour computer to offer as an educational machine? Having had the TRS-80 Colour up and running for a month in school, it is already obvious that the machine appeals to the children. A coloured display with the quality of graphics available moves us into a new arena. We have yet to fully come to terms with the potential available to us. I am sad, though, that the Tandy Colour does not capitalize on the extensive range of software available on the Model I and III. However, Tandy intends to reproduce the many educational packages it has marketed for other machines. Let's hope that the schools and colleges really do realize the power of this machine and begin to harness it.

- 0000000000 -

INPUT/INPUT continued

From: Mr. N. Joyce, Wauchope, N.S.W.

I am having a problem with an article which was in Issue 21 (August 81) page 6. The article was "Reversing the role of PRINT and LPRINT statements on the TRS-80". I have a System 80 and find that the first instruction works - that is 'POKE16422,88:POKE16423,Ø4 ENTER' and also to return to normal mode POKE16422,141:POKE16423,Ø5 works but to direct all PRINT statements to the line printer instead of the screen, 'POKE 16414,141:POKE16415,Ø4 ENTER and 'POKE 16414,88:POKE 16415,Ø4 (to return to normal mode) does not work. When tried the computer had to be switched off to return it to normal. Could you please help me with this problem.

(Whoops! It is not surprising that you are having a problem Mr. Joyce - there was a typographical error in the article referred to. To direct all PRINT statements to the line printer instead of the screen, you should use POKE 16414,141:POKE 16415,Ø5 ENTER not..... POKE 16415,Ø4 as printed. All other POKEs in the article are correct. Please accept our apologies for any inconvenience caused -Ed.)

* * * * SOFTWARE SECTION * * * *

***** ATOMIC TABLES - PEACH and CC *****

This program was originally published in issue 24 for the TRS-80 Model I. It has now been modified to run on both the Peach and the TRS-80 Colour Computer. Separate listings for each computer are at the back of the magazine. Note that the colour computer listings use the symbol ^ in place of the ↑ (up arrow).

This program not only aims to teach, but also can act as a reference to students studying chemistry, for it shows not only the name, symbol and element atomic number, but also gives the group or series, and accurately gives the electron shells of the said atom. As the movements of atoms in the different shells (or energy levels) is not always easily predictable, this program is of particular usefulness. It is also capable of telling either of name, number or symbol from any of those given. The program uses one large data array to find all three particulars of an atom. Once it finds them it prints those which are required by the user. When it completes one question display, it searches its data for the required element, during which time the user has to try to think of the element. If the user wants only to find the data, there will be a short wait before the answer is displayed.

- 0000000000 -

***** MULTIPLE REGRESSION ANALYSIS - PEACH and CC *****

This program was originally published in issue 24 for the TRS-80 Model I. It has now been modified to run on both the Peach and the TRS-80 Colour Computer. Separate listings for each computer are at the back of the magazine. Note that the colour computer listings use the symbol ^ in place of the ↑ (up arrow).

Multiple regression analysis is used to test the relationship between a dependent variable (Y) and a number of independent variables (X1,X2,X3.....XN).

This routine allows multiple regression analysis between the independent variable Y and two dependent variables X1 and X2, in accordance with the model:-

$$Y = A + BX1 + CX2.$$

The routine calculates the coefficients for the multiple regression equation, the mean values of X1, X2 and Y, the coefficient of multiple regression (an indication of closeness of fit, where 1 is perfect), and the percentage variation in Y due to X1, X2, and X1 and X2 jointly. In addition, a residual table can be printed on command and estimations of Y computed for entered values of X1 and X2.

The routine will accept up to 100 data sets, which are verified and can be edited following entry.

- 0000000000 -

***** SPACE COMMANDER L1/4K (C) S. Williamson *****

This is a game in which you must buy and sell Battlestars at fluctuating prices. The objective is to avoid capture of all your Jet Fighters. This is achieved by purchasing more Battlestars. Battlestars are paid for with Jet Fighters. When the program asks you how many Battlestars you want to buy, reply with a number. The program will then check to see if you have enough Jet Fighters to buy that many Battlestars. If you reply 0 (zero), the program will ask you how many Battlestars you wish to sell.

After you reply, the program will credit you with the correct number of Jet Fighters, i.e. if Battlestars cost, say, 5 Jet Fighters each and you sell 5 Battlestars, you would be credited with 25 Jet Fighters.

The program then asks you how many Battlestars you want put into action. After you reply, the program displays the results of the battle and once again asks how many Battlestars you wish to buy.

- 0000000000 -

***** SPACE GALAXY LI/4K by Br. P. Van Eeken *****

In this space war game a mathematical problem is posed by the program. Before the game can be played the program requires that the correct answer to the problem is given. With this program your children can learn and have fun at the same time. After the question has been answered correctly the game itself will commence. Three guns are displayed at the bottom of the screen. The gun that is to fire is selected by repeatedly pressing the ENTER key. An asterisk "*" is displayed underneath the gun that is active. Each time ENTER is pressed the asterisk will move one position to the left, or if hard left it will then move one position right for each time that ENTER is pressed. The Invaders move from right to left across the screen at different heights. To fire a gun to the right, press the right arrow key; to fire to the left, press the left arrow key, and to fire directly upwards press the space bar.

Your controls are active any time that the white bar at the top right of the screen is visible. If the program is not ready for your input at any time, it will display the word "WAIT" in place of the white bar. While this is displayed the program will not react to the control keys.

The invaders fight back. When all of your guns have been destroyed the game is over. High scoring players are asked to enter their name at the end of the game. The score and the name are then displayed during other games until such time as that score is beaten.

- 0000000000 -

***** READ-A-LINE L2/16K (C) A. Cartwright *****

I have often had late nights (and early mornings) looking for typo errors in some program or other that I have typed in from a magazine (not always MICRO-80). Whilst trying to list 16 lines at one time on the screen, the lines that I want would zoom through and I would hit the BREAK key trying to push it through the table. Of course, I could type LIST No. - No. but in the early hours it's too much trouble. Then with a screen full of lines to be checked, I keep getting lost looking from the screen to listing and back again. Now this makes for slow program checking.

So I decided that it was about time that something was done about it, and not being called Edwin Paay, I would have write a program in BASIC.

First I looked at how Foxy-80 stored my programs in memory, so I wrote a complicated program to help me find out, like:

```
1Ø PRINT A+B
```

Now it happens that in Level 2 BASIC, memory addresses 16548 and 16549 point to the 'start of BASIC program', 16548 being the least significant byte (LSB), and 16549 being the most significant byte (MSB). If you PEEK at these addresses, you will find that they hold 233 and 66 respectively. Doesn't mean much until you apply the Foxy-80 maths formula LSB + MSB * 256 = Start of BASIC Program location or 233 + 66 * 256 = 17129 (a bit complicated but Foxy knows what to do).

Typing: FORX=17129TO17139:PRINTX;"=";PEEK(X):NEXT the following streamed onto the screen:

```
17129 = 242
1713Ø = 66
17131 = 1Ø
17132 = Ø
19133 = 178
17134 = 65
17135 = 2Ø5
17136 = 66
17137 = Ø
17138 = Ø
17139 = Ø
```

Having PEEKed at my program I can see that addresses 17129 and 1713Ø point to the start of the next line. Don't ask me why, they just do! So applying the Foxy-80 formula (242+66*256) I come up with 17138. Now seeing as my program is only one line, looking at locations 17138 and

17139 I see that they are set to Ø, which means that there is nothing to follow (end of program). Anyway, getting back to the start, the next locations 17131 and 17132 hold the line number (using the formula again). Then we come to the program statements themselves starting at 17133.

Programs are composed of ASCII codes. Looking at my Level 2 Manual reveals that the ASCII codes 178 = PRINT65="A",2Ø5="+",66="B". The next location (17137) is Ø which is an end of line marker, the next location is 17138 and like my programs, I have already jumped there and jumped back again. Now thinking about it, I suppose I have just found out where the five bytes of line overhead have gone to, 2 bytes for the next line address, 2 for the line number and 1 for the end of line marker. Having got this far, the thing to do is to pick out the statements and miss the addresses. That is to say, use the values in locations 17131 through 17137 inclusive, convert 17131 and 17132 using the now famous Foxy formula for the line number and print the characters of the ASCII code for the rest.

My first attempt worked really well and I thought that I had done a good job (so easy too!). Alas, I tried listing CWORD (MICRO-80 issue 14). Well, I'm afraid Foxy and I crashed in double-quick time. It went something like this:

```
1ØCOMPUTER**CWORD BY L. G. RHEDEY**
```

Now then I thought, I know it's a computer stupid but that is not what Mr. Rhedey wrote in line 1Ø. It should be REM and not COMPUTER. Back to the drawing board!!

It took me a long time to realise that my super program was not reading its own data, but reading data from the program it was looking at. I was getting funny statements like

'2ØSUMPALOVEB:COMPUTERTHIS IS RIGHT' so data reading had to go. This is why I have used extra memory dimensioning instead of reading data lines. The next thing was, after reading 6Ø or 7Ø lines I would come across a typo error, so I would use the BREAK key then edit the line and then start listing from scratch.

By the time I had listed 6Ø or 7Ø lines again and got back to where I had left off, I had taken more time than I would have taken not using the program at all!

Back to the drawing board! (I wonder if professional programmers have the same problems?)

I had to find a way of saving where I was without losing the data when editing, so I decided to try the suicidal way of POKEing the location in the Start of BASIC Program Pointers (these are not set to zero when one enters Edit mode). Typing with fingers crossed (which is difficult) I found that it worked, so that's the way it is. After a lot of typing and retyping, I have finished a program which I can break out of, edit a line, and then start back in from where I left off. I have called it READ-A-LINE (after all the trouble I had with data reading!)

I shall explain a few lines.

64999 Stops any program crashing into READ-A-LINE.
65ØØØ Is the entry point, it also dimensions X$ by 127.
65Ø8Ø Clear the screen and saves the start of program pointer in P1 & P2, P = the location of the current statement to be read.
65Ø9Ø Is a test for the end of program. If yes then it restores the start of BASIC pointer.
651ØØ Tests for end of line.
6511Ø Tests for function/statement (e.g. PRINT, FOR, NEXT) and prints them.
6512Ø Prints the ASCII character.
6513Ø Increments P (next location).
6514Ø Pause at end of line routine. Also puts a new address into start of BASIC pointer, then P1 & P2 are updated to the next line address, and P is once again set to point to the next statement.
6515Ø Calculates the line number, clears the screen from location 448.

How to use READ-A-LINE.

1. Load READ-A-LINE into memory.
2. Type in your program.
3. If you need to check your typing against the listing then type RUN65ØØØ and press ENTER/NEWLINE. READ-A-LINE will then proceed to list your program one line at a time. If you need to change a line, just press the BREAK key, then enter Edit mode by the usual way. After the changes, type RUN 65ØØØ, press ENTER/NEWLINE and away you go again from where you pressed the BREAK key.
4. When using READ-A-LINE, it is important that you end the program correctly otherwise the pointers will be wrong and, at best, your program will not RUN at all but your Foxy-80 could bomb out leaving you with a pretty pattern on your screen, and locked up tight.
5. It's always a good idea to SAVE a program often when typing one in but before making your final SAVE be sure to delete lines 64999 through 6515Ø.

I hope that you will find READ-A-LINE useful.

- 0000000000 -

***** FLIP L2/16K by R. Williams and A. Quick *****

Flip is a habit-forming game devised by Ross Williams whilst bored at a football game. He had a number of 20 cent coins and arranged these in a four by four grid. By starting with a random pattern and turning the coins over one by one, always turning the coin adjacent to the last turned coin and never going back the way he had come, he devised the game "Flip", the object being to start in the bottom left corner and to make the grid all heads or all tails.

The program uses a four by four graphics grid. The player starts in the bottom left corner and may move up or right. Later moves may go in any of the four directions, provided that you don't try to move back to where you just came from.

Each time you move, the position into which you move changes "colour". You then have to get all of the positions the same "colour" (white or black). Moving is done with the following keys:

U - UP
D - DOWN
L - LEFT
R - RIGHT

The only rules: - You cannot go back the way you just came
- You cannot move out of the "grid".

There is a strategy to the game and, once discovered, it can be solved in a few minutes. If you don't know the strategy, a game can take quite some time! The smallest known number of moves to solve a game is 7. Others have been known to take more than 200. The display shows the number of moves made, the current position as a pair of co-ordinates, and the direction in which you last moved. You may not move off the grid, or back where you have come from, and an indication is given on the screen if you try.

The grid is numbered:

A4 B4 C4 D4
A3 B3 C3 D3
A2 B2 C2 D2
A1 B1 C1 D1

```
HIT ANY KEY TO CONTINUE

     NOTES ON 'FLIP'
YOU ALWAYS START IN THE
BOTTOM LEFT CORNER (A,1)

WHENEVER YOU ENTER A NEW
SQUARE, IT CHANGES COLOUR!
YOU CANNOT MOVE BACK TO A
POSITION YOU HAVE JUST COME
FROM. I.E RIGHT, THEN
LEFT, ISN'T OK.

ENTER 'Q' TO QUIT.
```

VIEW OF SCREEN DISPLAY

```
     NOTES ON 'FLIP'
FLIP IS AN ANNOYING GAME...
YOU HAVE TO CHANGE THE 16
SQUARES TO EITHER ALL WHITE
OR ALL BLACK.
THE AIM IS TO DO THIS IN
THE LEAST NUMBER OF MOVES.

YOU MOVE WITH THE KEYS:
      U - UP
      D - DOWN
      L - LEFT
      R - RIGHT
```

- 0000000000 -

***** KEWORD UTILITY vers. 1.5 by Mr. F. Ellett *****

Keword provides single key entries AND single key functions. It is actually a group of 6 utilities linked together with a command table using the Z-80's transparent control code facility.

