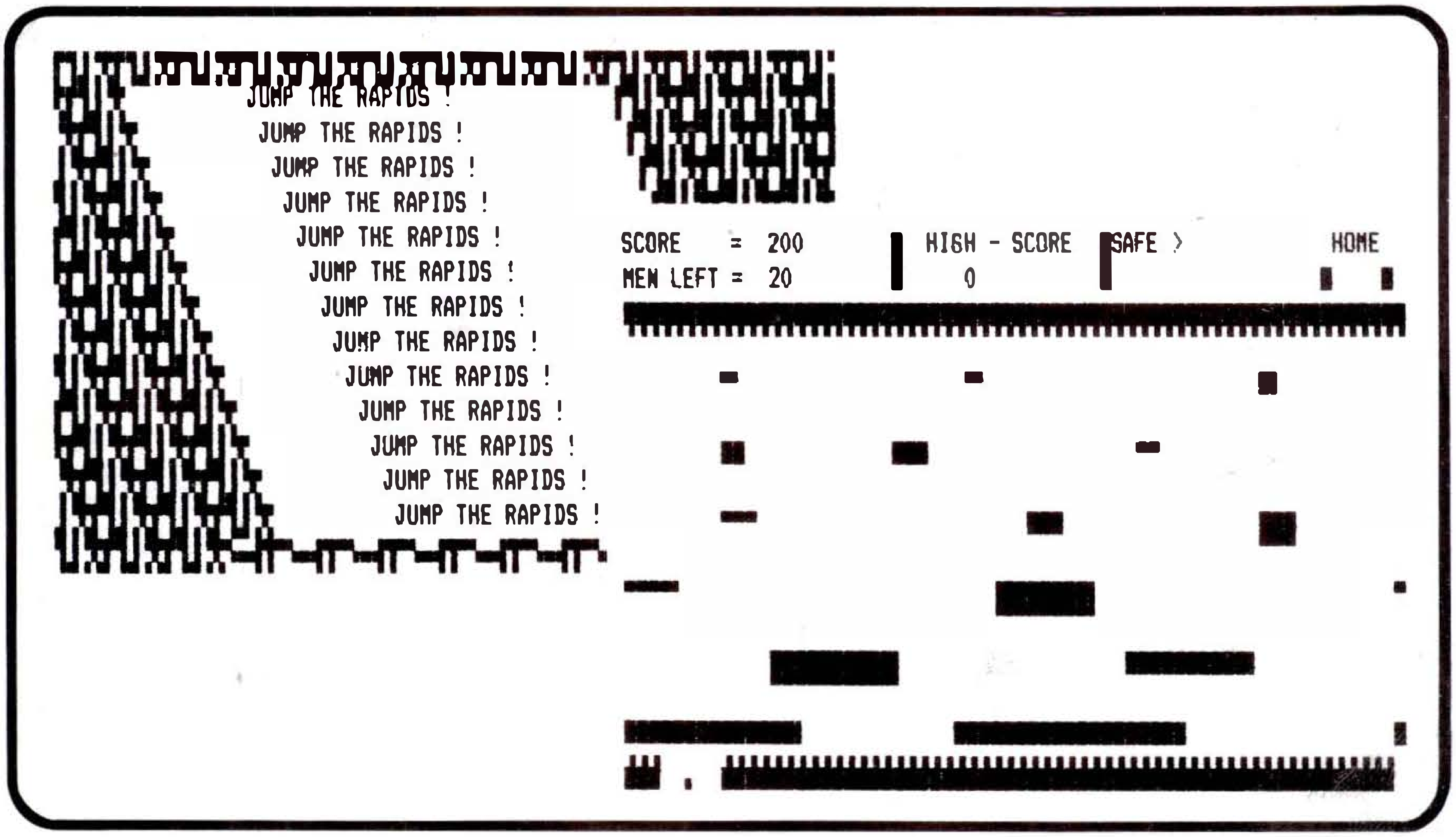
•TRS-80 •SYSTEM 80 •VIDEO GENIE
•PMC-80 •HITACHI PEACH
•TRS-80 COLOUR COMPUTER

Vol. 3, Issue 9, August 1982



```
JUMP THE RAPIDS !
 JUMP THE RAPIDS !
  JUMP THE RAPIDS !
   JUMP THE RAPIDS !
    JUMP THE RAPIDS !
     JUMP THE RAPIDS !
      JUMP THE RAPIDS !
       JUMP THE RAPIDS !
        JUMP THE RAPIDS !
         JUMP THE RAPIDS !
          JUMP THE RAPIDS !
           JUMP THE RAPIDS !
            JUMP THE RAPIDS !

SCORE     =  200     HIGH - SCORE   SAFE >        HOME
MEN LEFT  =  20           0
```

Also in this issue:

PROGRAMMING:
Less Basic BASIC

HARDWARE:
Clock Speed-Up for System-80
Add-On Keypad

SOFTWARE:
- FAULT FINDER — Level II
- CHEQUE-ACCOUNT MANAGER — Level II
- MORSE PRACTICE — Level II
- CHECK BOOK DATA FILE — Colour
- PAYROLL — 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
MICRO-80 is registered by Australia Post — Publication SQB 2207 Category B

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

EDITOR:	RYSZARD WIWATOWSKI
ASSOCIATE EDITORS:	
SOFTWARE :	CHARLIE BARTLETT
HARDWARE :	EDWIN PAAY

MICRO-80 is an international magazine devoted to the Tandy TRS-80 Model I, Model III and Colour microcomputers, the Dick Smith System 80/Video Genie and the Hitachi Peach. 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

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

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/Video Genie or Peach and its peripherals. MICRO-80 is in no way connected with any of the Tandy, Dick Smith or Hitachi 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 microcomputer to earn some extra income is included in every issue.

** CONTENT **

Each month we publish at least one applications program in BASIC for each of the microcomputers we support. We also publish Utility programs in BASIC and Machine Language. We publish articles on hardware modifications, constructional articles for useful peripherals, articles on programming techniques both in Assembly Language and BASIC, new product reviews for both hardware and software and we print letters to the Editor.

** COPYRIGHT **

All the material published in this magazine is under copyright. This means that you must not copy it, except for your own use. This applies to photocopying the magazine itself or making copies of programs on tape or disk.

** 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 *****

Competition among the larger computer distributors must be becoming more fierce. While making enquiries about possible new developments for 1983, I was greeted with tight-lipped, non-committal, but polite, responses. I'm afraid I don't react too kindly to this sort of treatment since it reminds me a lot of some frustrating experiences I have had as a customer.

The Dick Smith organisation, apparently have no intentions of pursuing Video-Genie III or colour computer look-alikes. On a slightly more positive note, they do have something else afoot but, as negotiations are at a delicate stage, could say no more. I suppose we will have to wait and see.

Tandy were able to confirm rumours about the new Model 12 - an upgraded version of the Model II featuring double-sided, double density slim line drives, expansion to 512K of memory, a green screen and the possibility of increased processing power with a MC68000 CPU board. The Model 12 has already been released in the U.S.A. priced at $US2,995. However, they could not verify information concerning the Model IV - supposedly an upgraded Model III with a green screen, high resolution graphics, a Z80A with 128K memory and dual double-density drives. It seems feasible as Tandy is already marketing high resolution 640 x 240 graphics options for the Model II, III and 16, but rather than speculate, we shall watch future developments.

Although Tandy have dropped their prices on the Colour Computer and Model III, there are other rumours that suggest more, but limited, price reductions are on the way. There may be some truth here if a range of new products is sloted for release.

On a more definite note, Tandy had announced a 5MByte Hard Disk Drive for the Models I and III which operates under LDOS 5.1.3 costing $3,699 for the primary drive. Logical Systems, the authors of LDOS, indicate that Tandy will also carry floppy disk based versions of LDOS as stock items. Perhaps this is true in the U.S.A. but there is no mention of these in the current Tandy computer catalogue. I suspect that Tandy may wish to sort out the minor incompatibilities between LDOS and TRSDOS before taking this step.

While on the subject of LDOS, a RAM-based version is currently being developed to reside below 3000H. This would suit machines that have only 64K of RAM with only a bootstrap loader in ROM, like the Model II and it may be implemented on the Model II. If so, then naturally this version should suit the Model 12 and 16 as well. Although I've not had much opportunity to explore LDOS 5.1.3, I can tell you that it works fine with the System 80. Perhaps what impresses me most so far is the amount of information and support provided by the supplier. This is, by no means, a simple DOS but certainly has much to offer the dedicated programmer and compares favourably with NEWDOS 80 Version 2.0.

But not be outdone, Apparat has a NEWDOS 80 Version 3.0 on the way, featuring a high degree of compatibility with TRSDOS and hard disk support. With the release of NEWDOS 80 Version 2.0, Apparat had announced that they would go their own way at the expense of compatibility, particularly with Model III TRSDOS which cannot be read at all by NEWDOS 80 Version 2.0. This makes life very difficult for the Model III owner and NEWDOS sales would have suffered because of it. This change of mind will, no doubt, increase its popularity among Model III owners.

The imminent Federal elections here in Australia may have repercussions that will affect the price of imported software in the areas of sales tax and customs duty. Already the viability of obtaining and distributing this software is questionable and supplies are becoming scarce. Coupled with rumours of price rises at the point of origin, the situation seems to be suspended in an atmosphere of uncertainty. Any dramatic price increases will lead to undesirable practices and to detrimental side-effects, since software piracy will, no doubt, increase and this can benefit no one. The software developers will find their efforts less rewarding and either increase prices further still or cease production altogether. This is something to be avoided at all costs as much of the high-quality software is written by those who have the experience of the mediocre software behind them. There is little we can really do but await the outcome.

- 0000000000 -

***** PEEKing (UK) (by Tony Edwards) *****

In the UK we have strong laws about advertising and the new colour computer, the Sinclair ZX Spectrum has been in trouble with them. As seems to be the case with many new computers there is a very long waiting time between ordering the machine and actually receiving it. Manufacturers seem to be over confident about the spped with which they can get a new machine from the prototype stage to the user and this causes many a long wait and some unhappy customers. With regard to the Spectrum some of those unhappy would-be customers have complained to the Advertising Standards Authority that the suggestion in the advertisements that the delivery time would be 28 days is untrue. The Authority has reacted by requesting that the publishers do not carry further advertisements until the promised 28 day delivery can be met. The figures quoted by the Authority are that of 31,000 orders only 9,000 have been dispatched and none of these within the 28 day period. However I bet that the Spectrum given to the Japanese by the UK Prime Minister, Mrs. Thatcher, during a recent official visit was delivered in less than 28 days. I hope it was not the one I ordered!