VIEW OF SCREEN DISPLAY

```
* * KEYWORD UTILITY (VER 1.5) (C) F.ELLETT SEPT 1981 * *
FOR ALL FUNCTIONS PRESS SHIFT-DOWN ARROW (CTL) & ALPHA CHR.
A= BREAK       B= B LAZY SAVE  C= CLONE        D=DISABLE KWD
E= ENABLE KWD  F= FIX (U) KEY  G= GET TAPE     N= NUMBER
O= CURSOR OFF  P= PRINT SCREEN Q= QUIT SOUND   R= CURSOR ON
S= S-80 SOUND  T= TRS SOUND    V= VIDEO TEXT
SHIFT KEY WORDS :-
A=ASC(  B= CLS    C= CSAVE"A" D= DATA  E= ELSE     F=FORI=1TO
G= GOTO H=GOSUB   I= INPUT    J= ","   K=IK$=INKEY$ ETC.
L= LEN( M= MID$(  N= NEXT     O= POKE  P= PEEK(    Q= CHR$(
R= RETURN         S=STRING$(  T= THEN  U=OUT254,255
V= VAL( W= STR$(  X= RIGHT$(  Y= RND(  Z= LEFT$(
LIST: HOLD (S) TO SLOW.  PRESS (W) TO HALT SCROLL
ANY KEY FOR NEXT LINE. (L) TO RESUME LIST
READY
>
```

Control codes are produced by pressing Shift-Down Arrow (CTRL on System 80) and one Alphabet key all at the one time. For instance, Shift-Down Arrow-A produces the code 01Hex (which is the control code for BREAK) whilst Shift-Down Arrow-Z produces control code 1AHex or decimal 26.

Of the 26 control codes generated in this manner there are 15 unused by the system and available to the user. They are B to G and N to V.

This program uses most of them. The routines can all stand alone so do not be afraid to hack and delete those ideas which do not appeal to you. The control codes will return:-

A. Unaltered. Will return the BREAK code it represents.
B. B'Lazy Save. A real peace-maker. This little gem will make two CSAVE's with 4 second leaders completely unattended.
C. CLONE. Who hasn't had his favourite "copy protected" "Super Klingons" spiked or erased by eager helpers? CLONE will duplicate tapes that will not otherwise copy - indeed, tapes too large to fit in the user's memory. It requires two tape recorders. Recorder number 1 holding the tape to be copied, is set in PLAY mode with the Ear and Remote plugs from the computer in place. Recorder number 2 must have the Aux. plug from the computer in its AUX socket. It holds the new tape and must be put in record mode and set in motion before ENTER is pressed, activating CLONE. Recorder 2 must be manually turned off but pressing BREAK during any sort of read will exit CLONE.

   CLONE operates by searching the Source tape until a pulse is detected. A duplicate pulse is immediately created and fed out through the cassette logic via the Aux lead to the copy cassette. Thus, an exact replica is produced Bit by Bit. CLONE's weakness is that if the pulses of the Source tape are stretched and distorted so will be the CLONED copy.
D. Disables the single key words.
E. Enables single key words. The choice of text for each keyword was quite arbitrary and if you are typing the program in using Edtasm the text may be altered to suit your needs.

   To find the end of the keyword text the program checks each character to see if bit 7 is set (lines 1040 to 1060). So be sure to set bit 7 on your last character. For example, the opening bracked Hex28 +80Hex (Bit 7) becomes A8Hex.
F. FIX the user defined key. This allows the user to fix the text in the shift U to any text required. Up to a limit of 40 charactders. This key can be altered as often as required as your programming needs change. Its default value is "OUT 254,255". When typing this line in (line 1820) be sure to pad out to 40 characters with blank spaces.

   Both BREAK and ENTER (NEW LINE) terminate text input.
G. GET TAPE ADDRESSES. If you have Edwin Paay's ROM Reference Manual you will be aware that each tape type has a different header. BASIC has 3 D3's followed by a file letter. EDTASM has only 1 D3 followed by a 6 letter name, whilst SYSTEM's header is a U then a 6 letter name. GET TAPE reads an unknown tape and prints the type and file name to screen. If the tape is a SYSTEM tape the program will read the entire tape and print out Start End & Entry points. Note: the contents are not stored in memory; GET TAPE's only aim is to identify unknown tapes. BREAK will terminate during read.
N. NUMBER HEX TO DECIMAL. Number is the companion tape to GET TAPE. It will return the decimal value for any hex input, plus the decimal value for the L.S.B. and M.S.B. for use in BASIC pokes.
O. OFF. Disables the flashing cursor by placing a return token in the first byte of the flashing cursor routine.
P. Print screen to line printer. If printer is unavailable the command is ignored. (A CALL to 0501H in line 3360 returns a zero in A register if printer is ready).

   The LPRINT command is completely transparent to any program running and can be used without ill effect at any time the keyboard is being scanned. As our printer cannot support graphic characters all non alpha characters are printed as the hatch symbol.

Keybounce. We have used a tone output to the cassette port as our key scan delay. The following codes are used:

Q. QUIET. Loads A register with 00H. The delay works but nothing is output. This is the default condition. Q should be typed before doing a CSAVE.
S. SYSTEM-80 Enables cassette port 2 (OUT 254,255) and sets the out bytes to 05H and 06H.
T. TRS-80 Sets the out bytes to 01 and 02H that is, signal out, motor off. Tandy cassette port 1.
R. REPEATING CURSOR. Whilst a flashing cursor can be an irritation it is a great reminder that the keyboard is waiting input.

Command R enables the cursor by making sure the first byte of the routine contains a null value. The flashing cursor operates by setting a counter (FTIME line 8070) and decrementing it each time the keyboard is strobed. When the counter reaches zero the character at cursor position (4020H contains the current cursor position) is toggled between the normal cursor dash and the graphic block by XORing its value with D3H then FTIME is reset to 0300H. The flash rate can be changed by altering the value loaded into HL in line 8020.

ENTERING THE PROGRAM.

A Hex dump of the 16K version of the program is listed on pages 33 and 34. Unfortunately, the EDTASM source code is too long to publish in the magazine. A photo-copy of the well-documented source code is available for $5.00. The source code is recorded on both the cassette and disk editions. Please note that the commented source code is too long to load into a 16K machine. To enter the program listed here, you will need a monitor program which does NOT use the area 7380 Hex to 7FFF Hex. ZMONL from the MICRO-80 Software Library - Vol.1 would be suitable. Use the Edit Memory function starting at 7380 Hex and enter the 3193 bytes shown then punch a tape having the following parameters:

START=738Ø END=7FF8, ENTRY=738Ø, NAME=KEWORD

SINGLE KEY ENTRIES.

With Keywords enabled, pressing SHIFT followed by an alpha character will enter the keyword assigned to that key, i.e. press SHIFT K and the keyword IK$=INKEY$ will be displayed. A user defined shifted key entry is provided for, (SHIFT U), see Keyword Function - FIX.

INTERRUPTS.

For those wishing to run on Disk, Interrupts are disabled during initialisation. If this does not suit, you can then use the BASIC command CMD" R" and CMD"T". Our TRS-80 with interface and no disk will hang up on a disable interrupt instruction. If this is your case, null out the DI instruction in line 7Ø.

To run on Disk, assemble to leave 50-60 bytes at top of memory. Load under DOS command LOAD, then load BASIC. Answer MEM SIZE with 1 byte less than ORG address. Then enter by typing System then slash and the decimal value of ORG address.

LIST DELAY.

The program intercepts the list routine at the DOS address 41DFH. Holding S key will slow list scan to about 1 line a second. Pressing the W key (wait) will halt the scan and then any other key will display the next line. The L key resumes normal list.

HOW TO LOAD FROM CASSETTE.

For those who subscribe to the cassette edition of MICRO-80, follow the procedure below to load the program from tape.

Answer MEMORY SIZE (READY?) with 29567 press ENTER/NEWLINE Type: SYSTEM
Answer the prompt with:-

KEWORD

After the program has loaded, answer the second prompt with:- /29568

The KEWORD Menu will then be displayed.

HOW TO LOAD FROM DISK.

If you subscribe to the disk edition of MICRO-80 or have transferred the program to disk, follow the procedure below to load KEWORD (48K version).

DISTRIBUTION DOS

From DOS, type:-

LOAD KEWORD/CMD press ENTER/NEWLINE
BASIC,62335 press ENTER/NEWLINE
SYSTEM press ENTER/NEWLINE

Answer the prompt with /62336 press ENTER/NEWLINE.

TRSDOS

From DOS type:-

LOAD KEWORD/CMD press ENTER/NEWLINE
BASIC press ENTER/NEWLINE
Answer HOW MANY FILES with press ENTER/NEWLINE
Answer MEMORY SIZE with 62335 press ENTER/NEWLINE
SYSTEM press ENTER/NEWLINE
Answer the prompt with /62336 press ENTER/NEWLINE

***** ATOMIC TABLES *****
EXT. BASIC COLOUR COMPUTER

```
10 DIMD$(312)
20 'THE ELEMENTS  BY DAVID BALAI
C, 194/125 NAPIER ST., FITZROY
30 '3065.   COMPLETED 31/1/81.
40 CLS:PRINT@10,"THE ELEMENTS";:
PRINT@43,"MAIN INDEX":PRINT"
YOU MAY CHOOSE ANY OF THE      F
OLLOWING:":PRINT"A -FOR AN ELEME
NT'S ATOMIC NO.":PRINT"B -FOR AN
 ELEMENT'S SYMBOL":PRINT"C -FOR
AN ATOMIC NO.'S ELEMENT"
50 PRINT"D -FOR AN ATOMIC NO.'S
SYMBOL":PRINT"E -FOR A SYMBOL'S
ATOMIC NO.":PRINT"F -FOR A SYMBO
L'S ELEMENT":PRINT"G -FOR A LIST
 OF ALL RELEVANT       DATA ASSOC
IATED WITH AN          ELEMENT.";
60 A$= INKEY$:IFA$=""THEN60ELSE
IFA$="A"THEN140ELSE IFA$="B"THEN
90ELSE IFA$="C"THEN70ELSE IFA$="
D"THEN190ELSE IFA$="E"THEN210ELS
E IFA$="F"THEN260ELSE IFA$="G"TH
EN310ELSE60
70 CLS:PRINT@320,"PLEASE ENTER A
TOMIC NO.";:INPUTA:GOSUB590:FORN
=1TOA*3+2:READD$:NEXT:CLS:PRINT@
320,"THE ELEMENT FOR"A"IS "D$;:P
RINT@481,"HIT ANY KEY TO CONTINU
E";:RESTORE
80 IF INKEY$ =""THEN80ELSE20
90 CLS:PRINT@450,"PLEASE ENTER E
LEMENT";:INPUTF$:GOSUB590:FORN=1
TO312:READD$(N):IFD$(N)=F$THEN12
0ELSE NEXT
100 GOSUB160
110 GOTO90
120 CLS:N=N+1:READD$(N):PRINT@32
0,"THE SYMBOL OF "F$" IS "D$(N):
PRINT@481,"HIT ANY KEY TO CONTIN
UE";:RESTORE
130 IF INKEY$ =""THEN130ELSE20
140 CLS:PRINT@450,"PLEASE ENTER
ELEMENT";:INPUTS$:GOSUB590:FORN=
1TO312:READD$(N):IFD$(N)=S$THEN1
70ELSE NEXT:GOSUB160
150 GOTO140
160 CLS:RESTORE:PRINT@288,"SORRY
, TRY AGAIN";:FORJ=1TO870:NEXT:R
ETURN
170 CLS:PRINT@320,"THE ATOMIC NO
. OF "S$" IS "D$(N-1);:PRINT@481
,"HIT ANY KEY TO CONTINUE";:REST
ORE
180 IF INKEY$ =""THEN180ELSE20
190 CLS:PRINT@320,"PLEASE ENTER
ATOMIC NO.";:INPUTA:GOSUB590:FOR
N=1TOA*3+3:READD$:NEXT:CLS:PRINT
@320,"THE SYMBOL OF"A"IS "D$;:PR
INT@481,"HIT ANY KEY TO CONTINUE
";:RESTORE
200 IF INKEY$ =""THEN200ELSE20
210 CLS:PRINT@320,"PLEASE ENTER
SYMBOL";:INPUTS$:GOSUB590:FORN=1
TO312:READD$(N):IFD$(N)=S$THEN24
0ELSE NEXT
220 GOSUB160
230 GOTO210
240 CLS:PRINT@320,"THE ATOMIC NO
. OF "S$" IS "D$(N-2);:PRINT@481
,"HIT ANY KEY TO CONTINUE";:REST
ORE
250 IF INKEY$ =""THEN250ELSE20
260 CLS:PRINT@320,"PLEASE ENTER
SYMBOL";:INPUTS$:GOSUB590:FORN=1
TO312:READD$(N):IFD$(N)=S$THEN29
0ELSE NEXT
270 GOSUB160
280 GOTO260
290 CLS:PRINT@320,"THE ELEMENT F
OR "S$" IS "D$(N-1);:PRINT@482,"
HIT ANY KEY TO CONTINUE";:RESTOR
E
300 IF INKEY$ =""THEN300ELSE20
310 CLS:PRINT@289,"PLEASE ENTER
EITHER ATOMIC NO., SYMBOL OR ELE
MENT";:INPUTX$:GOSUB590:IF VAL(X
$)< >OTHENN= VAL(X$):IFN>103THEN
310ELSE340ELSEJ= LEN(X$)
320 FORN=1TO312:READD$(N):IFD$(N
)=X$THEN330ELSE NEXT:RESTORE:GOT
O310
330 RESTORE:N=N/3:N= INT(N):IFJ<
=2THENN=N-1
340 FORNN=1TON*3+1:READD$(NN):NE
XT:CLS:PRINT@7,"ATOMIC NO.:"N;:P
RINT@32,"ELEMENT:  ";:READD$(NN):
PRINTD$(NN)",  SYMBOL:  ";:NN=NN+1
:READD$(NN):PRINTD$(NN)".":PRINT
@66,"NO. OF PROTONS:"N;:PRINT@10
3,"ELECTRONS:"N","
350 PRINT@367,N"P";:GOSUB390:PRI
NT@416,"GROUP:  "GN$;:PRINT@334,
".-K="K"-.";:PRINT@302,".-L="L"-
.";:PRINT@270,".-M="M"-.";:PRINT
@238,".-N="P"-.";:PRINT@206,".-O
="Q"-.";:PRINT@174,".-P="R"-.";:
PRINT@142,".-Q= 0 -.";:RESTORE
360 PRINT@480,"HIT ANY KEY TO CO
NTINUE";:K=0:L=0:M=0:R=0:P=0:Q=0
370 IF INKEY$ =""THEN370ELSE20
380 DATA541,544,
390 GN$="TRANSITIONAL ELEMENTS":
IFN=30RN=110RN=190RN=370RN=550RN
=87THENGN$="I"ELSE IFN=40RN=120R
N=200RN=380RN=560RN=88THENGN$="I
I"
395 IFN=50RN=130RN=310RN=490RN=8
1THENGN$="III"ELSE IFN=60RN=140R
N=320RN=500RN=82THENGN$="IV"ELSE
 IFN=70RN=150RN=330RN=510RN=83TH
ENGN$="V"
400 IFN=80RN=160RN=340RN=520RN=8
4THENGN$="VI"ELSE IFN=90RN=170RN
=350RN=530RN=85THENGN$="VII"
405 IFN=20RN=100RN=180RN=360RN=5
40RN=86THENGN$="VIII OR 0 - NOBL
E GASES"ELSE IFN=1THENGN$="HYDRO
GEN":K=0:R=0:Q=0
410 IFN> =90THENGN$="ACTINIDE SE
RIES"ELSE IFN> =58ANDN< =71THENG
N$="LANTHANIDE SERIES"
420 IFN> =2THENK=2:N=N-2ELSEK=N:
GOTO480
430 IFN> =8THENL=8:N=N-8ELSEL=N:
GOTO480
440 IFN> =18THENM=18:N=N-18ELSEM
=N:GOTO480
450 IFN> =32THENP=32:N=N-32ELSEP
=N:GOTO480
460 IFN> =32THENQ=32:N=N-32ELSEQ
=N:GOTO480
470 IFN> =18THENR=18:N=N-18ELSER
=N
480 N=K+L+M+P+Q+R:IFN=19THENM=M-
1:P=1:GOTO580
```

```
490 IFN> =20ANDN< =30THENM=M-2:P
=2:GOTO580
500 IFN=37THENP=P-1:Q=1:GOTO580
510 IFN=61THENP=P-1:Q=2:GOTO580
520 IFN=62THENQ=2:GOTO580
530 IFN> =38ANDN< =62THENP=P-2:Q
=2:GOTO580
540 IFN=69THENQ=Q-1:R=1:GOTO580
550 IFN=93THENQ=Q-1:R=2:GOTO580
560 IFN=94THENR=2:GOTO580
570 IFN> =70ANDN< =94THENQ=Q-2:R
=2:GOTO580
580 RETURN
590 CLS:PRINT@265,"COMPILING DAT
A";:RETURN
600 DATA1,HYDROGEN,H,2,HELIUM,HE
,3,LITHIUM,LI,4,BERYLIUM,BE,5,BO
RON,B,6,CARBON,C,7,NITROGEN,N,8,
OXYGEN,O,9,FLOURINE,F,10,NEON,NE
,11,SODIUM,NA,12,MAGNESIUM,MG,13
,ALUMINIUM,AL,14,SILICON,SI,15,P
HOSPHORUS,P,16,SULPHUR,S,17,CHLO
RINE,CL,18,ARGON,A
605 DATA19,POTASSIUM,K,20,CALCIU
M,CA
610 DATA21,SCANDIUM,SC,22,TITANI
UM,TI,23,VANADIUM,V,24,CHROMIUM,
CR,25,MANGANESE,MN,26,IRON,FE,27
,COBALT,CO,28,NICKEL,NI,29,COPPE
R,CU,30,ZINC,ZN,31,GALLIUM,GA,32
,GERMANIUM,GE,33,ARSENIC,AS,34,S
ELENIUM,SE,35,BROMINE,BR,36,KRYP
TON,KR,37,RUBIDIUM,RB
620 DATA38,STRONTIUM,SR,39,YTTRI
UM,Y,40,ZIRCONIUM,ZR,41,NIOBIUM,
NB,42,MOLYBDENUM,MO,43,TECHNETIU
M,TC,44,RUTHENIUM,RU,45,RHODIUM,
RH,46,PALLADIUM,PD,47,SILVER,AG,
48,INDIUM,IN,49,CADMIUM,CD,50,TI
N,SN,51,ANTIMONY,SB,52,TELLURIUM
,TE
630 DATA53,IODINE,I,54,XENON,XE,
55,CAESIUM,CS,56,BARIUM,BA,57,LA
NTHANUM,LA,58,CERIUM,CE,59,PRASE
ODYMIUM,PR,60,NEODYMIUM,ND,61,PR
OMETHIUM,PM,62,SAMARIUM,SM,63,EU
ROPIUM,EU,64,GADOLINIUM,GD,65,TE
RBIUM,TB,66,DYSPROSIUM,DY,67,HOL
MIUM,HO
640 DATA68,ERBIUM,ER,69,THULIUM,
TM,70,YTTERBIUM,YB,71,LUTETIUM,L
U,72,HAFNIUM,HF,73,TANTALUM,TA,7
4,TUNGSTEN,W,75,RHENIUM,RE,76,OS
MIUM,OS,77,IRIDIUM,IR,78,PLATINU
M,PT,79,GOLD,AU,80,MERCURY,HG,81
,THALLIUM,TL,82,LEAD,PB,83,BISMU
TH,BI
650 DATA84,POLONIUM,PO,85,ASTATI
NE,AT,86,RADON,RN,87,FRANCIUM,FR
,88,RADIUM,RA,89,ACTINIUM,AC,90,
THORIUM,TH,91,PROTOACTINIUM,PA,9
2,URANIUM,U,93,NEPTUNIUM,NP,94,P
LUTONIUM,PU,95,AMERICIUM,AM,96,C
URIUM,CM,97,BERKELIUM,BK,98,CALI
FORNIUM,CF
660 DATA99,EINSTEINIUM,ES,100,FE
RNIUM,FM,101,MENDELEVIUM,MV,102,
NOBELIUM,NO,103,LAWRENCIUM,LW
```

***** MULTIPLE REGRESSION ANALYSIS *****
EXT. BASIC COLOUR COMPUTER

```
10 'ROUTINE FOR MULTIPLE REGRESS
ION ANALYSIS
20 '(C) COPYRIGHT 1980
30 'T.R.JONES,
40 CLS:DIMX(3,100),A(6,3),R(6,3)
:P$="#######.####"
50 PRINT"  MULTIPLE REGRESSION A
NALYSIS":PRINT"  ----------------
---------------"
60 PRINT:PRINT:PRINT:PRINT TAB(1
3)"MODEL":PRINT:PRINT TAB(7)"Y =
 A + BX1 + CX2"
70 PRINT@448,"TYPE <ENTER> TO CO
NTINUE ....";:INPUTI$
80 CLS:PRINT TAB(11)"ENTER DATA"
:PRINT:PRINT:PRINT"NUMBER OF DAT
A POINTS - MAXIMUM IS 100"
90 PRINT@172,"*";:INPUTND:IFND<2
ORND>100THEN PRINT@172,"
  ":GOTO90
100 PRINT:PRINT:FORI=1TOND:PRINT
"X1(";I;") =";:INPUTX(1,I):PRINT
"X2(";I;") =";:INPUTX(2,I):PRINT
" Y(";I;") =";:INPUTX(3,I):NEXTI
110 CLS:K=0:PRINT TAB(10)"DATA E
NTERED":PRINT:FORI=1TOND:K=K+1:P
RINT"X1(";I;") =";X(1,I):PRINT"X
2(";I;") =";X(2,I):PRINT" Y(";I;
") =";X(3,I):IFK<10THENNEXTI ELS
E PPRINT@448,"TYPE <ENTER> TO CO
NTINUE ...";INPUTI$:K=0:CLS:NEXT
I
120 PRINT@488,"EDIT DATA (Y/N)";
130 I$= INKEY$:IFI$=""THEN130ELS
E IFI$="Y"THEN140ELSE IFI$="N"TH
EN160ELSE130
140 CLS:INPUT"DATA POINT TO BE E
DITED";I:IFI>ND THEN140ELSE PRIN
T"X1(";I;") =";X(1,I):PRINT"X2("
;I;") =";X(2,I):PRINT" Y(";I;")
=";X(3,I)
150 PRINT"NEW X1(";I;") =";:INPU
TX(1,I):PRINT"    X2(";I;") =";:
INPUTX(2,I):PRINT"    Y(";I;")
=";:INPUTX(3,I):GOTO110
160 CLS:PRINT@265,"COMPUTING ...
."
170 FORI=1TO6:FORJ=1TO3:A(I,J)=0
:R(I,J)=0:NEXTJ,I
180 FORI=1TOND:A(1,1)=A(1,1)+X(1
,I):A(1,2)=A(1,2)+X(2,I):A(1,3)=
A(1,3)+X(3,I):NEXTI:A(2,1)=A(1,1
)/ND:A(2,2)=A(1,2)/ND:A(2,3)=A(1
,3)/ND
190 FORI=1TOND:A(3,1)=A(3,1)+(X(
1,I)^2):A(3,2)=A(3,2)+(X(2,I)^2)
:A(3,3)=A(3,3)+(X(3,I)^2):NEXTI
200 FORI=1TOND:A(4,1)=A(4,1)+(X(
1,I)*X(3,I)):A(4,2)=A(4,2)+(X(2,
I)*X(3,I)):A(4,3)=A(4,3)+(X(1,I)
*X(2,I)):NEXTI
210 A(5,1)=A(3,1)-(A(1,1)^2/ND):
A(5,2)=A(3,2)-(A(1,2)^2/ND):A(5,
3)=A(3,3)-(A(1,3)^2/ND)
220 A(6,1)=A(4,1)-((A(1,1)*A(1,3
))/ND):A(6,2)=A(4,2)-((A(1,2)*A(
1,3))/ND):A(6,3)=A(4,3)-((A(1,1)
*A(1,2))/ND)
230 R(1,1)=(A(6,1)^2)/(A(5,3)*A(
5,1)):R(1,2)= SQR(R(1,1)):IFA(6,
1)<0THENR(1,2)= -R(1,2)
240 R(2,1)=(A(6,2)^2)/(A(5,3)*A(
5,2)):R(2,2)= SQR(R(2,1)):IFA(6,
2)<0THENR(2,2)= -R(2,2)
250 R(3,1)=(A(6,3)^2)/(A(5,1)*A(
5,2)):R(3,2)= SQR(R(3,1)):IFA(6,
3)<0THENR(3,2)= -R(3,2)
260 R(4,1)=(R(1,2)-(R(2,2)*R(3,2
)))^2:R(4,1)=R(4,1)/((1-R(2,1))*
(1-R(3,1))):IFR(4,1)<0THENR(4,2)
=0ELSER(4,2)= SQR(R(4,1))
270 R(5,1)=(R(2,1)-(R(2,2)*R(3,2
)))^2:R(5,1)=R(5,1)/((1-R(1,1))*
(1-R(3,1))):IFR(5,1)<0THENR(5,2)
=0ELSER(5,2)= SQR(R(5,1))
280 RS=R(2,1)+R(1,1)-(2*R(1,2)*R
(2,2)*R(3,2)):RS=RS/(1-R(3,1)):I
FRS>1THENRS=1:RM=1ELSERM= SQR(RS
)
```

```
290 C=(A(6,3)*A(6,1))-(A(5,1)*A(
6,2)):C=C/(A(6,3)^2-(A(5,2)*A(5,
1)))
300 B=A(6,1)-(C*A(6,3)):B=B/A(5,
1)
310 D=A(2,3)-(B*A(2,1))-(C*A(2,2
))
320 CLS:PRINT"REGRESSION LINE EQ
UATION :":PRINT"YHAT = ";
330 IFD<0THENPRINT"-";ABS(D);ELS
E PRINTD;
340 IFB<0THENPRINT"-";ABS(B);"X1
";ELSE PRINT"+";B;"X1";
350 IFC<0THENPRINT" -";ABS(C);"X
2";ELSE PRINT" +";C;"X2";
360 PRINT:PRINT:PRINT"MEAN X1 ="
;A(2,1):PRINT"MEAN X2 =";A(2,2):
PRINT"MEAN Y  =";A(2,3)
370 PRINT:PRINT"CORRELATION COEF
FICIENT =";RM
380 R1=R(1,1)*100:R2=R(2,1)*100:
RS=RS*100:PRINT:PRINT"VARIATION
IN Y DUE TO :":PRINT TAB(11)"X1
=";R1;"%":PRINT TAB(11)"X2 =";R2
;"%":PRINT TAB(6)"X1 & X2 =";RS;
"%"
390 PRINT@481,"PRINT RESIDUAL TA
BLE (Y/N)";
400 I$= INKEY$:IFI$=""THEN400ELS
E IFI$="Y"THEN410ELSE IFI$="N"TH
EN490ELSE400
410 CLS:J=0
420 FORI=1TOND:YH=D+(B*X(1,I))+(
C*X(2,I)):RE=YH-X(3,I)
430 IFJ=0GOSUB530
440 PRINT USING"##";I;:PRINT TAB
(5);:PRINT USINGP$;YH;RE
450 J=J+1:IFJ>9THEN460ELSE470
460 PRINT@480,"TYPE <ENTER> TO C
ONTINUE ...";:INPUTI$:J=0:CLS
470 NEXTI
480 RS=RS/100:SD=(1-RS)*A(5,3):P
RINT:PRINT"SUM OF SQUARES OF DEV
IATION FROM REGRESSION =";SD
490 PRINT@480,"ESTIMATE VALUES O
F Y (Y/N)";
500 I$= INKEY$:IFI$=""THEN500ELS
E IFI$="Y"THEN510ELSE IFI$="N"TH
EN END ELSE500
510 CLS
520 INPUT"X1";XA:INPUT"X2";XB:YH
=D+(B*XA)+(C*XB):PRINT" Y =";YH:
GOTO520
530 CLS:PRINT"POINT";TAB(11)"YHA
T";TAB(22)"RES":PRINT:RETURN
```