Amongst the rush of new machines which are trying to break into the hobby micro-market in this country one stands out as different, the JUPITER ACE. Z80A processor, 8K ROM, 3K RAM, 32 x 24 memory mapped screen, 256 x 192 pixel graphics, £90, 28 days delivery. Not much unusual there you may say but look at the operating language - FORTH!! This new machine does not support BASIC but breaks new ground in providing the home computerist with a chance to program in a new language.

In the early days of micro-computing, not long ago, micro users were restricted to the use of BASIC unless they ventured into the unknown of machine code. Now things are rapidly changing. Not only is the world of machine code now well charted, but an increasing number of alternative languages are being made available for use on micro-computers. Tandy were first on this road with their 'Tiny PASCAL' which allowed users with 16K '80 machines to have a chance at programming in a structured language. True, this small version of PASCAL has some restrictions, but it is powerful enough to allow the user to get a taste of this rather different language and to learn the glories and pitfalls of using a compiled code. Another language available for the '80 is BASEX. This is a sort of half way house between BASIC and machine code and it has some of the advantages of both, but also carries some of the disadvantages of both. Certainly it is worth exploring as an interesting exercise, but it is likely to be eclipsed by the versions of compiled BASIC which are now becoming available. For an application where BASIC is too slow and a compiled code is too cumbersome, it is certainly worth looking into. FORTH is a language that is causing much interest in the U.K. at the moment. The first micro to use FORTH as its main language is now available, and a version of this language for use on the '80 is now on sale at a very reasonable price. Whatever your views on the question of structured vs. unstructured languages, it is well worth trying your hand at one of the newcomers to gain experience.

- 0000000000 -

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

From: Mr. R. Burling - Armidale, N.S.W.

I enjoy your magazine and find the programs quite good except for those programs (machine language) that have reserved memory for 16K machines. I have a 48K system and would dearly love to be able to convert these programs to utilise the memory I have available to me, yet time really prevents me translating and retyping the thing into higher memory locations. Is there any program around that will convert such programs similar to the BASIC Line Renumber Program Tandy give with their computers?

(Thank you for the compliment. Unfortunately, relocating machine language programs and subroutines is a considerably more complex task than renumbering a BASIC program. Eddy tells me that the MON4 monitor purported to offer this feature but did not successfully cope with all types of machine code programs. Such programs can contain imbedded data (e.g. text strings, vector tables, etc.) which make the job a lot more difficult than simply changing all absolute address references. In fact, you would need to know so much about the program that you could generate the source code form, which could be very easily relocated by changing the ORG statement and reassembling. The problem is the same for BASIC programs that POKE Machine code subroutines into protected memory, although these are normally all code without the problem of imbedded data and could be relocated relatively easily.

My own experience has shown that the time necessary to relocate a program varies exponentially with its size and I, too, would appreciate an utility that could reliably relocate machine code programs. - Ed.)

From: Mr. M. Merrylees - Carrathool, N.S.W.

Having read some of your early editions of MICRO-80, I decided it would be a fund idea to try my luck as an author with one of my early efforts at programming.

I chose a BASIC game called "ANAGRAMS" - my computerised version of the popular age-old word game. I checked it over carefully, removing every bug in it, then cheerfully sent off the tape with an explanatory letter.

"Enthusiastic joy" is a mild description of the thrill I experienced as I received your parcel containing my tape returned with the assurance that following your receipt of detailed instructions for the program, you would print it gratefully.

So I compiled a concise set of instructions which, to the best of my knowledge, contained anything any person using the program would possibly need to know. I sent it off to you promptly, and resigned myself to await its arrival in print.

I wasn't at all worried when in the first three issues there was no sign of it, for I had anticipated the delay which seems mandatory for any article intended for publishing. This is quite normal with all magazines, and no apology is required for such incidents.

However, I was under thirteen years of age when I sent those instructions away, in complete confidence that they should be printed. I am now FIFTEEN and I still haven't seen it in print.... Need I say more?

I think I am justified (though please correct me if I'm wrong) in demanding at least a reason for such an action (or more appropriately "lack of action"), and feel that it would be common courtesy to at least let me know if you decide against printing my work, as I find it infuriating to wait and wait and still to receive no recognition of my work - even in the form of a polite refusal.

Therefore I should like a prompt acknowledgement of this letter giving some form of explanation for this incident, whether you finally are going to print my program, and if not, full details as to my present rights of copyright (i.e. whether I now again have full rights over my work).

I am sorry if I appear unnecessarily harsh, however I am quite frustrated over the matter and I hope that you have simply quite accidentally overlooked it before now. I am expecting this letter to be printed in the magazine and preferably with little editing.

(I understand your frustration at the delay in publishing your program. Our intention is and always was to publish it. The reason for the delay is simply because of the response from people to have programs published. We have so many stored up that some people like yourself end up waiting and waiting and waiting..... We can't do a lot about it, short of making each issue about 500 pages long to clear the backlog of programs. Of course, our publication of the magazine being so far behind has not helped things either. - Software Ed.)

From: Mr. R. Dyball - Woodberry, N.S.W.

I wonder if you might be able to supply me with some information on how, in the TRS-80 expansion interface the real time clock is worked, whether it is hooked directly to memory, or to a couple of output ports. I have found no mention of it in the TRS-80 tech. manual, or anywhere else for that matter.

(The real time clock in the TRS-80 Model I (and System 80) expansion interface is driven by the interrupt line to the Z80 processor. The hardware interrupts the processor every 25 mS (provided interrupts are enabled) and there must be a software interrupt service routine to process the interrupt. The Disk Operating System provides this routine which counts the number of interrupts and adjusts as necessary the 'time of day' value stored in memory locations 4Ø41H-4Ø43H.

There are available hardware add-on products that use microprocessor compatible clock-chips to keep time. These are real hardware clocks that are port-addressed but the actual address varies with the manufacturer. In addition, these usually provide battery-backup and do not lose time the way the software clock does when the Z80 processor runs with interrupts disabled for any length of time. - Ed.)

From: D. Bareis, Maryborough, Qld.

I would like to show readers how to make relocatable calls in machine language, as I am sure there must be a lot who would like to produce relocatable code but just don't know how to.

There was a good article in "The Assembly Line" May '81 80-Microcomputing which shows how to produce relocatable code, but their method destroys the HL register. I have worked out a way to find your position in memory without destroying any registers.

"CALL 13E5H" returns with the HL register on the stack and the return address in HL. To produce a relocatable call you must now use a relative jump to branch to your subroutine. The subroutine then needs to "INC HL" twice (to move address past your JR SUBROUTINE instruction) and then swap this address with the original value of HL (still on the stack) with an "EX (SP),HL". Listing two, a relocatable version of listing one shows how this is done.

Variables can be loaded in a similar manner. The method shown in listing three is best when you do not wish to use the IX or IY registers. Any variable must be at the end of the program (or at least after any reference to it).

Listing one

```
                    ........
                    ........
                    CALL SUBROU
                    ........
                    ........
SUBROU              CALL 1C9H           ;CLEAR SCREEN
                    RET
```

Listing two (a relocatable version of listing one)

```
                    ........
                    ........
                    CALL 13E5H          ;FIND POSITION IN MEMORY
                    JR  SUBROU
```

```
                                ........
                                ........
SUBROU                          INC   HL
                                INC   HL               ;MOVE HL PAST JR SUBROU
                                EX    (SP),HL          ;PUT RETURN ADDRESS ON STACK
                                CALL  1C9H             ;REST OF SUBROUTINE THE SAME
                                RET
```

Listing three (a relocatable version of "LD A,(VAR)")

```
                                ........
                                CALL  13E5H            ;FIND LOCATION OF POSN IN MEMORY
POSN                            LD    DE,VAR-POSN      ;FIND RELATIVE POSITION
                                ADD   HL,DE            ;HL POINTS TO VAR
                                LD    A,(HL)
                                POP   HL               ;GET HL OFF THE STACK
                                ........
VAR                             DEFB  0                ;VARIABLE AT END OF PROGRAM
```

(Thank you Mr. Bareis. Whilst we have not tried this technique ourselves it should work satisfactorily. We look forward to having some relocatable ml. programs submitted for publication, using Mr. Bareis' technique -Ed.)

From: Mr. I.J. Vanco - Herston, Qld.

I have a System 80 MkII - Business computer on which I am experiencing problems loading object tapes created by EDTASM + Ver. 1.08. On loading the tape in the system mode the computer will "accept" approximately one third of the program (program being about 3.5K) before it turns off the cassette and the system prompt appears. I have tried the corrected routine as detailed in Vol.3, Issue 7 with the result that the cassette is turned off almost immediately it is started. Has any one a solution?!!!

(Unfortunately, Mr. Vanco, no new information has come to light, other than the fact that some people have no trouble whatsoever while others like yourself continue to have problems loading the assembler output object files.

A 3.5K program is a very large machine language program that should take about ten minutes to load. I have found that large system tapes very rarely load successfully the first time, particularly as the system tape loader compares the checksum on each and every block and should so much as one bit be incorrectly read, then a checksum error occurs. The problem is compounded by the fact that the power-supply in the System-80 is operating close to its limits and any noise spikes on the mains are coupled to the cassette input circuitry invariably causing loading errors (even if you fit the MOV's as suggested in one of the Technical Bulletins). I found that in my case, this problem could be alleviated by turning off an old refridgerator that was on the same mains line as my computer.

In your case, the problem seems to have gotten worse by making the suggested modifications so it would seem advisable to remove the patches and restore it to its original form. It would also be wise to make sure that neither the cassette mechanicals nor some external source is responsible for the loading problems. Perhaps some other reader has found a solution and can help? - Ed).

From: Mr. J.D. Smith - Hawthorn, S.A.

First, thanks for a most interesting magazine!

The main point of my letter is to ask for a couple of little things which will increase the value of your magazine to your readers in general.

Would you please clearly indicate in published programs whether "POKEs" and "PEEKs" are used or not, and whether graphics are used?

And, equally important, would you please explain to those who submit programs (perhaps in an article) the value of writing programs in straight BASIC unless there is real value in using such things as those mentioned above?

Programs written general terms, without the above special techniques, have the great advantage that they are very simply adapted for other computers than the one for which the program was written. There are few reasons for "PEEKs" and "POKEs" apart from some increase in speed, and many programs for general use don't really require them. Further, "PEEKs", etc., immediately limit the program to the one machine, as there is no uniformity in computer memory locations and graphics. And if one sees an interesting program written for a Model I TRS-80 or some other machine, and one has a Model II, as I have, it is most frustrating to slowly read through a listing, decide there are no "PEEKs", etc., then find one after spending a couple of hours keying in a program! (I'm not a fast typist!)

So I'm sure your co-operation in this matter would be appreciated by many of your readers!

(Thank you for the kind words. The points you have made certainly deserve consideration not only from us here, but all our readers. I have included your request in the Reader's Request section and will consult with our Software Editor regarding the other matter. - Ed.)

- 0000000000 -

***** READERS' REQUESTS *****

This column is a regular feature of MICRO-80. In it, we list all those articles, programs, etc. requested by our readers. We invite contributions from readers to satisfy these requests and will, of course, pay a publication fee for all articles, programs etc. printed. As a guide, we will pay a minimum publication fee of $10 for any article or review published. In the case of software reviews, we will aim to pay in accordance with the value of the program, up to a maximum of $25. So, if you write a good review which we publish and the usual selling price of the program in Australia is $19.95, then we would pay you $20. In that way, the successful reviewer will get the program he reviews, free. (Make sure you include the selling price in your review). Unfortunately, we cannot afford that policy on hardware (!) so we will pay in accordance with the merits of the review - generally of the order of $25. Submission of a review for publication automatically means that you are prepared to accept the figure we decide to pay you and no correspondence will be entered into. Payment will normally be made within 30 days of publication.

** ARTICLES **

- File handling on the '80
- Reviews of '80 compatible printers
- Reviews of commercially available software (including that produced by us!)
- Reviews of commercially available hardware

* A master index to the appropriate sections in the Tandy Manuals in level I, Level II, DOS etc.

- Comparative reviews of disk drives
- An explanation of how to make full use of USR, PEEK and POKE statements
- Discussion of the various electric fields produced by the keyboard, tape recorder, monitor disk drives etc., how to measure them, how important they are and how to combat them
- The value of writing programs in straight BASIC

** SOFTWARE **

- Stock market program

* Horse racing system

* Morse code decoder

- Sub-routine Forum
- A new STAR-TREK game
- Programs of pharmaceutical interest such as Pharmacokinetics, Patient Medication Records, Drug information services etc.
- Multiple file tape backup (as in Australian Personal Computing Aug/Sept '81 for PET)
- A "PRINT SCREEN" command for the Model I
- A football game using graphics and sound
- Graphics car race as seen from the driver's position
- Tank game for a single player

** HARDWARE **

- Interfacing the '80 to external hardware
- Review on the performance of line filters
- Real Time clock

* Radio Teletype/Morse interfacing

- RFI (Radio Frequency Interference) suppression
- Fast tape storage
- High speed card readers, interfacing and documentation for the Model II.

NOTE: An * denotes that we already have some suitable material on hand for this topic.

COLOUR COMPUTER OWNERS

If you would like some specific information in the form of an article, a software or hardware feature, write and let us know. We will include your request in this section in the hope another reader can contribute the information.

- 0000000000 -

***** LESS BASIC BASIC - by Dr. G. Brokensha *****

Inspired by the excellent article by Ken Smith in MICRO-80 of April 1982 (Vol 3, Issue 5) I began thinking about the problem he posed and solved. The question was how to format the numbers 1 to 100 on the screen in ordered columns. The problem of deciding when to end a line was solved by this method (where I was the number between 1 and 100).

```
10 IF X/10 = INT(X/10) THEN PRINT
```

The use of this remaindering technique for deciding when to print on the next line is a good illustration of this very useful basic BASIC trick.

I got to thinking. Why not arrange the loops to do the decision-making for us? An outer loop for each line of ten and an inner loop for the items in each line. When we fall out of the inner loop we can print with line feed to begin a new line:-

```
10 FOR I=0 TO 9:'outer loop prints line
20 FOR J=1 TO 10:'inner loop prints values on line
30 K=I*10+J:'ten times outer loop plus inner loop = our number
40 IF K<10 THEN PRINT " ";:'pads out 0-9 with leading space
50 PRINT K;:NEXT J
60 PRINT:NEXT I
70 END
```

That's a reasonable solution. It's quicker too because comparisons are fairly slow in most high-level languages and BASIC is no exception. Timings averaged 20% faster with this routine. If you run this program you'll find that the final 100 at the end is misplaced. Furthermore, the test for the single digit succeeds only nine times out of a hundred - that's inefficient. Is there a better way, perhaps with more flexibility and general application?

One of BASIC's strong points is its powerful string handling commands. Since I cut my computing teeth without such aids I really enjoy using string-handling techniques. And so I started thinking about the problem another way. Each new number can be added to a string. When the string is full we can print it. But how to get the columns?

Try in comand mode:

```
PRINT STR$(1),STR$(12345)
```

...no help? How to right-justify the numbers? Why, use the BASIC string function RIGHT$ of course! Try:

```
PRINT RIGHT$("     "+STR$(1),5)+RIGHT$("      "+STR$(12345),6)
```

...and there you are. Both numbers printed in fields of six! This routine is a little slower, about the same as the original remaindering solution and it looks like this:

```
10 CLEAR 500
20 S$=STRING$(20," "):'a blank string for the leading blanks
30 FOR I=0 TO 9:'outer and inner loops again
40 FOR J=1 TO 10
50 K=I*10+J:'calculate our number again
60 A$=A$+RIGHT$(S$+STR$(K),4):'add spaces to beginning of number string and then select right
                               most 4 characters
70 '
80 '
90 NEXT
100 PRINT A$:'print string when ten numbers concatenated
110 A$="":'clear the string before using again
120 NEXT
130 END
```

Why bother with strings at all? No reason for this simple problem but the method has great advantages in flexibility. If you want beautiful formatted-screen output for your programs use strings but don't forget to use all your available memory for string storage by CLEARing plenty of space. This reduces the MICROSOFT garbage-collection delays.

In an idle moment try this routine. Its slightly greater complexity is amply repaid by its flexibility. It's as fast as our other solutions too.

```
10  CLEAR 500
20  INPUT "START";S
30  INPUT "FINISH";F
40  INPUT "FIELD WIDTH";FW:FW=FW-1:'BASIC function STR$ adds a space in front so we have to
                                    allow for it!
```

```
50  '
60  INPUT "NUMBER OF FIELDS PER LINE";NF
70  S$=STRING$(FW," ")
80  FOR I=S/NF TO F/NF:'set up loops
90  FOR J=0 TO NF-1
100 K=I*NF+J:'calculate current value to "print"
110 A$=A$+RIGHT$(S$+STR$(K),FW):NEXT:'add  it,  right-justified  in  field  of  width  FW,  to  string
                                    A$
120 '
130 PRINT A$:A$="":NEXT I
140 END
```

- 0000000000 -

***** MICROBUGS *****

Although we make every effort to ensure accuracy in the material we publish, inevitably errors and omissions will occur. In this section, we try to correct the bugs that have been found.

MICRO GRAND PRIX - Vol.3., Issue 7, June 1982 (pp25-29).

One of our readers, Mr. R. Gerstner, who typed in the program has noted three errors in the source listing.

Line 3790 should have a closing quote after the first one with 16 blanks in between them.

Line 2270 should read "LD BC,49" instead of 50 since MSGE15 at 3650 contains only 49 characters in the message.

Line 2230 should read "LD BC,39" instead of 40 since MSGE16 at 3660 contains only 39 characters in the message.

***** FREE SOFTWARE LIBRARY - VOL. 1 *****

There is an error on page 47 of the Free Software Library Manual Volume 1. This page lists replacement lines for the disk version of Household Accounts. A replacement line has been omitted from this page causing problems when the program is typed in from the book. This error does not affect the program as supplied on disk. The additional line number to be corrected for conversion of the program to disk operation appears below.

```
7 ON X GOTO8,38,29,33,26,43,77,90,95
```

- 0000000000 -

***** USERS' GROUPS *****

The following is a list of Users' groups of which we are aware. Many are interested in a wide variety of computers. For further information about any of the groups, contact the person indicated. If you have a group not mentioned here, please let us know so that it can be included in this section.

** AUSTRALIA **

AUSTRALIAN CAPITAL TERRITORY

CANBERRA GROUP

Cont: Bill Cushing,
10 Urambi Village, Kambah, 2902.
Meet: 3rd Thurs. monthly, 7.30pm.
Urambi Village Comm. Centre,
Crozier Circuit, Kambah.

NEW SOUTH WALES

BLUE MOUNTAINS OF N.S.W.

Cont: Greg Baulman
Tel: Home (047) 51 3221
Meet: 1st Fri. monthly, 7.30pm.
Springwood Civic Centre.

COMPUTERTOWN CAMDEN

Cont: Keith Stewart
P.O. Box 47, Camden 2570.

NEWCASTLE MICRO USERS GROUP

Cont: Dennis Jackson,
Tel: (040) 63 1910
Meet: Last Wed. monthly 7.30pm, Hall corner Fowler and Ogden Sts., Hamilton South.

TRS-80 SYDNEY EASTERN SUBURBS USERS GROUP MAPPER CP/M USERS GROUP

Cont: Dan Lawrence
G.P.O. Box 2551, Sydney, 2001.

WOLLONGONG GROUP

Cont: Paul Janson
P.O. Box 397, Dapto, 2630.

VICTORIA

BALLARAT COMPUTER USERS GROUP

Cont: John Preston, Tel (053) 31 4363
Meet: 2nd Tues. monthly, Chisholm College,
Frankston.

EASTERN SUBURBS USERS GROUP

Cont: John Fletcher
Tel: Home (03) 737 9544
Bus (03) 89 0677 (9-4)
Meet: 4th Wed. monthly, 7.00pm
Kingswood College, 355 Station St. Box Hill.

GEELONG COMPUTER CLUB

Cont: P.O. Box 6, Geelong, 3220.
Meet: 2nd Thurs. monthly, Tybar Engineering,
Hampton St. Newton.

MICROCOMPUTER CLUB OF MELBOURNE

Cont: MICOM, P.O. Box 60, Canterbury 3126
Meet: 3rd Sat. monthly 2.00pm. Burwood State College,
Burwood Hwy.

NORTHERN & WESTERN SUBURBS COMPUTER USERS GROUP

Cont: David Coupe, Tel: (03) 370 9590
Meet: CPM Data Systems, 284 Union Rd. Moonee Ponds
Alt. Thurs. 7pm.

PENINSULA COMPUTER GROUP

Cont: George Thompson, 3 Patterson St. Bonbeach,
3196. Tel. 772 2674.
Meet: 2nd Tues monthly, Chisholm College, Frankston.