***** ATOMIC TABLES *****
HITACHI PEACH

```
10 DIMD$(312)
20 'THE ELEMENTS BY DAVID BALAIC, 194/12
5 NAPIER ST., FITZROY
30 '3065. COMPLETED 31/1/81.
40 CLS:LOCATE31,3:PRINT"THE ELEMENTS";:L
OCATE32,4:PRINT"MAIN INDEX";:LOCATE7,6:P
RINT"YOU MAY CHOOSE ANY OF THE FOLLOWING
:";:LOCATE17,8:PRINT"A   -TO OBTAIN AN E
LEMENT'S ATOMIC NO.";:LOCATE17,9:PRINT"B
   -TO OBTAIN AN ELEMENT'S SYMBOL";
50 LOCATE17,10:PRINT"C   -TO OBTAIN AN A
TOMIC NO.'S ELEMENT"
60 LOCATE17,11:PRINT"D   -TO OBTAIN AN A
TOMIC NO.'S SYMBOL";:LOCATE17,12:PRINT"E
   -TO OBTAIN A SYMBOL'S ATOMIC NO.";:LO
CATE17,13:PRINT"F   -TO OBTAIN A SYMBOL'
S ELEMENT";
70 LOCATE17,14:PRINT"G   -FOR A LIST OF
ALL RELEVANT DATA ASSOCIATED WITH":LOCAT
E22,15:PRINT"AN ELEMENT."
80 A$=INKEY$:IFA$=""THEN80ELSEIFA$="A"TH
EN160ELSEIFA$="B"THEN110ELSEIFA$="C"THEN
90ELSEIFA$="D"THEN210ELSEIFA$="E"THEN230
ELSEIFA$="F"THEN280ELSEIFA$="G"THEN330EL
SE80
90 CLS:LOCATE22,10:PRINT"PLEASE ENTER AT
OMIC NO.";:INPUTA:GOSUB610:FORN=1TOA*3+2
:READD$:NEXT:CLS:LOCATE23,10:PRINT"THE E
LEMENT FOR"A"IS "D$;:LOCATE26,18:PRINT"H
IT ANY KEY TO CONTINUE";:RESTORE
100 IFINKEY$=""THEN100ELSE20
110 CLS:LOCATE25,10:PRINT"PLEASE ENTER E
LEMENT";:INPUTF$:GOSUB610:FORN=1TO312:RE
ADD$(N):IFD$(N)=F$THEN140ELSENEXT
120 GOSUB180
130 GOTO110
140 CLS:N=N+1:READD$(N):LOCATE24,10:PRIN
T"THE SYMBOL OF "F$" IS "D$(N):LOCATE26,
18:PRINT"HIT ANY KEY TO CONTINUE";:RESTO
RE
150 IFINKEY$=""THEN150ELSE20
160 CLS:LOCATE24,10:PRINT"PLEASE ENTER E
LEMENT";:INPUTS$:GOSUB610:FORN=1TO312:RE
ADD$(N):IFD$(N)=S$THEN190ELSENEXT:GOSUB1
80
170 GOTO160
180 CLS:RESTORE:LOCATE29,10:PRINT"SORRY,
 TRY AGAIN";:FORJ=1TO870:NEXT:RETURN
190 CLS:LOCATE23,10:PRINT"THE ATOMIC NO.
 OF "S$" IS "D$(N-1);:LOCATE26,18:PRINT"
HIT ANY KEY TO CONTINUE";:RESTORE
200 IFINKEY$=""THEN200ELSE20
210 CLS:LOCATE25,10:PRINT"PLEASE ENTER A
TOMIC NO.";:INPUTA:GOSUB610:FORN=1TOA*3+
3:READD$:NEXT:CLS:LOCATE28,10:PRINT"THE
SYMBOL OF"A"IS "D$;:LOCATE26,18:PRINT"HI
T ANY KEY TO CONTINUE";:RESTORE
220 IFINKEY$=""THEN220ELSE20
230 CLS:LOCATE25,10:PRINT"PLEASE ENTER S
YMBOL";:INPUTS$:GOSUB610:FORN=1TO312:REA
DD$(N):IFD$(N)=S$THEN260ELSENEXT
240 GOSUB180
250 GOTO230
260 CLS:LOCATE23,10:PRINT"THE ATOMIC NO.
 OF "S$" IS "D$(N-2);:LOCATE26,18:PRINT"
HIT ANY KEY TO CONTINUE";:RESTORE
270 IFINKEY$=""THEN270ELSE20
280 CLS:LOCATE25,10:PRINT"PLEASE ENTER S
YMBOL";:INPUTS$:GOSUB610:FORN=1TO312:REA
DD$(N):IFD$(N)=S$THEN310ELSENEXT
290 GOSUB180
300 GOTO280
310 CLS:LOCATE22,10:PRINT"THE ELEMENT FO
R "S$" IS "D$(N-1);:LOCATE26,18:PRINT"HI
T ANY KEY TO CONTINUE";:RESTORE
320 IFINKEY$=""THEN320ELSE20
330 CLS:LOCATE11,10:PRINT"PLEASE ENTER E
ITHER ATOMIC NO., SYMBOL OR ELEMENT";:IN
PUTX$:GOSUB610:IFVAL(X$)<>0THENN=VAL(X$)
:IFN>103THEN330ELSE360ELSEJ=LEN(X$)
340 FORN=1TO312:READD$(N):IFD$(N)=X$THEN
350ELSENEXT:RESTORE:GOTO330
350 RESTORE:N=N/3:N=INT(N):IFJ<=2THENN=N
-1
360 FORNN=1TON*3+1:READD$(NN):NEXT:CLS:L
OCATE14,3:PRINT"ATOMIC NO.:"N", ELEMENT:
 ";:READD$(NN):PRINTD$(NN)", SYMBOL: ";:
NN=NN+1:READD$(NN):PRINTD$(NN)".":LOCATE
15,4:PRINT"NO. OF PROTONS:"N", ELECTRONS
: "N"."
370 LOCATE37,13:PRINTN"P";:GOSUB410:LOCA
TE7,8:PRINT"GROUP:";:LOCATE7,9:PRINTGN$;
:LOCATE35,12:PRINT".-K="K"-.";:LOCATE35,
11:PRINT".-L="L"-.";:LOCATE35,10:PRINT".
-M="M"-.";:LOCATE35,9:PRINT".-N="P"-.";:
LOCATE35,8:PRINT".-O="Q"-.";:LOCATE35,7:
PRINT".-P="R"-.";:L
380 LOCATE26,18:PRINT"HIT ANY KEY TO CON
TINUE";:K=0:L=0:M=0:R=0:P=0:Q=0
390 IFINKEY$=""THEN390ELSE20
400 DATA541,544,
410 GN$="TRANSITIONAL ELEMENTS":IFN=30RN
=11ORN=19ORN=37ORN=55ORN=87THENGN$="I"EL
SEIFN=4ORN=12ORN=20ORN=38ORN=56ORN=88THE
NGN$="II"ELSEIFN=5ORN=13ORN=31ORN=49ORN=
```

```
81THENGN$="III"ELSEIFN=60RN=140RN=320RN=
500RN=82THENGN$="IV"
420 IFN=70RN=150RN=330RN=510RN=83THENGN$
="V"ELSEIFN=80RN=160RN=340RN=520RN=84THE
NGN$="VI"ELSEIFN=90RN=170RN=350RN=530RN=
85THENGN$="VII"ELSEIFN=20RN=100RN=180RN=
360RN=540RN=540RN=86THENGN$="VIII OR 0 -
 NOBLE GASES"ELSEIFN=1THENGN$="HYDROGEN"
:K=0:R=0:Q=0
430 IFN>=90THENGN$="ACTINIDE SERIES"ELSE
IFN>=58ANDN<=71THENGN$="LANTHANIDE SERIE
S"
440 IFN>=2THENK=2:N=N-2ELSEK=N:GOTO500
450 IFN>=8THENL=8:N=N-8ELSEL=N:GOTO500
460 IFN>=18THENM=18:N=N-18ELSEM=N:GOTO50
0
470 IFN>=32THENP=32:N=N-32ELSEP=N:GOTO50
0
480 IFN>=32THENQ=32:N=N-32ELSEQ=N:GOTO50
0
490 IFN>=18THENR=18:N=N-18ELSER=N
500 N=K+L+M+P+Q+R:IFN=19THENM=M-1:P=1:GO
TO600
510 IFN>=20ANDN<=30THENM=M-2:P=2:GOTO600
520 IFN=37THENP=P-1:Q=1:GOTO600
530 IFN=61THENP=P-1:Q=2:GOTO600
540 IFN=62THENQ=2:GOTO600
550 IFN>=38ANDN<=62THENP=P-2:Q=2:GOTO600
560 IFN=69THENQ=Q-1:R=1:GOTO600
570 IFN=93THENQ=Q-1:R=2:GOTO600
580 IFN=94THENR=2:GOTO600
590 IFN>=70ANDN<=94THENQ=Q-2:R=2:GOTO600
600 RETURN
610 CLS:LOCATE31,10:PRINT"COMPILING DATA
";:RETURN
620 DATA1,HYDROGEN,H,2,HELIUM,HE,3,LITHI
UM,LI,4,BERYLIUM,BE,5,BORON,B,6,CARBON,C
,7,NITROGEN,N,8,OXYGEN,O,9,FLOURINE,F,10
,NEON,NE,11,SODIUM,NA,12,MAGNESIUM,MG,13
,ALUMINIUM,AL,14,SILICON,SI,15,PHOSPHORU
S,P,16,SULPHUR,S,17,CHLORINE,CL,18,ARGON
,A,19,POTASSIUM,K
630 DATA20,CALCIUM,CA,21,SCANDIUM,SC,22,
TITANIUM,TI,23,VANADIUM,V,24,CHROMIUM,CR
,25,MANGANESE,MN,26,IRON,FE,27,COBALT,CO
,28,NICKEL,NI,29,COPPER,CU,30,ZINC,ZN,31
,GALLIUM,GA,32,GERMANIUM,GE,33,ARSENIC,A
S,34,SELENIUM,SE,35,BROMINE,BR,36,KRYPTO
N,KR,37,RUBIDIUM,RB
640 DATA38,STRONTIUM,SR,39,YTTRIUM,Y,40,
ZIRCONIUM,ZR,41,NIOBIUM,NB,42,MOLYBDENUM
,MO,43,TECHNETIUM,TC,44,RUTHENIUM,RU,45,
RHODIUM,RH,46,PALLADIUM,PD,47,SILVER,AG,
48,INDIUM,IN,49,CADMIUM,CD,50,TIN,SN,51,
ANTIMONY,SB,52,TELLURIUM,TE
650 DATA53,IODINE,I,54,XENON,XE,55,CAESI
UM,CS,56,BARIUM,BA,57,LANTHANUM,LA,58,CE
RIUM,CE,59,PRASEODYMIUM,PR,60,NEODYMIUM,
ND,61,PROMETHIUM,PM,62,SAMARIUM,SM,63,EU
ROPIUM,EU,64,GADOLINIUM,GD,65,TERBIUM,TB
,66,DYSPROSIUM,DY,67,HOLMIUM,HO
660 DATA68,ERBIUM,ER,69,THULIUM,TM,70,YT
TERBIUM,YB,71,LUTETIUM,LU,72,HAFNIUM,HF,
73,TANTALUM,TA,74,TUNGSTEN,W,75,RHENIUM,
RE,76,OSMIUM,OS,77,IRIDIUM,IR,78,PLATINU
M,PT,79,GOLD,AU,80,MERCURY,HG,81,THALLIU
M,TL,82,LEAD,PB,83,BISMUTH,BI
670 DATA84,POLONIUM,PO,85,ASTATINE,AT,86
,RADON,RN,87,FRANCIUM,FR,88,RADIUM,RA,89
,ACTINIUM,AC,90,THORIUM,TH,91,PROTOACTIN
IUM,PA,92,URANIUM,U,93,NEPTUNIUM,NP,94,P
LUTONIUM,PU,95,AMERICIUM,AM,96,CURIUM,CM
,97,BERKELIUM,BK,98,CALIFORNIUM,CF
680 DATA99,EINSTEINIUM,ES,100,FERNIUM,FM
,101,MENDELEVIUM,MV,102,NOBELIUM,NO,103,
LAWRENCIUM,LW
```