QUEENSLAND

BRISBANE GROUP

Cont: Lance Lawes,
Tel: Home (07)396 2998
Bus. (07)268 1191 Ext. 15
Meet: 1st Sun. monthly, 21 Rodney St. Lindum

COMPUTER OWNERS GROUP

Cont: Betty Adcock, Tel: (07)263 4268

TOWNSVILLE GROUP

Cont: Townsville Amateur Radio Club
Meet: 2nd Tues. monthly, State Energy Serv. HQ,
Green St. West End.

SOUTH AUSTRALIA

ADELAIDE MICRO USER GROUP

Cont: Rod Stevenson, 36 Sturt St. Adelaide, 5000.
Tel: 515241 between 9-4

NORTHERN TERRITORY

DARWIN GROUP

Cont: Tony Domigan
P.O. Box 39086, Winnellie, 5789.

WESTERN AUSTRALIA

CPU - THE CLUB FOR PEACH USERS

Cont: Brendon Butcher, Tel: (09)367 5880

PERTH '80 USERS GROUP

Cont: C. Powell (09) 457 6849
Meet: 1st Tues. monthly, 7.30pm Comm. Rec. Hall,
MacDonald St. Yokine.

TASMANIA

DEVONPORT COMPUTER INTEREST GROUP

Cont: John Stevenson, Tel: (004)92 3237
First meeting, Mon. 18th April time and place
to be announced.

NORTH-WEST TASMANIAN USER GROUP

Cont: Rod McLeod,
Tel: Home (004) 372064
Bus (004) 301611

** UNITED KINGDOM **

COMPUTERTOWN NORTH-EAST

Cont: C/- 2 Claremont Pl. Gateshead, Co.
Tyne & Wear NE8 1TL.
Tel: 0632-770036/643417/679119/559167.

COMPUTERTOWN UNITED KINGDOM

Cont: Dave Tebbutt, C/- 14 Rathbone Pl.
London W1P 1DE

INTERNATIONAL TRS-80 LEVEL I USER GROUP

Cont: Mr. N. Rushton, 123 Roughwood Dr. Northwood,
Kirkley, Merseyside, L33 9U9.

NATIONAL TRS-80 USERS GROUP

Cont: Brian Pain, 40A High St. Stoney Stratford, Milton
Keynes.

NEWCASTLE PERSONAL COMPUTING SOCIETY

Cont: John Stephen Bone - 0632 770036

NORTH-EAST TRS-80 USERS GROUP

Cont: Barry Dunn, 8 Ethick Tce. North Craighead, Stanley,
Co. Durham DH9 6BE. Tel: 0207 30184.

NORTH WEST TRS-80 USERS GROUP

Cont: The Secretary,
40 Cowlees, West Houghton
Bolton BL5 3EG

TANDY OWNERS PROGRAM & INFORMATION CO-OP

Cont: Derek Higbee, 12 Shelley Close, Ashley Heath,
Ringwood.
Tel: Ringwood 6720.

TRS-80 EDUCATIONAL USERS GROUP

Cont: Dave Futcher - Head Teacher,
Beaconsfield First & Middle School,
Beaconsfield Rd, Southall,
Middlesex.

WEST HERTS 80 USERS GROUP

Cont: Terry Bradbury, 20 Spruce Way,
St. Albans, Herts. Tel: Park St. 73663.

** NEW ZEALAND **

AUCKLAND

Cont: Ron Feasy, Bus. 799366 Home: 469455
Meet: 1st Tues. monthly, 7.30pm
NZ Solenoid Co. Ltd., 28 Kalmia St. Ellerslie,
Auckland.

- 0000000000 -

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

We have had a number of enquiries from System 80 owners requesting instructions for modifying the clock frequency of their computer to speed up execution of programs and cassette saving/loading. Mr. R. Dyball of Woodberry, N.S.W. has sent us details of a modification he has developed and successfully installed on his own System 80 which speeds up the clock frequency from 1.79 MegaHertz to 2.6 MegaHertz ie. by 50%. This is possible because the Z80 is rated to work at 2.5 MHz and virtually any Z80 will function satisfactorily at higher frequencies than this. If your system is not reliable at 2.6 MHz then you could replace your Z80 with a Z80A which is rated for operation up to 4 MHz.

The modification described is not for those with little or no experience of electronics. WARNING - INSTALLING THIS MODIFICATION WILL VOID YOUR WARRANTY. You must use a low voltage, well-earthed soldering iron in carrying out this modification. Failure to do so can cause catastrophic and EXPENSIVE damage to the components in your computer. Although the author states that the same modification can be used on a TRS-80, and he includes in the article a cross-reference table for that purpose, in the letter accompanying the article he points out that he has not actually tested this out on a TRS-80. MICRO-80 advises readers that we have not tested out either of these modifications and can accept no responsibility for their performance. Readers should also note that Disk Drives should only be accessed to read or write with the clock frequency in the normal (1.79 MHz) mode. Now, over to Mr. Dyball.

** SYSTEM 80 CLOCK MODIFICATION - by R. Dyball **

The following article describes a means of changing the System 80's clock so that under software control the '80 can operate at either 1.79MHz or 2.6MHz This means that programs can run 50% faster and also the cassette can load and save programs at 500 or 750 baud. The same modification can be made to a TRS-80 if appropriate IC's are used (see table), note that this modification will void any warranty on both machines.

OPERATION

An unused section of Z38 (Z56) is used to divide by two the 5.3MHz clock (used by the video section) to give about 2.6MHz. The original clock frequency is the main 10.644MHz clock divided by 6, also in Z38. This version uses port 127 so that when address lines A0 - A7 have 127 on them and the $\overline{\text{OUT}}$ line goes low and D0 has a 1 on it, this is decoded by IC1 and IC2 so that the flip flop, IC4, is clocked over, this then goes to the simple data selector made up of 4 nand gates in IC5, enabling the 2.6MHz clock and disabling the 1.79MHz one. The reverse happens when a 0 is sent.

To read a value from the port, the $\overline{\text{IN}}$ line goes low, and since only D0 is connected either 254 (low speed), or 255 (high speed) is read.

You will need:
1 each - 74LS00, 74LS02, 74LS30, 74LS74, 74LS367, TTL IC's
Phillips screwdriver.
10-30W soldering iron (not scope type)
Solder, desoldering braid (Solder-wick)
Some perforated board, veroboard or a suitable PC board.
Fine plastic coated wire and (opt.) 4-6" of 8 way ribbon cable.

Firstly construct the circuit shown. I used a piece of perforated board and some low profile IC sockets, and wire-wrapped it, although you should be able to make up a small PC board from the diagram. To test out the board, either tin the wires and plug them into a breadboard, or tack them onto the board temporarily with the soldering iron. Put the A0 line low and A1 - A7 high, and connect the leds as shown. Next connect a 5v supply, (or 3 dry cells). Should any of the leds appear to be on too bright, check out the wiring. Now touch one of the clock input wires to ground. If this doesn't cause led2 to blink the other wire should. Next bring the $\overline{\text{IN}}$ line low, led1 should remain off, then taking off the $\overline{\text{IN}}$ line, bring the data line D0 high and $\overline{\text{OUT}}$ high, this loads a 1 into the flip flop IC4. Now bringing the $\overline{\text{IN}}$ low, led1 should light and touching the other clock input to ground will cause led2 to blink. If all works OK, then turn the System 80 over and undo the screw holding the lid on, remove the lid, and undo the screws holding the keyboard board down, carefully pull the wires out of the edge connector at the upper left. Locate Z38 (74LS92) and desolder pin 8, carefully lift the pin out of its hole in the board. Alternatively, you can remove the CPU board and cut the track leading from pin 8 of Z38 to pin 2 of Z37. Connect the board up as shown in Fig. 3. Solder a link from pin 9 of Z37 to pin 14 of Z38.

The easiest place to put the board, I found, was at the front near the gap between the CPU board and the interface board. The board can be bolted down to one of the nearby pillars or taped to the case.

Replace the keyboard, first checking that you haven't left any little bits of wire, globs of solder etc. lying around. Also, be careful when soldering the board that you don't short out any tracks. If everything looks OK, then turn the '80 on. Should the screen be filled with garbage, turn everything off and check the board for any wiring errors and so on. If everything works OK a "READY"? should turn up and everything should work as normal. The board is wired up so that on turning on the '80, the clock speed is normal (1.79MHz).

If you want high speed merely type: OUT127,1 or for low speed, type: OUT127,0 . To see what speed the '80 is at, either in the command mode or during a program: PRINT INP(127) should give either 254 or 255.

Should the modifications still not be working, you can always get the '80 back to its original state by removing all the added wiring and making a link from pin 8 of Z38 to pin 2 of Z37, or solder the gap in the track if you cut it, or if you can, put the pin back in its hole and resolder it in.

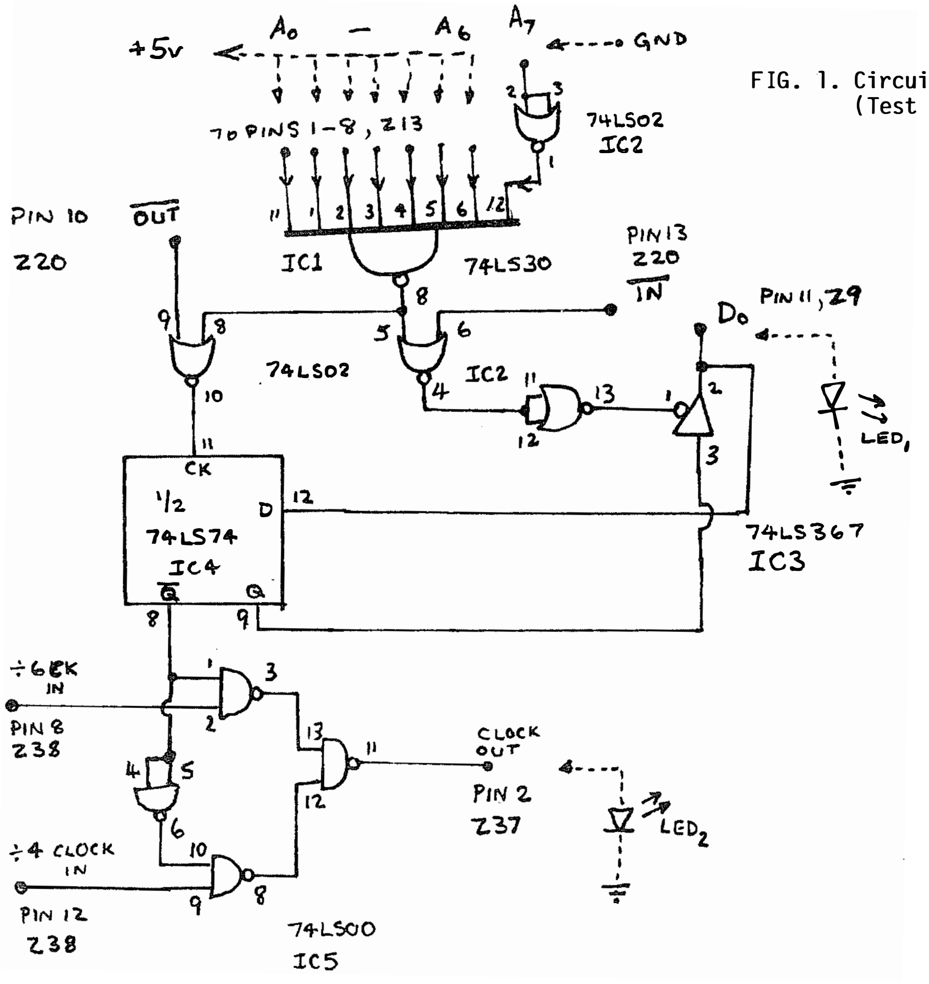
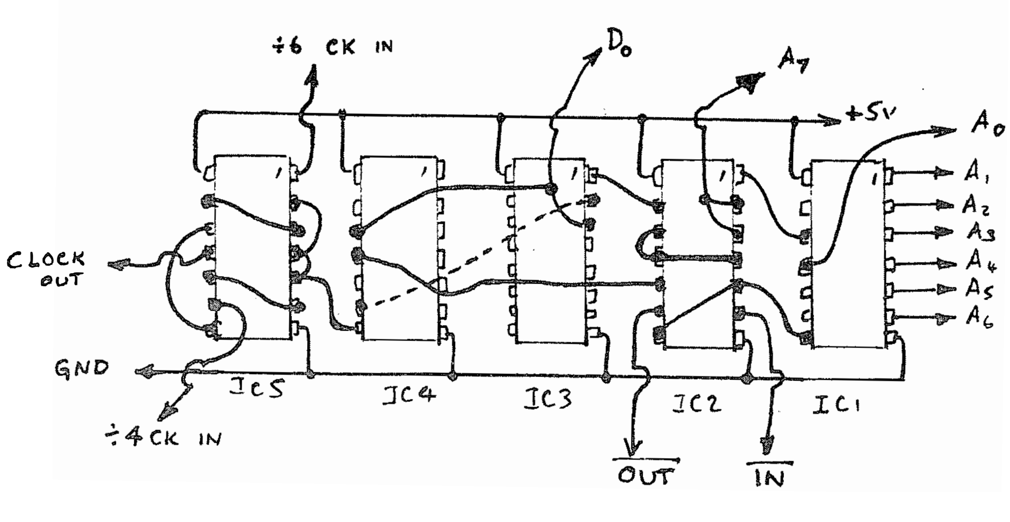


FIG. 1. Circuit Diagram of Modification
(Test connections shown dashed)

FIG. 2. Wiring on Circuit Board
(As viewed from BELOW)



N.B. You can use other output ports by rearranging the inverted address lines. Other data lines may also be used.

REFERENCES: SYSTEM 80 Technical Manual
TRS-80 Technical Manual.

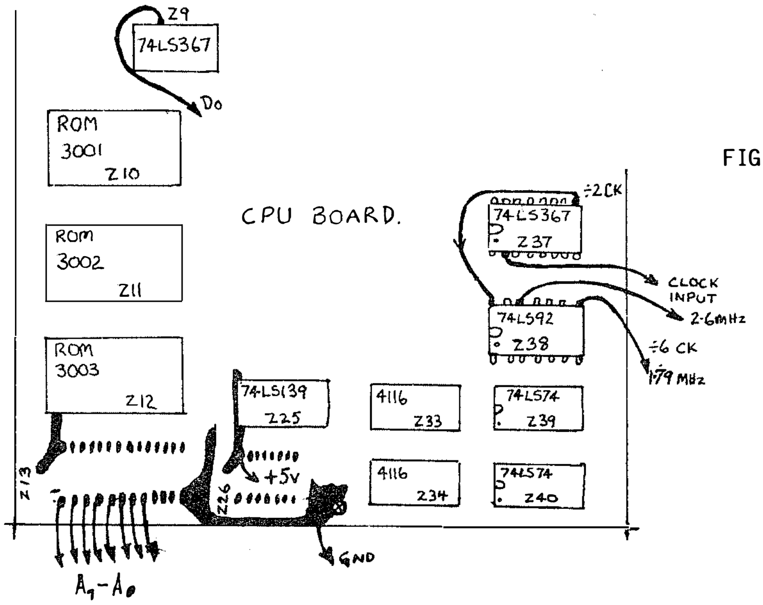


FIG. 3.(a) Positions of wires needed to hook up board (System 80 only). CPU Board Plan View.

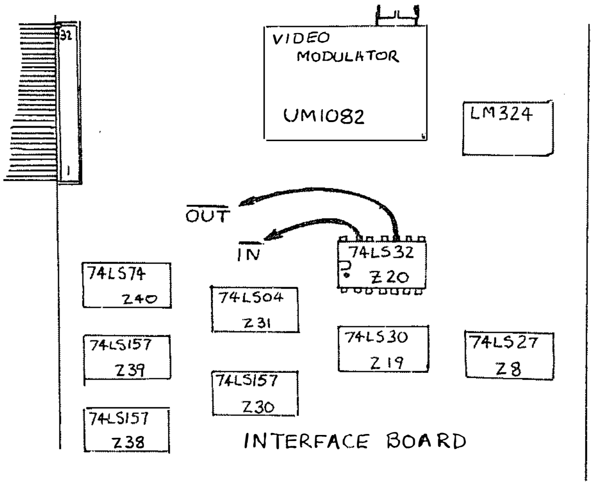


FIG. 3.(b) Plan view of Interface Board

CORRESPONDING IC's IN THE TRS-80

	SYSTEM 80		TRS-80
ck ÷6	Z38 (74LS92)		Z56 (pinout is the same, 74LS92)
CK ÷2	Z37,pin9 (74LS367)		Z70, pin 9 (74LS74)
D_0	Z9,pin11 (74LS367)		Z76,pin 11 (74LS74)
$\overline{OUT}$	Z20,pin13 (74LS32)		Z25,pin9 (74LS32)
$\overline{IN}$	Z20,pin10 (")		Z25,pin4 (")
A_0-A_7	vacan socket Z13	A_0	Z55,pin11 (74LS367)
		A_1	Z55,pin13 (")
		A_2	Z22,pin11 (")
		A_3	Z22,pin13 (")
		A_4	Z39,pin7 (")
		A_5	Z39,pin9 (")
		A_6	Z39,pin5 (")
		A_7	Z39,pin11 (")

- 0000000000 -

***** SOFTWARE REVIEW OF ESOS - by R. McDougall *****

A BASIC Operating System for the Stringy Floppy.

I always wondered why Exatron opted to use BASIC commands preceded by '@' in the TRS-80 Stringy Floppy (henceforth ESF) firmwave. Since the ESF was intended as a disk alternative, it would have been nice if they had used the disk- BASIC commands. Particularly since that approach would have made disk-BASIC software compatible with the ESF (and vice versa). Possibly the 2K ROM was insufficient to allow this. From the ASP ads -

> ".. emulate a full disk operating system with named files/
> passwords/directories/random records..",

the ESF BASIC Operating System (henceforth ESOS) seemed a panacea.

ESOS was written by Thomas Wheeler in 1980 and according to my User's Manual the current version is 1.4.

The Software

ESOS is distributed as file 1 on a 5' wafer. It is loaded and coped in the usual way. Aside from the gratuitous software supplied with the ESF at purchase, this was the first piece of software that I had bought for the ESF. I had the same problems trying to load ESOS that I had when I first tried to load the DATA I/O utility, but with persistence (and a lot of swearing) I succeeded in loading it and backed it up immediately.

Running ESOS initialises the appropriate (outlined below) L3 vectors, shuffles the start of the BASIC program area up to 4C34H and returns to the BASIC command line. ESOS resides between 42E9H and 4C34H, i.e. it occupies about 2K of RAM. It also requires a further 1000 bytes (directly above the BASIC program) to load in a wafer directory.

The BASIC commands available under ESOS are as follows:-

CMD "I"

Initialises a wafer directory, effectively destroying the previous contents. The ESOS directory is a 1000 byte file placed at the start of a wafer.

CMD "D"

lists the wafer Directory to the screen.

CMD "DL"

lists the wafer Directory to the screen with file Lengths. The name of each file is displayed along with its file number (ambiguously called its Logical Record Number in ESOS) on the wafer, its length in bytes and its maximum length in bytes (ambiguously called its Maximum Record Length). The total file space used in bytes (ambiguously called the Total Wafer Length) is also displayed.

SAVE "filespec"

Saves the resident BASIC program to wafer under the name filespec, which may be a string of any characters up to a length of 32. Anything after a period in a filespec is treated as a password by ESOS. The size of the saved file on wafer is, by default, the size of the program in memory. (See next command).

SAVE "filespec", maxlen

as before, but maxlen (in bytes) specifies the size of the file which the program will occupy on wafer, i.e. the maximum length in the directory listing. Subsequent saves of the same filespec with default file length will, provided the program is not longer than the maximum length originally specified, cause the old file to be overwritten by the new one. Its new length is reflected in the directory listing. This facility is useful during program development.

LOAD "filespec"

loads a BASIC or machine program into memory. Machine language programs under ESOS have the same format on wafer as before and may therefore be @LOADed. BASIC programs, however, are stored under an incompatible format. The filespec must be exactly the same as that used when the program was saved, otherwise a file not found error will result. The LOAD command can also be used from within a BASIC program to provide a chaining facility with automatic start of chained programs. Ironically, while the standard ESF provides variable integrity during chaining, ESOS does not. All variables arecleared by chaining another program.

RUN "filespec"

loads a BASIC program and automatically executes it. Useful in command mode.

PUT string, n

dumps a BASIC string variable, padded out to 255 bytes if necessary, to wafer as file number n.

LOF (n)

a function (invoked by e.g. A$=LOF(2)) which assigns the contents of file number n (provided it is 255 bytes long) to a string variable.

GET n, varlist

loads the contents of file n into the variables listed in varlist.

EOF(d)

a function (invoked by e.g. X=EOF(0)) which assigns the next free file (according to the directory) on the wafer to a variable ('d' is a dummy agrument).

KILL "filespec"

removes a file's entry from the directory. The tape freed by this command is unusable unless the file killed was the last in the directory.

SAVE# "filespec", start, length, autostart

saves a machine language program to wafer with specified starting address, length and autostart address, under the name filespec.