***** MULTIPLE REGRESSION ANALYSIS *****

HITACHI PEACH

```
10 'ROUTINE FOR MULTIPLE REGRESSION ANAL
YSIS
20 '(C) COPYRIGHT 1980
30 'T.R.JONES,
40 CLS:DEFINTI-N:DIMX(3,100),A(6,3),R(6,
3):P$="#######.#####"
50 PRINTTAB(26)"MULTIPLE REGRESSION ANAL
YSIS":PRINTTAB(26)STRING$(28,45)
60 PRINT:PRINT:PRINT:PRINTTAB(37)"MODEL"
:PRINT:PRINTTAB(31)"Y = A + BX1 + CX2"
70 LOCATE24,18:PRINT"TYPE <ENTER> TO CON
TINUE ....";:INPUTI$
80 CLS:PRINTTAB(35)"ENTER DATA":PRINT:PR
INT:PRINTTAB(21)"NUMBER OF DATA POINTS -
 MAXIMUM IS 100"
90 PRINTTAB(36)"*";:INPUTND:IFND<2ORND>1
00THENPRINTCHR$(30);CHR$(26);CHR$(13);CH
R$(30):GOTO90
100 PRINT:PRINT:FORI=1TOND:PRINT"X1(";I;
") =";:INPUTX(1,I):PRINTTAB(30)CHR$(30)"
X2(";I;") =";:INPUTX(2,I):PRINTTAB(52)CH
R$(30)"Y(";I;") =";:INPUTX(3,I):NEXTI
110 CLS:K=0:PRINTTAB(34)"DATA ENTERED":P
RINT:FORI=1TOND:K=K+1:PRINT"X1(";I;") ="
;X(1,I);TAB(30)"X2(";I;") =";X(2,I);TAB(
52)"Y(";I;") =";X(3,I)
120 IFK<10THEN130ELSELOCATE24,18:PRINT"T
YPE <ENTER> TO CONTINUE ...";:INPUTI$:K=
0:CLS
130 NEXTI
140 LOCATE30,18:PRINT"EDIT DATA (Y/N)";
150 I$=INKEY$:IFI$=""THEN150ELSEIFI$="Y"
THEN160ELSEIFI$="N"THEN180ELSE150
160 CLS:INPUT"DATA POINT TO BE EDITED";I
:IFI>NDTHEN160ELSEPRINTTAB(12)"X1(";I;")
 =";X(1,I);TAB(32)"X2(";I;") =";X(2,I);T
AB(52)"Y(";I;") =";X(3,I)
170 PRINT"NEW X1(";I;") =";:INPUTX(1,I):
PRINTTAB(32)CHR$(30);"X2(";I;") =";:INPU
TX(2,I):PRINTTAB(52)CHR$(30);"Y(";I;") =
";:INPUTX(3,I):GOTO110
180 CLS:LOCATE32,10:PRINT"COMPUTING ....
"
190 FORI=1TO6:FORJ=1TO3:A(I,J)=0:R(I,J)=
0:NEXTJ,I
200 FORI=1TOND:A(1,1)=A(1,1)+X(1,I):A(1,
2)=A(1,2)+X(2,I):A(1,3)=A(1,3)+X(3,I):NE
XTI:A(2,1)=A(1,1)/ND:A(2,2)=A(1,2)/ND:A(
2,3)=A(1,3)/ND
210 FORI=1TOND:A(3,1)=A(3,1)+(X(1,I)^2):
A(3,2)=A(3,2)+(X(2,I)^2):A(3,3)=A(3,3)+(
X(3,I)^2):NEXTI
220 FORI=1TOND:A(4,1)=A(4,1)+(X(1,I)*X(3
,I)):A(4,2)=A(4,2)+(X(2,I)*X(3,I)):A(4,3
)=A(4,3)+(X(1,I)*X(2,I)):NEXTI
230 A(5,1)=A(3,1)-(A(1,1)^2/ND):A(5,2)=A
(3,2)-(A(1,2)^2/ND):A(5,3)=A(3,3)-(A(1,3
)^2/ND)
240 A(6,1)=A(4,1)-((A(1,1)*A(1,3))/ND):A
(6,2)=A(4,2)-((A(1,2)*A(1,3))/ND):A(6,3)
=A(4,3)-((A(1,1)*A(1,2))/ND)
250 R(1,1)=(A(6,1)^2)/(A(5,3)*A(5,1)):R(
1,2)=SQR(R(1,1)):IFA(6,1)<0THENR(1,2)=-R
(1,2)
260 R(2,1)=(A(6,2)^2)/(A(5,3)*A(5,2)):R(
2,2)=SQR(R(2,1)):IFA(6,2)<0THENR(2,2)=-R
(2,2)
270 R(3,1)=(A(6,3)^2)/(A(5,1)*A(5,2)):R(
3,2)=SQR(R(3,1)):IFA(6,3)<0THENR(3,2)=-R
(3,2)
280 R(4,1)=(R(1,2)-(R(2,2)*R(3,2)))^2:R(
4,1)=R(4,1)/((1-R(2,1))*(1-R(3,1))):IFR(
4,1)<0THENR(4,2)=0ELSER(4,2)=SQR(R(4,1))
290 R(5,1)=(R(2,1)-(R(2,2)*R(3,2)))^2:R(
5,1)=R(5,1)/((1-R(1,1))*(1-R(3,1))):IFR(
5,1)<0THENR(5,2)=0ELSER(5,2)=SQR(R(5,1))
300 RS=R(2,1)+R(1,1)-(2*R(1,2)*R(2,2)*R(
3,2)):RS=RS/(1-R(3,1)):IFRS>1THENRS=1:RM
=1ELSERM=SQR(RS)
310 C=(A(6,3)*A(6,1))-(A(5,1)*A(6,2)):C=
C/(A(6,3)^2-(A(5,2)*A(5,1)))
```

```
320 B=A(6,1)-(C*A(6,3)):B=B/A(5,1)
330 D=A(2,3)-(B*A(2,1))-(C*A(2,2))
340 CLS:PRINT"THE EQUATION FOR THE REGRE
SSION LINE IS :":PRINT:PRINT"YHAT = ";
350 IFD<0THENPRINT"-";ABS(D);ELSEPRINTD;
360 IFB<0THENPRINT"-";ABS(B);"X1";ELSEPR
INT"+";B;"X1";
370 IFC<0THENPRINT"-";ABS(C);"X2";ELSEPR
INT" +";C;"X2";
380 PRINT:PRINT:PRINT"MEAN X1 =";A(2,1):
PRINT"MEAN X2 =";A(2,2):PRINT"MEAN Y  ="
;A(2,3)
390 PRINT:PRINT"COEFFICIENT OF MULTIPLE
CORRELATION =";RM
400 R1=R(1,1)*100:R2=R(2,1)*100:RS=RS*10
0:PRINT:PRINT"VARIATION IN Y DUE TO : X1
 =";R1;"%":PRINTTAB(32)"X2 =";R2;"%":PRI
NTTAB(27)"X1 & X2 =";RS;"%"
410 LOCATE25,18:PRINT"PRINT RESIDUAL TAB
LE (Y/N)";
420 I$=INKEY$:IFI$=""THEN420ELSEIFI$="Y"
THEN430ELSEIFI$="N"THEN510ELSE420
430 CLS:J=0
440 FORI=1TOND:YH=D+(B*X(1,I))+(C*X(2,I)
):RE=YH-X(3,I)
450 IFJ=0THENGOSUB550
460 PRINTUSING"##";I;:PRINTTAB(13);:PRIN
TUSINGP$;X(1,I);X(2,I);X(3,I);YH;RE
470 J=J+1:IFJ>9THEN480ELSE490
480 LOCATE24,18:PRINT"TYPE <ENTER> TO CO
NTINUE ...";:INPUTI$:J=0:CLS
490 NEXTI
500 RS=RS/100:SD=(1-RS)*A(5,3):PRINT:PRI
NT"SUM OF SQUARES OF DEVIATION FROM REGR
ESSION =";SD
510 LOCATE25,18:PRINT"ESTIMATE VALUES OF
 Y (Y/N)";
520 I$=INKEY$:IFI$=""THEN520ELSEIFI$="Y"
THEN530ELSEIFI$="N"THENENDELSE520
530 CLS
540 INPUT"X1";XA:PRINTTAB(29)CHR$(30);:I
NPUT"X2";XB:YH=D+(B*XA)+(C*XB):PRINTTAB(
50)CHR$(30);"Y =";YH:GOTO540
550 CLS:PRINT"POINT";TAB(19)"X1";TAB(30)
"X2";TAB(41)"Y";TAB(51)"YHAT";TAB(63)"RE
S":PRINT:RETURN
```

***** SPACE COMMANDER L1/4K *****

```
1 C.
2 A(45)=0:GOS.100
3 A=A+1
8 G=0
9 E=0:C=1000:D=20:F=28:G=0
10 P."IN THE SPACE YEAR";A;"WE CAPTURED";B;"ENEMY JET FIGHTERS"
11 P."THE ENEMY CAPTURED";E;"OF OURS,BRINGING THE TOTAL TO";C
12 P.
13 P."YOU NOW HAVE";D;"BATTLE STARS"
20 P.:P."BATTLE STARS COST";F;"JET FIGHTERS EACH"
50 I."HOW MANY BATTLE STARS DO YOU WISH TO BUY";G
55 IFG<>0THEN70
56 IFG*F>CTHEN50
57 H=0
60 I."HOW MANY BATTLE STARS DO YOU WISH TO SELL";H
70 I."HOW MANY BATTLE STARS DO YOU WANT IN ACTION";I
75 IFI>DTHEN70
99 G.104
100 A=0:B=0:C=0:D=0:E=0:G=0:F=0:H=0:I=0:J=0:K=0:L=0:M=0:N=0:O=0:P=0
101 Q=0:R=0:S=0:T=0:U=0:V=0:W=0:X=0:Y=0:Z=0
102 A=2500:C=1000
103 U=4:Y=10:G.15000
104 A(1)=RND(7)
105 B=A(1)*I
107 A=A+1
108 A(2)=RND(1000)
109 IFA(2)>CTHEN108
110 E=A(2)
111 C=C+B-E
112 IFC<20THEN10000
113 IFI>DTHEN70
114 F=RND(32)
115 IFF<24THEN114
116 D=D+G
117 T=G*F
118 C=C-T
119 IFC<20THEN10000
120 IFD=0THEN10000
196 Z=Z+1
198 C.
1010 D=D-H
1020 V=H*F
1025 F=RND(32)
1026 IFF<24THEN1025
1030 C=C+V
1198 G=0:H=0
1199 G.10
9999 C=0
10000 P."YOU HAVE BEEN DEFEATED "
10001 P."YOU HAD NO FIGHTERS LEFT"
11000 F.X=0TO1000:N.X
12000 G.1
15000 I."DO YOU WANT INSTRUCTIONS (Y OR N)";U
15003 C.
15005 IFU=NTHEN9
15010 P."THE OBJECT OF THE THE GAME IS NOT TO BE ANNIHILATED BY"
```

```
15011 P."THE ENEMY."
15015 P."THE MORE BATTLE STARS YOU PUT IN ACTION THE MORE "
15016 P."ENEMY FIGHTERS YOU CAN CATCH "
15020 P.:P.:P."PRESS ENTER TO BEGIN":IN.A$:C.:G.9
```

***** SPACE GALAXY L1/4K *****

```
10 B=0:R=1:REM BY BR. P. VAN EEKEN
11 D.24,12,S,P,A,C,E," ",G,A,L,A,X,Y,128,53
12 D.Y,O,U,R," ",C,O,N,T,R,O,L,S," ",O,P,E,R,A,T,E," ",W,H,E,N
13 D." ",T,H,E," ",L,I,N,E," ",A,B,O,V,E," ",I,S," ",V,I,S,I,B
14 D.L,E,.,256,47,Y,O,U," ",H,A,V,E," ",3," ",S,H,I,P,S," ",T
15 D.O," ",S,H,O,O,T," ",D,O,W,N," ",T,H,E," ",A,L,I,E,N,S," "
16 D.W,I,T,H,.,320,34,P,L,U,S,"," ","A," ",L,I,M,I,T,E,D," "
17 D.A,M,O,U,N,T," ",O,F," ",P,H,A,S,E,R,S,.,448,18,C,O,N,T,R
18 D.O,L,S,:,"  ",P,R,E,S,S,.,.,.,577,32,E,N,T,E,R,"     ",T,O
19 D." ",C,H,A,N,G,E," ",T,H,E," ",S,H,I,P," ",F,I,R,I,N,G,.
20 D.642,26,J," ",^,"      ",T,O," ",F,I,R,E," ",L,E,F,T," ",O
21 D.R," ",R,I,G,H,T,.,705,21,S,P,A,C,E," ",B,A,R," ",T,O," "
22 D.F,I,R,E," ",U,P,.,871,20,P,R,E,S,S," ",C,L,E,A,R," ",T,O
23 D." ",B,E,G,I,N,G,A,M,E," ",O,V,E,R
30 C.:F.X=120T.127:S.(X,1):N.X:S=0
40 F.P=1T.9:REA.C,Y:GOS.810:N.P:GOS.1004
60 IFP.(122,1)G.60
90 R=R.(9)+20:X=R.(99):C.
91 P.A.0:P.:P.A.0;"FIRSTLY, WHAT IS...
92 P.R;"INTO";X*R;"=";:I.J
93 IFJ<>XP." SORRY, TRY AGAIN...":G.91
94 C.:T=0:J=0:S=0
95 IFB>0P.A.978;"HI SCORE";A.993;"OF";B;"BY";A.1007;B$;
100 F.X=30T.90S.30:S.(X,42):S.(X-1,43):S.(X,43):S.(X+1,43)
110 S.(X-2,44):S.(X,44):S.(X+2,44):N.X
120 Z=R.(3):P.A.Z*15+960;"*";:F.X=0T.12:A(X)=0:N.X:GOS.999
150 F.A=0T.4:IFP.(127,1)=0G.300
151 IFP.(120,1)=0P=-1:G.400
152 IF(P.(124,1)=0)*(P.(127,1))P=0:G.400
153 IF(P.(120,1))*(P.(123,1)=0)*(P.(124,1))P=1:G.400
154 IF(A(A)>0)*(A(A+5)<99)*(R.(4)=1)G.600
159 IFK=-1GOS.1004:K=0
160 IFA(A)>0G.169
161 A(A)=3+4*R.(9):A(A+5)=125
169 X=A(A+5):Y=A(A)
170 IFP.(Z*30,44)=0G.1010
175 IFX<2G.160
176 R.(X-2,Y):R.(X+2,Y):R.(X-1,Y-1):R.(X+1,Y-1):R.(X,Y)
177 R.(X-1,Y+1):R.(X+1,Y+1):IFD=-1D=0:G.159
195 W=R.(20):F.V=0T.4:IFV=AG.199
196 IFA(V)<>A(A)G.199
197 IF(X-W>A(V+5)-5)*(X-W<A(V+5)+5)W=0
199 N.V:IFX-W<1A(A)=6+R.(8)*4:A(A+5)=126:G.220
200 IFX-W>2X=X-W
201 C=Y-3:IFC/4<>I.(C/4)Y=Y+1
205 S.(X-1,Y+1):S.(X+1,Y+1):S.(X,Y):S.(X-1,Y-1):S.(X+1,Y-1)
206 IFY=7S.(X-2,Y):S.(X+2,Y)
210 A(A+5)=X:A(A)=Y
220 N.A:G.150
300 GOS.999:P.A.Z*15+960;" ";:Z=Z+1:IFZ=4Z=1
305 IFA(Z+9)=0G.310
306 Z=Z+1:IFZ>3Z=1
307 G.305
310 P.A.Z*15+960;"*";:K=-1:G.159
400 X=Z*30:Y=41:P.A.60;"WAIT";:F=99
405 IF(X=2)+(Y<8)+(X=125)R=R-1:GOS.999:G.159
406 S.(X,Y):G=X+P:IF(P.(G,Y-1))+(P.(G,Y-2))G.499
407 IF(P<>0)*(P.(G-1,Y-1))+(P=1)*(P.(G+1,Y-1))G.499
440 R.(X,Y):X=X+P:Y=Y-2:G.405
499 R.(X,Y):X=X+P:Y=Y-1:F.V=0T.4
530 IF(X>A(V+5)+4)+(X<A(V+5)-4)+(Y>A(V)+2)+(Y<A(V)-2)N.V
550 X=A(V+5):Y=A(V):L=0:R=R-1
560 F.G=39T.YS.-4:L=L+5:N.G:IFY=7L=R.(2)*100
565 D=-1:T=T+L:A(V)=0:GOS.820:GOS.999:G.175
600 X=A(A+5):Y=A(A)+2:IFY<7G.159
601 K=-1:P.A.60;"WAIT";:F.G=42T.44:IFP.(X,G)G.605
602 N.G:G.620
605 F.N=YT.40:IFP.(X,N)G.620
610 N.N:P=0:G.700
620 Q=X:P=1:IFR.(2)=1P=-1
625 F.N=YT.40:IF(Q<5)+(Q>125)G.159
635 G=Q+P:IF(P.(G+1,N))+(P.(G-1,N))+(P.(G,N))+(P.(Q,N+1))G.159
640 Q=Q+P:N.N
700 F.N=YT.43:S.(X,N):IF(P.(X+P,N+1))+(P.(X,N+1))G.750
710 R.(X,N):X=X+P:N.N:G.159
750 R.(X,N):S=S+1:P=2:IFX>80P=3
760 IFX<40P=1
765 E=P:IFS>JGOS.999:J=S
770 A(P+9)=1:P=P*30:R.(P,42):R.(P-1,43):R.(P,43):R.(P+1,43)
780 R.(P-2,44):R.(P,44):R.(P+2,44):IFE=ZG.300
795 G.159
810 P.A.C;:F.O=1T.Y:REA.A$:P.A$;:GOS.820:N.O:IFY<>20Y=40:O=60
820 IFO<=YRET.
830 O=O-1:G.820
999 P.A.0:P.:P.A.24;"SPACE GALAXY";
1001 P.A.64;" SCORE";T;A.88;"PHASERS LEFT";R;A.114;
1002 IFS>0P.S;"SHIP";:IFS>1P."S";
1003 IFS>0P." LOST
1004 P.A.61;:IF(S<3)*(R>0)F.C=120T.127:S.(C,1):N.C:RET.
1010 C.:Y=9:C=468:GOS.810:S=0:R=1
1020 P.A.596;"YOUR SCORE WAS";T
1025 REST.:F.X=1T.281:REA.A$:N.X
1030 IFT>BB=T:P.A.660;:I."YOUR NAME PLEASE ";B$
1040 P.A.724;"PRESS CLEAR TO SEE THE INSTRUCTIONS
1050 P.A.794;"OR Q FOR A QUICK RESTART":GOS.1004
1060 IFP.(120,1)=0REST.:G.30
1070 IF(P.(122,1)=0)*(P.(120,1))G.90
1080 G.1060
```