LINE frequency, duration

generates a tone of specified frequency and duration via the cassette port.

Since ESOS allows any character in a filespec, extensions (e.g. /BAS, /CMD) are impicitly valid. However, if used, they are treated as part of the filename. Consequently, if a file is saved with an extension, that extension must be specified at load time.

As mentioned before, ESOS also provides a primitive password mechanism, in that anything following a period in a filespec is not displayed in the directory listing. But, to load or kill that file, the same filespec that was specified during save (including password) must be used. Thus, the only difference between a protected file and an unprotected file is that part of the filespec is not displayed in the directory listing for a protected file.

Any ESOS command which references the directory (i.e. SAVE, LOAD, RUN, KILL, SAVE#) requires that the directory be first read into memory before the command is executed. Directory updates after SAVE and KILL are also mandatory. While this technique assures directory integrity, it also makes these commands even slower than their standard ESF counterparts. A system which provided separate commands to read and write the directory, while more user-error prone, would have been faster and certainly more versatile. For example, if files on a wafer were only temporary i.e. they would ordinarily be deleted at the end of a session, it would not be necessary to rewrite the directory before removing the wafer.

Program files in ESOS, as with the standard ESF, consist of variable length blocks. Each block is the length of the program, except for BASIC programs saved using the maximum length option. Storing data under ESOS is somewhat inconsiustent in that data files as such are not opened and closed as per disk-BASIC or even ESF DATA I/O. In fact, data files cannot appear in the directory. The commands PUT and GET and the function LOF are used to access data files, which are a standard 255 bytes in length. Being 255 bytes long allows the assignment of string variables to these files and vice-versa. The function EOF returns the next free file number (after the last program in the directory) for use in creating these data files. So much for disk-BASIC compatability! The only advantage of this data file arrangement over ESF DATA I/O is that each 255 byte record can be accessed randomly (which, using a contiguous medium, is a dubious one).

The Documentation

The manual supplied with ESOS is a 24 page long, plastic spined affair. It purports to be written in the same vein as the Tandy Level 1 User's Manual. Why do so many software authors insist on writing documentation in such a demeaning and often embarassing manner? Who needs statements like -

> "The computer stopped, and gave you the error message, which
> in effect says, 'no, no dummy -- there's no way for me to do that!'"?
>
> (ESOS Manual, Tom Wheeler
> His grammar and punctuation)

Don't misunderstand me. I appreciate David Lien's intentions in the Level 1 manual, but I think that it, and others like it, could be written in a more mature (and often more informative) style without detracting from its first-user friendliness. In any case, grammar and personal preferences aside, the ESOS manual takes the user through the various commands by the use of appropriate examples. A rather scant memory map is included along with diagrams (sketches) showing ESOS wafer organisation and directory format. What is lacking is a list of useful entry points and locations used by the system.

All in all, ESOS seems to perform its tasks as intended provided its limitations are realised. However, I think that it is highly unjustified to advertise it as emulating a full disk operating system. It just doesn't. Along with other ESF owners, I'm sure, I look forward to the utility which does.

Supplier:	ASP Microcomputers 797 Dandenong Road East Malvern, Vic., 3145
Requires:	Model 1, Level II + ESF
Cost:	$27 (approximately)

- 0000000000 -

***** SYSTEM 80 ADD ON KEYPAD - by Stuart McMinn *****

In the February issue of MICRO-80 there was reference to cheap add on numeric keypad under the Readers' Requests section. I have found a simple solution for around $10 - $15. The details described below are for a SYSTEM 80 and may be used for a TRS-80 if a similar decoding method is used on the end of the keyboard.

The keypad that I used was one that is readily available from Dick Smith Electronics (K-2010), and I made a circuit board to suit the decoding system. A length of 12 way ribbon cable and some telephone wire (for links) are the only other hardware needed for the modification.

A look at the existing hardware on the computer explains how easy it is to add a keypad. At the right hand end of the keyboard there is a set of solder pads which are connected to the computer's keyboard address system. The address lines used here are DK0 to DK7 and AK3 to AK6. From the technical manual it can be found that you can encode these lines as:

Numbers 0 to 9; Decimal Point; New Line, Operators +,-,*,/; and Backspace. Something not in the manual is that the keys F1 to F4 can also be encoded from the lines AK4 and DK4 to DK7. By using the correct sequence on the keypad the connection is simple and efficient.

Method of Construction

It is advisable to use Scotchcal or a set of transfers (D.S. cat H-5656) to reproduce the circuit board, both for neatness and accessibility in between tracks. If using the transfers to make the board, be sure to mark out or drill the screw hole positions before starting (drilling through a track can be messy). Another point to watch is the way in which the pad for key strike is made. Two methods may be employed to carry this out. The first is to use a .200" donut pad and carefully cut or scrape lines onto it (use a very sharp, pointy knife for this). The other is to bend the thin lines supplied in the transfer set into an approximate shape of the pad. WARNING: Be careful when using this method; these thin transfers tend to etch away also.

After etching the board 20 wire links are soldered into place (see overlay), by bending the links back along the track and soldering. The reason for this is that there are no peaks to stick up off the board and damage the rubber key matrix. Although not really essential to do this, it makes the board look much neater. After soldering on the links you only needed to solder on the hookup wire from the keyboard to the keypad. First of all, find a place to mount your keypad, then cut off enough wire plus about 25mm. Solder one end to the keyboard, give the ribbon a half-twist and solder the other end to the keypad.

You should now be ready to use your keypad.

The key pad should look like this:

F1	F2	F3	F4	
*	7	8	9	
/	4	5	6	
+	1	2	3	NL
-	.	0	BS	NL

Here are the keyboard addresses as viewed from the top of the keyboard at the R.H. end.

D	D	D	D	D	D	D	D	A	A	A	A
K	K	K	K	K	K	K	K	K	K	K	K
4	6	3	7	5	1	2	0	4	5	6	3

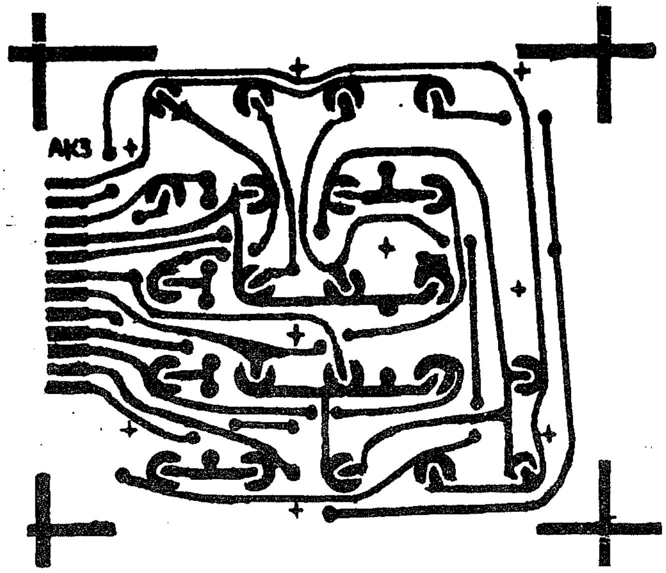
To mark the F1 to F4 keys use this method or similar:

1. Fill in engraved portion with paint (or similar) of colour of your choice.

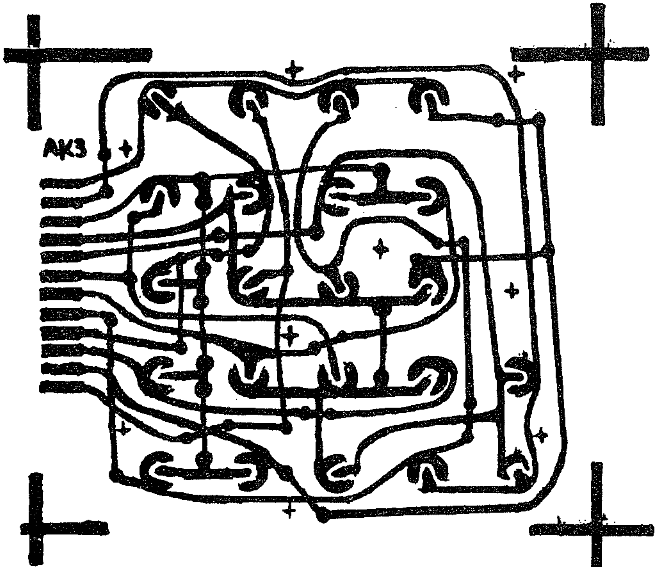
2. Give a light sand a second coat of paint.
3. Using E-Z transfers mark lettering
4. Apply 2-3 coats of clear lacquer.

The same method may be applied to the backspace and new line keys.

I hope that this modification is helpful to those with a lot of data processing.



The circuit board pattern
('+' indicates screw holes)



This shows the positions of
links (compare with above)

- 0000000000 -

* * * * SOFTWARE SECTION * * * *

***** CHECK BOOK DATA FILE - Peach and CC *****

The original version of this program (which appeared in the July '81 issue) was for a Level I, 4K machine. The Hitachi Peach has a much larger display area comprising 80 columns by 24 (or 25) lines while the Tandy Colour Computer has a text screen of 32 columns by 16 lines. These differences between the two video displays led to two different approaches to the conversion from Level I.

The objective for the Peach was to obtain a display that resembles the form of an actual bank cheque account statement. The ability to set a smaller scrolling window on the larger screen means that headings and prompts for information can be displayed outside this window and will therefore be unaffected by the transaction lines displayed in the scrolling window. To distinguish bank charges and deposits from cheques that are drawn, these must be denoted by the mnemonics

"%FEE" and "DEP" somewhere in the 'Particulars' field (The '%' sign was included to differentiate between an actual fee and a cheque made out to, say ACME FEEDS, for example). Up to thirty transactions can be saved to a cassette data file, but this number can be altered by modifying the DIM statement and the bounds check value in line 260.

In normal operation you load the most recent data file to recover the closing balance (as well as view the previous statement), enter new transactions and, lastly, save a new data file.

With the limited screen display of the Tandy, a different approach was taken where only transactions are displayed. To make deposits into the account, enter positive amounts, while for cheques written and fees charged, enter negative amounts.

A screen layout similar to a bank statement is not physically possible with the text screen of the Tandy. The cassette operation of both machines has been reliable which means that use of the program can be quick and easy. The more experienced could probably modify the program to work with disk and get even better reliability.

- 000000000 -

***** PAYROLL - Peach and CC *****

This program is a simple payroll designed to operate on a cassette system (a printer is optional). Originally published in the April '81 issue, the program has been modified to run on the colour computers, and most of Mr. Lawes' (the original author) explanation is still relevant and reproduced here.

The program caters for up to 20 employee records at a time. It will load from and save to cassette the year-to-date pay records for each employee in the batch. These records include:

- hours worked
- gross pay
- group tax
- 3 wage deductions
- net pay

and will handle weekly and hourly payroll.

The program handles simple payroll calculations. For complex calculations, work out your figures separately and enter them into the program manually, using the manual override functions. The tax calculation formula is based on rates applicable in Australia as at 1st July 1980. As tax rates change, new schedules may be obtained from the taxation office and the program amended and edited accordingly. The tax calculations commence at Line 750 (Tandy)/740(Peach). Overseas readers may like to restructure this part of the program to suit their own requirements.

The employee master file (the author's term for DATA lines entered directly into the program using line numbers 200-290 (Tandy), 210-300 (Peach)), contains the following data:

LINE NO.	PAY	NAME	HRLY RATE	WEEKLY RATE	TAX CODE
200 DATA	1,	A APPLE,	6.25,	20.00,	1

The example line above is exploded for clarity but is just a normal DATA line and would be entered into the program as:

```
200 DATA 1,A APPLE,6.25,20.00,1
```

IMPORTANT:DO NOT DELETE OR CHANGE LINE 300/310, it contains information used by the program during execution.

All data lines MUST contain 5 items, so if data is not applicable to your use of the program, the unwanted section must be filled out with zeros or blanks, e.g:

```
200 DATA 1,A APPLE,6.25,0,1
```

NOTE THE ZERO (O) ABOVE REPLACES THE UNWANTED FIGURE OF 20.00 IN THE PREVIOUS EXAMPLE.

Data input during program execution is simple, (normal BASIC program input as against the manual entry of data into program lines as previously described). Instructions appear on the screen in question-and-answer format. When all the data has been entered the user is asked to accept or reject the batch of data just entered.

The tax codes seen at the end of each line are selected from the following table:

1 = NO REBATE (single man's tax)

```
2 = $559 REBATE          (single parent tax)
3 = $800 REBATE          (married man's tax)
6 = NO EXEMPTION/NO REBATE
```

4 & 5 are used manually override the above or automatically access 1,2,3 and 6.

MAIN MENU

The program commences by displaying a menu on the screen as shown below:

```
PAYROLL CASSETTE BASED SYSTEM

1==) UPDATE MASTER FILE
2==) LOAD OLD FILE
3==) PROCESS PAYROLL
4==) SAVE NEW FILE
5==) EMPLOYEE INQUIRY

SELECT REQUIRED FUNCTION
```

Menu items 2 and 4 LOAD and SAVE your data file from cassette. Don't forget to take the precaution of writing down your tape file position and the number of records saved on the tape and remember to position the tape to the start of the data when reloading. Payroll input is a question and answer procedure and the computer invites the required responses. Menu item 5 accesses an employee's year-to-date earnings and will output to either the screen or a printer.

** EXAMPLE OF DATA ENTRY **

The screen prompts during data entry and a set of sample responses are shown below:

```
PAY NUMBER ?  1

EMPLOYEE NAME   : A APPLE

SELECT 1==) HRLY RATE:  2==) WKLY RATE:  3==) BOTH ? 1
HRS-ORD,TIME&HALF,DOUBLE  ? 40,3,3
```

(ie. 40 hours normal, 3 hours time-and-a-half and 3 hours at double time).

```
EQUIV ORD TIME : 50.5

SELECT 1==) AUTO:  2==) MANUAL RATE INPUT :
HOURLY RATE          :  6.25
HOURLY EARNINGS      : 315.625
GROSS EARNINGS       : 315.625

TAX CALCULATIONS
1==) No REBATE        3==) $800 REBATE        5==) AUTO CLC.
2==) $559 REBATE      4==) MANUAL OVERRIDE    6==) NO EXEM/REB
GROUP TAX DEDUCTED    : 78.35
DEDUCTION A :? 10
DEDUCTION B :? 5
DEDUCTION C :? 7.75
```

SAMPLE PAYROLL OUTPUT

The following is a sample of the output from this payroll program:

```
P/NO                 EMPLOYEE NAME                      DATE
  1                  A APPLE                            27/3/81
ORD.H  1.5HR  2.0HR  EQ/ORD  RATE    HRLY    WEEKLY     GROSS
40.00   3.00   3.00  50.50   6.2500  315.63  0.00        315.63
GROSS         GRP/TAX      DED/A     DED/B    DED/C      NET PAY
  315.63       78.35       10.00      5.00     7.25      215.03
-----------------------------------------------------------------
P/NO                 EMPLOYEE NAME                      DATE
  2                  B BRAVO                            27/3/81
ORD.H  1.5HR  2.0HR  EQ/ORD  RATE    HRLY     WEEKLY    GROSS
40.00   2.00   3.00  49.00   5.5000  269.50   0.00      269.50
GROSS         GRP/TAX      DED/A     DED/B    DED/C      NET PAY
  269.50       52.80       10.00      0.00     5.00      201.70
-----------------------------------------------------------------
P/NO                 EMPLOYEE NAME                      DATE
  3                  C CHARLIE                          27/3/81
ORD.H  1.5HR  2.0HR  EQ/ORD  RATE    HRLY     WEEKLY    GROSS
40.00   0.00   0.00  40.00   4.7500  190.00   0.00      190.00
GROSS         GRP/TAX      DED/A     DED/B    DED/C      NET PAY
  190.00       22.55        0.00      0.00     0.00      167.45
```

MICRO-80 PRODUCTS CATALOGUE

This catalogue contains a selection from the wide range of peripherals, interfaces, computers and software carried by MICRO-80 for your computer. If you don't see the item you want, contact us, we probably have it anyway!

MICRO-80 has been supplying customers throughout Australia and the Pacific region by mail-order for 2½ years. Our customers find this a simple and efficient way to do business. You may place your order by telephone or by mailing the order form from any issue of MICRO-80 magazine. Generally, it takes about one week from receipt of order until despatch. 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 2½-3 weeks.

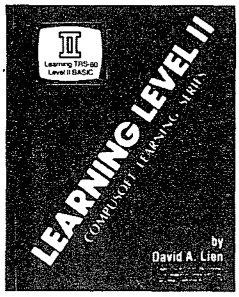
WARRANTY AND SERVICE

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 will be repaired by our own team of technicians in our Adelaide workshops.

TRADE-INS AND TERMS

MICRO-80 can accept your existing equipment as a trade-in on new equipment. We can also arrange consumer mortgage financing or leasing on larger hardware purchases. Contact us for details.

BOOKS



LEARNING LEVEL II

by David A. Lien

Written by the author of the Level I Users Manual, *Learning Level II* covers all Level II BASIC beyond Level I, plus much more. It shows you how to use the Editor, explains what the many error messages are really saying, and leads you through conversions of Level I programs to Level II.

Dual cassettes, printers, the Expansion Interface with clock and other features are explained in the same easy-to-learn style that made the Level I Manual famous. *Learning Level II* is an invaluable supplement to the TRS-80 Level II and System 80 manuals and is now only. $7.95 (plus $1.20 p&p).

BASIC FASTER AND BETTER AND OTHER MYSTERIES

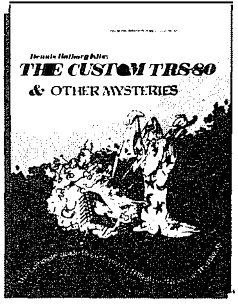
by Lewis Rosenfelder

Basic is not nearly as slow as most programmers think. *Basic Faster and Better* shows you how to super charge your BASIC with almost 300 pages of fast, functions and subroutines.

You won't find any trivial poorly designed "check-book balancing" programs in this book — it's packed with *useful* programs.

Tutorial for the beginner, instructive for the advanced, and invaluable for the professional, this book doesn't just talk . . . it shows how! Basic Faster and Better is $32.50 (plus $1.20 p&p).

BOOKS



THE CUSTOM TRS-80 AND OTHER MYSTERIES

by Dennis Bathory Kitsz

Ever wanted to do things to your TRS-80 that Radio Shack said couldn't be done? How about reverse video, high resolution graphics, and audible keystrokes?

Now enough? How about turning an 8-track into a mass storage device, making music, controlling a synthesiser, individual reverse characters, and a real-time clock just to name a few?

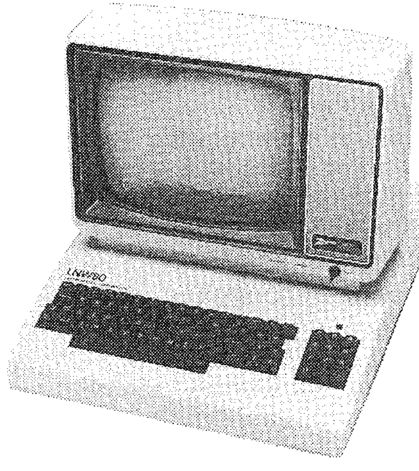
The *Custom TRS-80 and Other Mysteries* is packed with more than 290 pages of practical information and can be yours for only $32.50 (plus $1.20 p&p).

TRS-80 DISK AND OTHER MYSTERIES

by H.C. Pennington

TRS-80 Disk and Other Mysteries is the definitive fix-it book for disk users. More than 130 pages of easy to read, entertaining and immensely useful information. Find out how to recover disk files, the layout of information on disks, memory maps, problem solutions . . . the list goes on! Many readers have saved days of work by recreating disk files that were unreadable. *TRS-80 Disk and Other Mysteries,* which has received favorable reviews in several magazines, is yours for only $27.00 (plus $1.20 p&p).

THE LNW80 MkII MICROCOMPUTER



Manufactured in America by LNW Research Corporation, the LNW80 II has the following outstanding features:

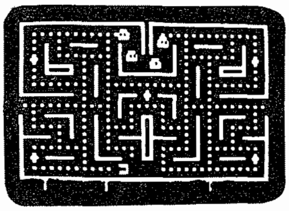
- Completely software and hardware COMPATIBLE with the TRS-80 Model 1.
- HIGH RESOLUTION COLOUR GRAPHICS — 4 MODES:
  - B/W LO-RES 128 x 48
  - B/W HI-RES 480 x 192
  - COLOUR LO-RES 128 x 192 in 8 COLOURS
  - COLOUR HI-RES 480 x 192 in 8 COLOURS
- CP/M Disk Operating System.
- Single and Double Density Disk Operation.
- Supports 5¼ inch or 8 inch Floppy Disk Drives.
- 48K RAM in TRS-80 mode plus 16K High Resolution graphics RAM.
- 64K RAM in CP/M mode plus 32K Banked in, usable in BASIC, plus the 16K High Resolution Graphics RAM.
- 4 MHz Z80A microprocessor — over twice the operating speed of the Model 1.
- HI-RES COLOUR (R-G-B) and B&W video outputs.
- 3 screen display modes:
  - 64 characters x 16 lines
  - 80 characters x 16 lines
  - 80 characters x 24 lines
- SOFTWARE SUPPORT

Apart from being able to run all TRS-80 Model 1 software and all CP/M software, there is also an extended BASIC interpreter available for the LNW80 II using most of the same commands as the TRS-80 Colour Computer but with full LNW Graphics Resolution, SET, RESET, POINT, LINE and CIRCLE as well as special commands to generate sound effects and tones. TRS-80 Colour Computer BASIC programs can be transferred to the LNW with only minor changes.