***** READ-A-LINE L2/16K *****

```
64990 '           ********************************
64991 '           *          READ-A-LINE         *
64992 '           *  (C)  ALLAN CARTWRIGHT.      *
64993 '           *        4, AMBER PLACE,       *
64994 '           ********************************
64999 END
65000 DIMX$(127):X$(0)="END":X$(1)="FOR":X$(2)="RESET":X$(3)="SE
T":X$(4)="CLS":X$(5)="CMD":X$(6)="RANDOM":X$(7)="NEXT":X$(8)="DAT
A":X$(9)="INPUT":X$(10)="DIM":X$(11)="READ":X$(12)="LET":X$(13)="
GOTO":X$(14)="RUN":X$(15)="IF"
65010 X$(16)="RESTORE":X$(17)="GOSUB":X$(18)="RETURN":X$(19)="RE
M":X$(20)="STOP":X$(21)="ELSE":X$(22)="TRON":X$(23)="TROFF":X$(24
)="DEFSTR":X$(25)="DEFINT":X$(26)="DEFSNG":X$(27)="DEFDBL":X$(28)
="LINE":X$(29)="EDIT":X$(30)="ERROR"
65020 X$(31)="RESUME":X$(32)="OUT":X$(33)="ON":X$(34)="OPEN":X$(
35)="FIELD":X$(36)="GET":X$(37)="PUT":X$(38)="CLOSE":X$(39)="LOAD
":X$(40)="MERGE":X$(41)="NAME":X$(42)="KILL":X$(43)="LSET":X$(44)
="RSET":X$(45)="SAVE":X$(46)="SYSTEM"
65030 X$(47)="LPRINT":X$(48)="DEF":X$(49)="POKE":X$(50)="PRINT":
X$(51)="CONT":X$(52)="LIST":X$(53)="LLIST":X$(54)="DELETE":X$(55)
="AUTO":X$(56)="CLEAR":X$(57)="CLOAD":X$(58)="CSAVE":X$(59)="NEW"
:X$(60)="TAB(":X$(61)="TO":X$(62)="FN"
65040 X$(63)="USING":X$(64)="VARPTR":X$(65)="USR":X$(66)="FRL":X
$(67)="ERR":X$(68)="STRING$":X$(69)="INSTR":X$(70)="POINT":X$(71)
="TIME$":X$(72)="MEM":X$(73)="INKEY$":X$(74)="THEN":X$(75)="NOT":
X$(76)="STEP":X$(77)="+":X$(78)="-":X$(79)="*"
65050 X$(80)="/":X$(81)="[":X$(82)="AND":X$(83)="OR":X$(84)=">":
X$(85)="=":X$(86)="<":X$(87)="SGN":X$(88)="INT":X$(89)="ABS":X$(9
0)="FRE":X$(91)="INP":X$(92)="POS":X$(93)="SQR":X$(94)="RND":X$(9
5)="LOG"
65060 X$(96)="EXP":X$(97)="COS":X$(98)="SIN":X$(99)="TAN":X$(100
)="ATN":X$(101)="PEEK":X$(102)="CVI":X$(103)="CVS":X$(104)="CVD":
X$(105)="EOF":X$(106)="LOC":X$(107)="LOF":X$(108)="MKI$":X$(109)=
"MKS":X$(110)="MKD":X$(111)="CINT"
65070 X$(112)="CSNG":X$(113)="CDBL":X$(114)="FIX":X$(115)="LEN":
X$(116)="STR$":X$(117)="VAL":X$(118)="ASC":X$(119)="CHR$":X$(120)
="LEFT$":X$(121)="RIGHT$":X$(122)="MID$":X$(123)=" ":X$(124)=" ":
X$(125)=".":X$(126)="!":X$(127)="ISA"
65080 CLS:P1=PEEK(16548):P2=PEEK(16549):P=P1+P2*256+2:GOTO65150
65090 IFLN=64999THENPRINT@472,"END OF PROGRAM":POKE16548,233:POK
E16549,66:ENDELSEPRINTLN;
65100 IFPEEK(P)=0THEN65140
65110 IFPEEK(P)>=128THENPRINTX$(PEEK(P)-128);:GOTO65130
65120 PRINTCHR$(PEEK(P));
65130 P=P+1:GOTO65100
65140 PRINT@838," ";:INPUT"PRESS  =ENTER=  TO CONTINUE OR = E =
TO END";B$:IFB$="E"THENLN=64999:CLS:GOTO65090ELSEPOKE16548,P1:POK
E16549,P2:P1=PEEK(P+1):P2=PEEK(P+2):P=P+3
65150 LN=PEEK(P)+PEEK(P+1)*256:P=P+2:PRINT@448,CHR$(31);:GOTO650
90
```

***** FLIP L2/16K *****

```
10 ' "FLIP", BY R. WILLIAMS, 1979. TRS-80 VERSION COPYRIGHT (C)
A. QUICK 1981.
20 CLEAR2000:DEFSTR W,B,A,Z,T:DEFINTH,L,X,Y,I,E,P,V,S:CLS:PRINT@
530,"FLIP":PRINT:INPUT"           INSTRUCTIONS (Y/N)";A$:IFLEFT$(A
$,1)="Y"THEN270
30 RANDOM:W=STRING$(10,191):B=STRING$(10," "):H=0:L=0
40 CLS:FORX=1TO4:FORY=1TO4:P=RND(2):GOSUB200:NEXT:NEXT:Z="      "
:X=1:Y=1:S=0:X1=X:Y1=Y:GOSUB160
50 A=INKEY$:IFA=""THEN50
60 IFE<>1THENX2=X1:Y2=Y1:X1=X:Y1=Y
70 IFA="R"THENX=X+1:Z="RIGHT":IFX>4THENX=4:E=1:GOTO50:ELSE120
80 IFA="L"THENX=X-1:Z="LEFT ":IFX<1THENX=1:E=1:GOTO50:ELSE120
90 IFA="D"THENY=Y-1:Z="DOWN ":IFY<1THENY=1:E=1:GOTO50:ELSE120
100 IFA="U"THENY=Y+1:Z="UP   ":IFY>4THENY=4:E=1:GOTO50:ELSE120
110 IFA="Q"THEN150ELSE50
120 IFX2=XANDY2=YTHENZ="NO!!!":E=1:X=X1:Y=Y1:GOSUB160:GOTO50:ELS
EE=0
130 S=S+1:GOSUB180:GOSUB160:V=0
140 FORI=1TO4:FORJ=1TO4:V=V+POINT((I-1)*20,(4-J)*12):NEXT:NEXT:I
FV=0ORV=-16THEN220ELSE50
150 CLS:PRINT@192,"YOU QUIT!!":GOTO250
160 PRINT@44,"*** FLIP ***";:PRINT@236,"LAST GAME =";H;:PRINT@42
8,"LOW SCORE =";L;:PRINT@620,"MOVES =";S;
170 PRINT@812,"LAST MOVE = ";Z;:PRINT@1004,"NOW AT :";CHR$(X+64)
;",";Y;:RETURN
180 IFPOINT((X-1)*20,(4-Y)*12)=-1THENT=BELSET=W
190 FORI=(4-Y)*256TO(5-Y)*256-64STEP64:PRINT@I+(X-1)*10,T;:NEXT:
RETURN
200 IFP=1THENT=WELSET=B
210 GOTO190
220 CLS:PRINT@256,"YOU GOT IT IN"S"MOVES.":PRINT:PRINT"NOW.....
"
230 IFH=0ANDL=0THENH=S:L=S:GOTO250
240 H=S:IFS<LTHENL=S
250 PRINT:PRINT"LAST GAME:"H:PRINT"LOW SCORE:"L:PRINT:INPUT"ANOT
HER GAME (Y/N)";A$:IFLEFT$(A$,1)="Y"THEN40
260 CLS:END
270 CLS:PRINTTAB(5);"NOTES ON 'FLIP'"
280 PRINT"FLIP IS AN ANNOYING GAME...
YOU HAVE TO CHANGE THE 16"
290 PRINT"SQUARES TO EITHER ALL WHITE
OR ALL BLACK.
THE AIM IS TO DO THIS IN
THE LEAST NUMBER OF MOVES."
300 PRINT:PRINT"YOU MOVE WITH THE KEYS:
        U - UP
        D - DOWN
        L - LEFT
        R - RIGHT"
310 PRINT@960,"HIT ANY KEY TO CONTINUE";
```

```
320 A=INKEY$:IFA=""THEN320ELSECLS:PRINTTAB(5);"NOTES ON 'FLIP'"
330 PRINT"YOU ALWAYS START IN THE
BOTTOM LEFT CORNER (A,1)

WHENEVER YOU ENTER A NEW
SQUARE, IT CHANGES COLOUR!"
340 PRINT"YOU CANNOT MOVE BACK TO A
POSITION YOU HAVE JUST COME
FROM. I.E RIGHT, THEN
LEFT, ISN'T OK."
350 PRINT:PRINT"ENTER 'Q' TO QUIT."
360 PRINT@960,"HIT ANY KEY TO BEGIN";
370 A=INKEY$:IFA=""THEN370ELSE30
```

***** KEWORD L2/16K ml. *****

```
7380:  31 7A 73 2A 16 40 22 F7 7F F3 21 A1 73 22 16 40
7390:  3E C3 21 A0 75 32 DF 41 22 E0 41 CD 28 79 C3 CC
73A0:  06 CD E3 03 B7 CA DA 7F FE 01 C8 FE 02 CA B8 78
73B0:  FE 03 CA D5 76 FE 04 CA 8D 75 FE 05 CA 77 75 FE
73C0:  06 CA DE 75 FE 07 CA 3B 7C FE 16 CA 28 79 FE 10
73D0:  CA A0 76 FE 11 20 0A AF 32 64 75 32 6B 75 D3 FE
73E0:  C9 FE 0E CA E7 77 FE 12 20 05 AF 32 DA 7F C9 FE
73F0:  13 20 0F 3E 05 32 64 75 3C 32 6B 75 3E FF D3 FE
7400:  AF C9 FE 14 20 0B 3E 01 32 64 75 3C 32 6B 75 AF
7410:  C9 FE 0F 20 07 3E C9 32 DA 7F AF C9 C3 55 75 E5
7420:  FE 61 DA 54 75 FE 7B D2 54 75 DD E5 DD 21 0C 74
7430:  32 35 74 DD 7E 00 2A 87 74 85 6F 30 01 24 2B 22
7440:  87 74 21 50 74 22 16 40 AF DD E1 E1 CD 5B 75 C9
7450:  E5 2A 87 74 23 22 87 74 7E CB 7F 28 0E CB BF 21
7460:  89 74 22 87 74 21 A1 73 22 16 40 E1 C9 00 04 08
7470:  10 14 19 26 2B 31 37 3A 55 59 5E 63 68 6D 72 79
7480:  81 86 B1 B5 BA C1 C5 89 74 41 53 43 A8 43 4C 53
7490:  A0 43 53 41 56 45 22 41 A2 44 41 54 C1 45 4C 53
74A0:  45 A0 46 4F 52 20 49 20 3D 20 31 20 54 4F A0 47
74B0:  4F 54 4F A0 47 4F 53 55 42 A0 49 4E 50 55 54 A0
74C0:  22 2C A2 49 4B 24 3D 49 4E 4B 45 59 24 3A 20 49
74D0:  46 20 49 4B 24 3D 22 22 20 54 48 45 4E A0 4C 45
74E0:  4E A8 4D 49 44 24 A8 4E 45 58 54 A0 50 4F 4B 45
74F0:  A0 50 45 45 4B A8 43 48 52 24 A8 52 45 54 55 52
7500:  4E A0 53 54 52 49 4E 47 24 A8 54 48 45 4E A0 4F
7510:  55 54 32 35 34 2C 32 35 35 20 20 20 20 20 20 20
7520:  20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
7530:  20 20 20 20 20 20 20 20 20 A0 56 41 4C A8 53 54
7540:  52 24 A8 52 49 47 48 54 24 A8 52 4E 44 A8 4C 45
7550:  46 54 24 A8 E1 E5 21 60 30 18 04 E5 21 90 30 F5
7560:  C5 4C 45 3E 00 D3 FF 10 FE 45 3E 00 D3 FF 10 FE
7570:  0D 20 EF C1 F1 E1 C9 E5 CD C9 01 21 17 7C CD AE
7580:  78 21 1F 74 22 1D 74 AF E1 CD 55 75 C9 E5 CD C9
7590:  01 21 29 7C CD AE 78 21 55 75 22 1D 74 AF E1 C9
75A0:  3A 04 38 FE 80 28 0E FE 08 28 01 C9 C5 01 00 FF
75B0:  CD 60 00 C1 C9 E5 21 BD 75 22 E0 41 E1 CD 49 00
75C0:  FE 4C 28 10 E5 C5 C5 E1 4E 23 7E B1 28 04 C1 E1
75D0:  AF C9 C1 E1 E5 21 A0 75 22 E0 41 E1 AF C9 E5 21
75E0:  FC 75 22 16 40 CD C9 01 21 42 76 CD AE 78 21 0F
75F0:  75 22 21 76 21 C0 3C 22 20 40 E1 C9 E5 CD E3 03
7600:  B7 CA 6B 74 FE 01 28 1B FE 08 28 2A FE 0D 28 13
7610:  2A 21 76 F5 7E FE A0 28 26 F1 77 23 22 21 76 E1
7620:  C9 00 00 3E 80 2A 21 76 77 21 A1 73 22 16 40 3E
7630:  18 CD C9 01 E1 C9 2A 21 76 2B 22 21 76 E1 C9 F1
7640:  18 E7 46 2E 3D 20 46 49 58 20 55 53 45 52 20 4B
7650:  45 59 2E 20 20 20 20 54 59 50 45 20 41 4E 59 20
7660:  54 45 58 54 20 55 50 20 54 4F 20 34 30 20 43 48
7670:  52 53 2E 20 20 20 20 20 20 20 20 20 20 20 20 20
7680:  20 20 20 20 20 20 20 54 45 52 4D 49 4E 41 54 45
7690:  20 57 49 54 48 20 42 52 45 41 4B 20 4B 45 59 00
76A0:  F5 D9 CD D1 05 20 22 1E 00 21 FF 3B 01 00 40 23
76B0:  7E FE 20 38 18 FE 7B 30 18 CD 9C 03 10 F1 3E 0D
76C0:  CD 9C 03 1C 7B FE 10 20 E3 D9 F1 AF C9 3E 2E 18
76D0:  E8 3E 23 18 E4 F5 3E FF D3 FE D9 CD C9 01 21 24
76E0:  77 CD AE 78 CD 49 00 3E 04 D3 FF 21 3F 3C 3E 83
76F0:  77 DB FF 17 30 FB 06 32 10 FE 3E 05 D3 FF 06 10
7700:  10 FE 3E 06 D3 FF 06 10 10 FE 3E 04 D3 FF 7E EE
7710:  33 77 3A 40 38 E6 04 20 02 18 D6 D9 F1 AF D3 FF
7720:  CD C9 01 C9 2A 20 2A 20 2A 20 54 41 50 45 20 43
7730:  4C 4F 4E 45 20 2A 20 2A 20 2A 0D 4D 41 4B 45 53
7740:  20 45 58 41 43 54 20 43 4F 50 59 2E 20 52 45 51
7750:  55 49 52 45 53 20 54 57 4F 20 52 45 43 4F 52 44
7760:  45 52 53 20 49 4E 53 45 52 54 20 41 55 58 20 4C
7770:  45 41 44 20 49 4E 54 4F 20 20 20 41 55 58 20 53
7780:  4F 43 4B 45 54 20 4F 46 20 53 45 43 4F 4E 44 20
7790:  52 45 43 4F 52 44 45 52 20 41 4E 44 20 53 45 54
77A0:  20 49 54 20 54 4F 20 52 45 43 4F 52 44 0D 50 52
77B0:  45 53 53 20 41 4E 59 20 4B 45 59 20 54 4F 20 53
77C0:  54 41 52 54 2E 20 20 3C 42 52 45 41 4B 3E 20 57
77D0:  48 49 4C 53 54 20 52 45 41 44 49 4E 47 20 54 4F
77E0:  20 41 42 4F 52 54 00 F5 C5 D5 E5 CD C9 01 21 7D
77F0:  7F CD AE 78 CD 49 00 FE 01 CA 81 78 FE 08 20 19
7800:  3A D3 7F FE 00 28 ED 3D 32 D3 7F 2A D8 7F 23 22
7810:  D8 7F 3E 08 CD 33 00 18 DB FE 30 38 D7 FE 3A 30
7820:  02 18 08 FE 41 38 CD FE 47 30 C9 2A D8 7F 77 CD
7830:  33 00 2B 22 D8 7F 3A D3 7F 3C FE 04 28 05 32 D3
7840:  7F 18 B1 21 BB 7F CD AE 78 ED 5B D4 7F CD 95 78
7850:  32 D4 7F ED 5B D6 7F CD 95 78 32 D5 7F 2A D4 7F
7860:  CD AF 0F 21 C3 7F CD AE 78 21 D4 7F 6E 26 00 CD
7870:  AF 0F 21 CB 7F CD AE 78 21 D5 7F 6E 26 00 CD AF
7880:  0F 21 D7 7F 22 D8 7F AF 32 D3 7F E1 D1 C1 F1 C3
7890:  CC 06 C3 CC 06 7A CD A6 78 07 07 07 07 F5 7B CD
78A0:  A6 78 5F F1 B3 C9 D6 30 FE 0A D8 D6 07 C9 7E FE
78B0:  00 C8 CD 3A 03 23 18 F6 E5 2A 16 40 22 ED 78 21
78C0:  CD 78 22 16 40 21 F1 78 22 EF 78 E1 C9 E5 CD E3
78D0:  03 2A EF 78 7E 23 22 EF 78 CB 7F 28 0E CB BF 2A
78E0:  ED 78 22 16 40 21 F1 78 22 EF 78 E1 C9 00 00 F1
78F0:  78 46 4F 52 20 49 3D 31 54 4F 32 3A 4F 55 54 32
7900:  35 35 2C 34 3A 46 4F 52 20 5A 3D 31 54 4F 32 30
7910:  30 30 3A 4E 45 58 54 5A 3A 43 53 41 56 45 22 58
```

```
7920: 22 3A 4E 45 58 54 49 80 E5 CD C9 01 21 34 79 CD
7930: AE 78 E1 C9 2A 20 2A 20 4B 45 59 57 4F 52 44 20
7940: 55 54 49 4C 49 54 59 20 28 56 45 52 20 31 2E 35
7950: 29 20 3C 43 3E 20 46 2E 45 4C 4C 45 54 54 20 53
7960: 45 50 54 20 31 39 38 31 20 2A 20 2A 0D 46 4F 52
7970: 20 41 4C 4C 20 46 55 4E 43 54 49 4F 4E 53 20 50
7980: 52 45 53 53 20 53 48 49 46 54 2D 44 4F 57 4E 20
7990: 41 52 52 4F 57 20 3C 43 54 4C 3E 20 26 20 41 4C
79A0: 50 48 41 20 43 48 52 2E 0D 41 3D 20 42 52 45 41
79B0: 4B 20 20 20 20 20 20 20 20 42 3D 20 42 20 4C 41 5A
79C0: 59 20 53 41 56 45 20 20 43 3D 20 43 4C 4F 4E 45
79D0: 20 20 20 20 20 20 20 20 20 44 3D 44 49 53 41 42 4C
79E0: 45 20 4B 57 44 0D 45 3D 20 45 4E 41 42 4C 45 20
79F0: 4B 57 44 20 20 46 3D 20 46 49 58 20 3C 55 3E 20
7A00: 4B 45 59 20 20 47 3D 20 47 45 54 20 54 41 50 45
7A10: 20 20 20 20 20 4E 3D 20 4E 55 4D 42 45 52 0D 4F
7A20: 3D 20 43 55 52 53 4F 52 20 4F 46 46 20 20 50 3D
7A30: 20 50 52 49 4E 54 20 53 43 52 45 45 4E 20 51 3D
7A40: 20 51 55 49 54 20 53 4F 55 4E 44 20 20 20 52 3D
7A50: 20 43 55 52 53 4F 52 20 4F 4E 0D 53 3D 20 53 2D
7A60: 38 30 20 53 4F 55 4E 44 20 20 54 3D 20 54 52 53
7A70: 20 53 4F 55 4E 44 20 20 20 20 56 3D 20 56 49 44
7A80: 45 4F 20 54 45 58 54 0D 53 48 49 46 54 20 4B 45
7A90: 59 20 57 4F 52 44 53 20 3A 2D 0D 41 3D 41 53 43
7AA0: 28 20 20 42 3D 20 43 4C 53 20 20 20 20 43 3D 20
7AB0: 43 53 41 56 45 22 41 22 20 44 3D 20 44 41 54 41
7AC0: 20 20 45 3D 20 45 4C 53 45 20 20 20 20 20 46 3D
7AD0: 46 4F 52 49 3D 31 54 4F 20 20 20 47 3D 20 47 4F
7AE0: 54 4F 20 48 3D 47 4F 53 55 42 20 20 20 49 3D 20
7AF0: 49 4E 50 55 54 20 20 20 20 4A 3D 20 22 2C 22 20
7B00: 20 20 4B 3D 49 4B 24 3D 49 4E 4B 45 59 24 20 45
7B10: 54 43 2E 0D 4C 3D 20 4C 45 4E 28 20 4D 3D 20 4D
7B20: 49 44 24 28 20 20 4E 3D 20 4E 45 58 54 20 20 20
7B30: 20 20 4F 3D 20 50 4F 4B 45 20 20 50 3D 20 50 45
7B40: 45 4B 28 20 20 20 20 51 3D 20 43 48 52 24 28 20
7B50: 20 20 20 20 52 3D 20 52 45 54 55 52 4E 20 20 20
7B60: 20 20 20 20 20 20 53 3D 53 54 52 49 4E 47 24 28
7B70: 20 20 54 3D 20 54 48 45 4E 20 20 55 3D 4F 55 54
7B80: 32 35 34 2C 32 35 35 0D 56 3D 20 56 41 4C 28 20
7B90: 57 3D 20 53 54 52 24 28 20 20 58 3D 20 52 49 47
7BA0: 48 54 24 28 20 20 59 3D 20 52 4E 44 28 20 20 5A
7BB0: 3D 20 4C 45 46 54 24 28 20 0D 4C 49 53 54 3A 20
7BC0: 48 4F 4C 44 20 3C 53 3E 20 54 4F 20 53 4C 4F 57
7BD0: 2E 20 20 50 52 45 53 53 20 3C 57 3E 20 54 4F 20
7BE0: 48 41 4C 54 20 53 43 52 4F 4C 4C 0D 41 4E 59 20
7BF0: 4B 45 59 20 46 4F 52 20 4E 45 58 54 20 4C 49 4E
7C00: 45 2E 20 3C 4C 3E 20 54 4F 20 52 45 53 55 4D 45
7C10: 20 4C 49 53 54 0D 00 4B 45 59 20 57 4F 52 44 20
7C20: 45 4E 41 42 4C 45 44 0D 00 4B 45 59 57 4F 52 44
7C30: 20 44 49 53 41 42 4C 45 44 0D 00 2A F7 7F 22 16
7C40: 40 CD C9 01 21 EC 7D CD AE 78 CD 49 00 FE 01 CA
7C50: C7 7D FE 0D 20 F4 AF CD 12 02 CD 96 02 21 D2 7D
7C60: 06 08 CD 35 02 77 23 10 F9 21 D2 7D 3E 55 BE CA
7C70: 9A 7C 3E D3 BE C2 48 7D 23 BE C2 66 7D CD C9 01
7C80: 21 83 7E CD AE 78 3A D5 7D CD 33 00 3E 22 CD 33
7C90: 00 CD F8 01 CD C1 7D C3 C7 7D 21 D9 7D 3E 3C BE
7CA0: C2 84 7D CD 2C 02 CD 35 02 47 CD 14 03 22 DC 7D
7CB0: 18 1D CD 2C 02 3A 40 38 E6 04 C2 84 7D CD 35 02
7CC0: FE 78 28 1E FE 3C 20 F5 CD 35 02 47 CD 14 03 85
7CD0: 4F CD 35 02 23 81 4F 10 F8 CD 35 02 B9 28 D3 C3
7CE0: 84 7D 2B 22 E2 7D CD 14 03 22 E8 7D CD F8 01 CD
7CF0: C9 01 21 08 7F CD AE 78 2A DC 7D 7C CD A2 7D ED
7D00: 43 DC 7D 7D CD A2 7D ED 43 DE 7D 2A E2 7D 7C CD
7D10: A2 7D ED 43 E2 7D 7D CD A2 7D ED 43 E4 7D 2A E8
7D20: 7D 7C CD A2 7D ED 43 E8 7D 7D CD A2 7D ED 43 EA
7D30: 7D 3E 20 32 D9 7D 21 D2 7D ED 5B 20 40 01 1C 00
7D40: ED B0 CD C1 7D C3 C7 7D CD C9 01 CD F8 01 21 A9
7D50: 7E CD AE 78 ED 5B 20 40 01 08 00 21 D2 7D ED B0
7D60: CD C1 7D C3 C7 7D CD F8 01 CD C9 01 21 D7 7E CD
7D70: AE 78 ED 5B 20 40 01 07 00 21 D2 7D ED B0 CD C1
7D80: 7D C3 C7 7D CD F8 01 CD C9 01 21 4E 7F CD AE 78
7D90: 21 D2 7D ED 5B 20 40 01 07 00 ED B0 CD C1 7D C3
7DA0: C7 7D 47 CB 3F CB 3F CB 3F CB 3F CD B7 7D 4F 78
7DB0: E6 0F CD B7 7D 47 C9 C6 30 FE 3A FA C0 7D C6 07
7DC0: C9 3E 0D CD 33 00 C9 E5 21 A1 73 22 16 40 E1 C3
7DD0: CC 06 20 20 20 20 20 20 20 20 20 20 20 20 20 20
7DE0: 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
7DF0: 20 20 20 20 20 20 20 20 2A 20 2A 20 2A 20 47 45
7E00: 54 20 54 41 50 45 20 4E 41 4D 45 20 2A 20 2A 20
7E10: 2A 0D 52 45 41 44 53 20 55 4E 4B 4E 4F 57 4E 20
7E20: 54 41 50 45 20 41 4E 44 20 47 49 56 45 53 20 54
7E30: 59 50 45 2C 20 4E 41 4D 45 20 26 20 41 44 44 52
7E40: 45 53 53 45 53 20 20 20 20 20 20 20 20 20 20 20
7E50: 20 20 20 20 4C 4F 41 44 20 54 41 50 45 20 26 20
7E60: 50 52 45 53 53 20 45 4E 54 45 52 2E 20 42 52 45
7E70: 41 4B 20 57 49 4C 4C 20 41 42 4F 52 54 20 52 45
7E80: 41 44 00 54 41 50 45 20 49 53 20 49 4E 20 42 41
7E90: 53 49 43 20 46 4F 52 4D 41 54 20 49 54 53 20 46
7EA0: 49 4C 45 20 49 53 20 22 00 55 4E 4B 4E 4F 57 4E
7EB0: 20 48 45 41 44 45 52 20 43 4F 44 45 2E 20 54 48
7EC0: 45 20 46 49 52 53 54 20 38 20 42 59 54 45 53 20
7ED0: 41 52 45 3A 2D 0D 00 54 48 49 53 20 49 53 20 41
7EE0: 4E 20 45 44 54 41 53 4D 20 53 4F 55 52 43 45 20
7EF0: 48 45 41 44 45 52 2E 20 49 54 53 20 4E 41 4D 45
7F00: 20 49 53 3A 2D 20 20 00 54 48 49 53 20 49 53 20
7F10: 41 20 53 59 53 54 45 4D 20 54 41 50 45 20 49 54
7F20: 53 20 41 44 44 52 45 53 53 45 53 20 41 52 45 0D
7F30: 20 4E 41 4D 45 20 20 20 20 20 20 53 54 41 52 54 20
7F40: 20 45 4E 44 20 20 20 45 4E 54 52 59 0D 00 41 20
7F50: 53 59 53 54 45 4D 20 54 41 50 45 20 42 55 54 20
7F60: 57 45 20 48 41 56 45 20 4C 4F 53 54 20 53 59 4E
7F70: 43 2E 20 4E 41 4D 45 20 49 53 3A 0D 00 45 4E 54
7F80: 45 52 20 41 20 34 20 44 49 47 49 54 20 48 45 58
7F90: 20 56 41 4C 55 45 2E 20 20 55 53 45 20 4C 45 41
7FA0: 44 49 4E 47 20 5A 45 52 4F 53 20 49 46 20 4E 45
7FB0: 45 44 45 44 0D 48 45 58 3D 20 00 20 20 44 45 43
7FC0: 3D 20 00 20 20 4C 53 42 3D 20 00 20 20 4D 53 42
7FD0: 3D 20 00 00 00 00 00 00 D7 7F 00 E5 2A F5 7F 2B
7FE0: 7C FE 00 20 0A 2A 20 40 7E EE D3 77 21 00 03 22
7FF0: F5 7F E1 AF C9 00 03 00 00
```

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

Next month's issue will contain at least the following programs plus the usual features and articles. An (80) after a program title indicates that the program will be for TRS-80 Model 1/3 or System 80/Video Genie computers. (Colour) indicates that the program will be for the TRS-80 Colour Computer and the Hitachi Peach.

** RUBIK CUBE SOLVER LII/16K (80) **

This program provides step by step instructions for solving the Rubik cube, that multi-coloured monster in your home that has driven everyone mad. This program functions in the same fashion as the various books that are available on the subject, providing step by step moves for each stage.

** JUMBLED PLAYERS LII/16K (80) **

You have probably all seen those competitions in the paper, where a name is all jumbled up and to win, all you have to do is unjumble the letters. Well, this program does that for you. Just type in the letters in any order and the program returns with the player's name.

** RESET 32K/Disk (80) **

As requested, here is a program that will reset the computer when you hold down the space bar and Clear key together, the program relocates itself to high memory and protects itself. It will also patch itself into any existing keyboard driver automatically.

** SKY DIVER LII/16K (80) **

This program takes you back to World War II and places you inside a Dakota transport plane. The year is 1942, behind the enemy lines. The cold French countryside is covered with snow, your mission is simply to land behind enemy lines - safely. You may encounter a FEW !! problems, snipers, high voltage lines, forests, deep water and even your parachute malfunctioning. Includes realistic graphics and sound.

** MEASUREMENTS (Colour) **

This program was primarily desinged to help high school students with their studies. It poses problems and asks questions on the following solid figures: cube, rectangle faced prism, rhombus faced prism, triangle faced prism, parallelogram faced prism, trapesium faced prism, cylinder, pyramid cone and sphere. All of the above figures are displayed graphically before the questions are asked.

** INCOME TAX (Colour) **

Before you know where you are, it will be that time of year again, so be ready for it with this program which will calculate the income tax payable for a person who would normally complete the Australian "S" Income Tax return.

APPLICATION FOR PUBLICATION OF A PROGRAM IN MICRO-80

Date .....

Tick where appropriate

To **MICRO-80** SOFTWARE DEPT. PO BOX 145 MORPHETTVALE SA 5162

Please consider the enclosed program for . . .

(i) Publication in MICRO-80 .....

(ii) Publication on disk or cassette only .....

(iii) Both .....

Name .....

Address .....

.....

.....Postcode .....

*** * * CHECK LIST * * ***

Please ensure that the cassette or disk is clearly marked with your name and address, program name(s), Memory size, Level I, II, System 1 or 2, Edtasm, System, etc. The use of REM statements with your name and address is suggested, in case the program becomes separated from the accompanying literature.

Ensure that you supply adequate instructions, notes on what the program does and how it does it, etc.

For system tapes, the start, end, and entry points, etc.

The changes or improvements that you think may improve it.

Please package securely — padabags are suggested — and enclose stamps or postage if you want your cassette or disk returned.

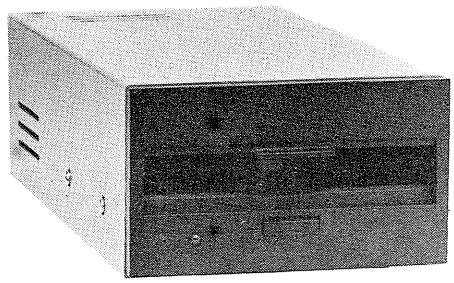
***** CASSETTE/DISK EDITION INDEX *****

The cassette edition of MICRO-80 contains all the software listed each month, on cassette. The cassette also contains the source code for machine language programs which may not have been printed due to space restrictions. All programs are recorded twice. Level I programs can only be loaded into a Level I TRS-80 unless the Level I in Level 2 program from the MICRO-80 Software Library - Vol. 1 is first loaded into your Level 2 TRS-80 or System 80/Video Genie. Note: System 80/Video Genie computers have had different tape-counters fitted at different times. The approximate start positions shown are correct for the very early System 80 without the volume control or level meter. They are probably incorrect for later machines. The rates for a cassette subscription are printed on the inside front cover of each issue of the magazine.

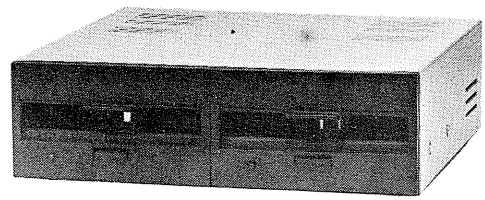
The disk edition contains all those programs which can be executed from disk, including Level I programs. level I disk programs are saved in the NEWDOS format. Users require the Level I/CMD utility supplied with NEWDOS+ or NEWDOS 80 version 1.0 to run them.

	TYPE	I.D.	DISK FILESPEC	APPROX. START POSITION CTR-41	CTR-80	SYSTEM-80
SIDE 1						
READ A LINE	LII/16K	R	READLINE/BAS	18	10	10
"	"	"	"	51	28	29
FLIP	LII/16K	F	FLIP/BAS	81	45	47
"	"	"	"	104	58	61
KEYWORD	SYSTEM	KEWORD	KEWORD/CMD	127	71	75
"	"	"	"	163	91	96
"	EDTASM	"	KEWORD/EDT	197	110	116
SIDE 2						
KEYWORD	EDTASM	"	KEWORD/EDT	18	10	10
SPACE COMMANDER	LI/4K	-	COMMANDR/LV1	222	125	130
"	"	-	"	247	139	146
SPACE GALAXY	LI/4K	-	GALAXY/LV1	273	154	162
"	"	-	"	322	185	194

SAVE A PACKET ON MICRO-80's DISK DRIVE PACKAGES FOR TRS-80 MODEL 1 AND SYSTEM 80 MICROCOMPUTERS



SINGLE DRIVE PACKAGE *from ...* $499



DUAL DRIVE PACKAGE *from ...* $874

Bigger volume means lower cost price, which we are passing on to you. Avoid the annoying bundle of cables, wires and separate boxes. MICRO-80 is now offering our well-proven MPI disk drives in attractive, self-contained single or dual-drive cabinets complete with internal power supply. Our drive Ø and dual-drive packages also include the appropriate version of DOSPLUS and dual-drive cable.

The best news of all is the specially reduced package prices ...
SAVE $23 — $107 over our already low prices!

Choose the appropriate system from the table below:

DRIVE TYPE	No. of Tracks	No. of Heads	Capacity	Dosplus Version	Price	* Saving
DRIVE Ø						
1 x MPI B51	40	1	100K	3.3	$499	$77.95
1 x MPI B52	40	2	200K	3.4	$639	$97.95
1 x MPI B92	80	2	400K	3.4	$799	$107.95
DRIVE 1						
1 x MPI B51	40	1	100K	—	$415	$23.00
1 x MPI B52	40	2	200K	—	$525	$23.00
1 x MPI B92	80	2	400K	—	$695	$23.00

*Represents the saving compared with buying all the items included in the package separately

•Drive Ø package includes one bare disk drive, self-contained single-drive cabinet/power supply as illustrated, two drive cable and the version of DOSPLUS indicated.

•Drive 1 package includes one bare disk drive and self-contained single-drive cabinet/power supply as illustrated.

If it's a dual-drive system you need, then take advantage of our dual-drive package and
SAVE a further $40 on the price of two single-drive packages ...

DRIVE TYPE	No. of Tracks	No. of Heads	Capacity	Dosplus Version	Price
2 x MPI B51	40 ea	1 ea	2 x 100K	3.3	$874
2 x MPI B52	40 ea	2 ea	2 x 200K	3.4	$1125
2 x MPI B92	80 ea	2 ea	2 x 400K	3.4	$1454

Dual-drive package includes two bare disk drives, self-contained dual-drive cabinet/power supply as illustrated, two drive cables and the version of Dosplus indicated.

NOTE: All 40 track drives are completely compatible with 35 track operating systems such as TRSDOS. DOSPLUS allows you to realise an additional 14% capacity compared with TRSDOS. Under DOSPLUS 3.4, 80 track drives can read 35/40 track diskettes.

All disk drive components are still available separately:

BARE DRIVES — MPI drives offer the fastest track-to-track access time (5 milliseconds) available. All drives are capable of operating in double density for 80% greater storage capacity.

	Price		Freight
MPI B51 40 track, single-head, 100K	$399	New, Reduced Price	$5.00
MPI B52 40 track, dual-head, 200K	$449		$5.00
MPI B92 80 track, dual-head, 400K	$619		$5.00
Simple, wrap-around cabinet	$12		$2.00
Separate, dual-drive power supply	$85		$8.00

	Price	Freight
Self-contained, single drive cabinet/power supply	$99	$5.00
Self-contained, dual-drive cabinet/power supply	$135	$5.00
Two drive cable	$39	$2.00
Fan drive cable	$49	$2.00
DOSPLUS 3.3	$99.95	$2.00
DOSPLUS 3.4	$149.95	$2.00

Prices are FOB Adelaide. Add $5.00 freight for single drive package, $10.00 for dual-drive package. Prices are in Australian dollars. Freight is road freight anywhere in Australia.

All items carry a 90-day parts and labour warranty. Repairs to be carried out in our Adelaide workshops.

MICRO-80

LEVEL 2 ROM

ASSEMBLY LANGUAGE TOOLKIT

by Edwin Paay

FOR TRS-80 MODEL 1, MODEL 3 AND SYSTEM 80/VIDEO GENIE

This is a new package consisting of two invaluable components:

- **A ROM REFERENCE** Manual which catalogues, describes and cross-references the useful and usable ROM routines which you can incorporate into your own machine language or BASIC programs.

- **DBUG**, a machine language disassembling debugging program to speed up the development of your own machine language programs. DBUG is distributed on a cassette and may used from disk or cassette.

Part 1 of the ROM REFERENCE manual gives detailed explanations of the processes used for arithmetical calculations, logical operations, data movements etc. It also describes the various formats used for BASIC, System and Editor/Assembly tapes. There is a special section devoted to those additional routines in the TRS-80 Model 3 ROM. This is the first time this information has been made available, anywhere. Differences between the System 80/Video Genie are also described. Part 1 is organised into subject specific tables so that you can quickly locate all the routines to carry out a given function and then choose the 'one which meets your requirements.

Part 2 gives detailed information about each of the routines in the order in which they appear in the ROM. It describes their functions, explains how to use them in your own machine language programs and notes the effect of each on the various Z80 registers.

Part 2 also details the contents of system RAM and shows you how to intercept BASIC routines. With this knowledge, you can add your own commands to BASIC, for instance, or position BASIC programs in high memory — the only restriction is your own imagination!

The Appendices contain sample programmes which show you how you can use the ROM routines to speed up your machine language programs and reduce the amount of code you need to write.

DBUG: Eddy Paay was not satisfied with any of the commercially available debugging programs, so he developed his own. DBUG: allows you to single-step through your program; has a disassembler which disassembles the next instruction before executing it or allows you to bypass execution and pass on through the program, disassembling as you go; displays/edits memory in Hex or ASCII; allows Register editing; has the ability to read and write System tapes and all this on the bottom 3 lines of your screen, thus freeing the rest of the screen for program displays. Four versions of DBUG are included in the package to cope with different memory sizes.

The best news of all is the price. The complete Level 2 ROM ASSEMBLY LANGUAGE TOOLKIT is only:

— Aus. $29.95 + $2.00 p&p
— UK £18.00 + £1.00 p&p

SPECIAL OFFER TO OWNERS OF THE LEVEL II ROM REFERENCE MANUAL ...

UPGRADE TO THIS ASSEMBLY LANGUAGE TOOKIT FOR ONLY $19.95!

Send back your original Level II ROM Reference Manual plus a cheque, money order or Bankcard authorisation for $19.95 plus $2.00 p&p and we will send you the new ASSEMBLY LANGUAGE TOOLKIT

MICRO-80