Prices include Sales Tax and are subject to change without notice. Prices are FOB Adelaide Add $20 road freight anywhere in Australia.
All equipment carries MICRO-80's Australia-wide 90-day warranty covering parts and labour.

The LNW80 II is the ideal computer for the serious hobbyist or businessman who is seeking a higher performance, more reliable computer to replace his TRS-80 Model 1 without sacrificing his investment in software or his programming experience. The LNW80 II uses standard Tandy or Tandy compatible disk drives. If you already have a disk TRS-80 system you may continue to use your existing disk drives on the LNW80 II.

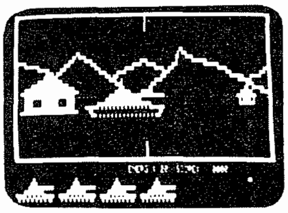
LNW80 II Computer — complete except for disk drives and monitor Includes: — CP/M Disk Operating System Dosplus 3.4 Double Density Disk Operating System — LNW Extended Colour Basic Interpreter	$2750 INC.S.T.
HI-RES Green Phosphor Monitor	$265 INC.S.T.
Super HI-RES Hitachi RGB Colour Monitor	$1250 INC.S.T.
Two Singlesided 40 Track Double Density Disk in cabinet with power supply and cable	$825 INC.S.T.



SCARFMAN

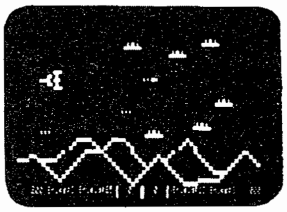
This incredibly popular game craze now runs on your TRS-80! It's eat or be eaten. You run Scarfman around the maze, gobbling up everything in your path. Try to eat it all before nasty monsters devour you. Excellent high speed machine language action game from the Cornsoft Group. With sound.

Price: $17.95



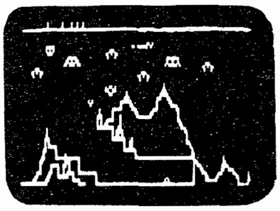
ARMORED PATROL

A realistic tank battle simulation. Your view is a 3--D perspective of an alien landscape. Maneuver your T-36 tank to locate and destroy enemy tanks and robots that lay hidden, ready to assault you. Clever graphics create the illusion of movement and dimension. From Adventure International. With sound. **Price $32.00**



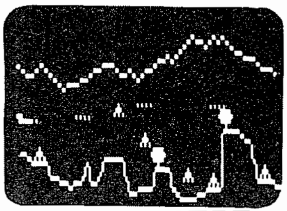
REAR GUARD

Deadly waves of enemy Cyborg craft attack your fleet from the rear. You are the Mothership's sole defender. You have unlimited firepower but the Cyborgs are swift, nimble attackers. Your abilities are tested hard in this game or lightening fast action and lively sound from Adventure International. **Price $26.50**



STRIKE FORCE

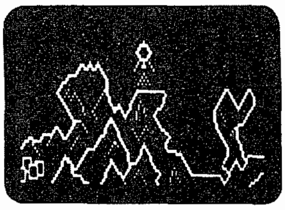
As the primary defender of a world of cities under deadly alien attack, your weaponry is the latest. rapid fire missiles, long range radar, and incendiary "star shells." Your force field can absorb only a limited number of impacts. A complex game ot strategy, skill and reflexes from Melbourne House. **Price: $26.50**



PENETRATOR

Soar swiftly over jagged landscape, swooping high and low to avoid obstacles and enemy missiles attacks. With miles of wild terrain and tunnels to penetrate, you're well armed with bombs and multiple forward missile capability. From Melbourne House. Features sound, trainer mode and customizing program.

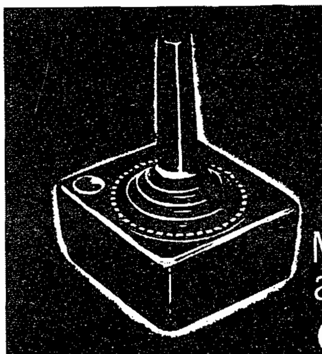
Price: $36.50



LUNAR LANDER

As a vast panoramic moonscape scrolls by, select one of many landing sights. The more perilous the spot, the more points scored -- if you land safely. You control LEM main engines and side thrusters. One of the best uses of TRS-80 graphics we have ever seen. From Adventure International. With sound.

Price: $26.50



STICKEROO JOYSTICK INTERFACE

for the TRS-80 MODELS I & III* and SYSTEM 80

FROM **$32.00**

ADD $2.00 p. & p.

CONVERT YOUR COMPUTER INTO AN ARCADE GAMES MACHINE

Micro-80's Stickeroo Interface Features:

•Compatible with Joysticks for Atari, Vic-20 and most video games •Saves your keyboard from abuse •Compatible with programs from leading US software houses: Big Five, Cornsoft, Melbourne House, Adventure International •Adds a whole new dimension of pleasure and fun to your favourite games •Will be supported in MICRO-80 •Can be used with your own basic or ML Programs •Comes complete, ready to plug in and use •Absolutely no modifications required to your computer

Due to popular demand, Stickeroo Interface is now available separately so you can use the Joystick of your choice.

PRICE INCLUDES ... STICKEROO + INSTRUCTIONS + DEMO PROGRAM LISTING

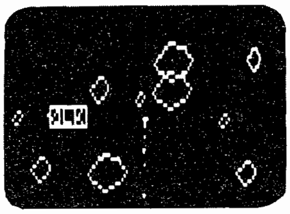
PLEASE SPECIFY TRS-80 MODEL I or III* OR SYSTEM 80 WHEN ORDERING

*Model III Interface available soon at slightly higher price, to be announced.

PISTOL GRIP JOYSTICK WITH FIRE BUTTON

$25 + $2 p & p (No p & p required if ordered with Stickeroo Interface)

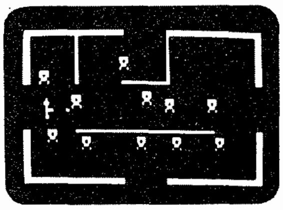
ALL GAMES ADVERTISED ON THIS PAGE ARE STICKEROO COMPATIBLE



SUPER NOVA

Asteroids float ominously around the screen. You must destroy the asteroids before they destroy you! (Big asteroids break into little ones). Your ship will respond to thrust, rotate, hyperspace and fire. Watch out for that saucer with the laser! As reviewed in May 1981 Byte Magazine.

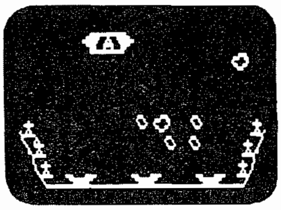
Price: $26.50



ROBOT ATTACK

Talks without a voice synthesizer, through the cassette port. With just a hand laser in a remote space station, you encounter armed robots. Some march towards you, more wait around corners. Careful, the walls are electrified. Zap as many robots as you dare before escaping to a new section. More robots await you

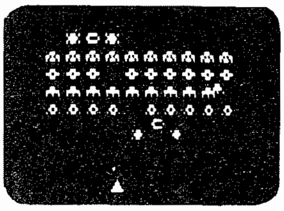
Price: $26.50



METEOR MISSION II

As you look down on your view, astronauts cry out for rescue. You must maneuver through the asteroids and meteors (Can you get back to the space station?) Fire lasers to destroy the asteroids, but watch out, there could be an alien Flagship lurking Includes sound effects!

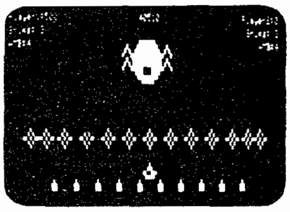
Price: $20.50



GALAXY INVASION

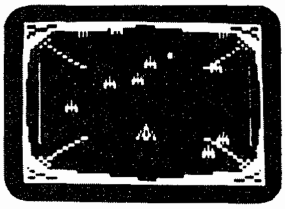
The sound of the klaxon is calling you! Invaders have been spotted warping toward Earth. You shift right and left as you fire your lasers. A few break formation and fly straight at you! You place your finger on the fire button knowing that this shot must connect! With sound effects!

Price: $26.50



DEFENSE COMMAND

The invaders are back! Alone, you defend the all important nuclear fuel canisters from the repeated attacks of thieving aliens, repeatedly. An alien passes your guard, snatches a canister and flys straight off. Quick! You have one last chance to blast him from the sky! With sound and voice.



STELLAR ESCORT

The latest super action game from Big Five. As the Federation's top space fighter you've been chosen to escort what is possibly the most important shipment in Federation history. The enemy will send many squadrons of their best fighters to intercept. With sound.

Price: $26.50

THE BEST IN ENTERTAINMENT FROM AMERICA'S TOP SOFTWARE HOUSES

MICRO-80 now has in stock some of the best games and adventures written for the '80s. These programs are supplied on cassette for the Level II/16K TRS-80 Model I (or III). They are also suitable for the System 80 but sound may not be available unless a hardware modification has been fitted to reverse the roles of recorders #1 and #2. Limited stock is available at these prices.

FROM BIG FIVE

COSMIC FIGHTER **$20.95**

Your ship comes out of hyperspace under a convoy of aliens, you destroy every one but another set appears, these seem more intelligent. You eliminate them too. Your fuel supply is diminishing. You must destroy 2 more sets before you can dock — includes sound effects.

ATTACK FORCE **$26.50**

In this fast paced, m/l game 8 alien ramships are warping towards your ship. You must dodge them and fire your missiles before they destroy you — but watch out for the flagship and its death beam!! — complete with sound effects.

FROM ADVENTURE INTERNATIONAL

ELIMINATOR **$26.50**

Your mission is to prevent the marauding alien hoards from recovering your energizers from the planet surface. There are several types of alien ships — each with different weapons to destroy you!! — with sound effects.

PLANETOIDS **$26.50**

It's your ship against a swarm of killer planetoids, as you try to destroy them before they destroy you — with sharp graphics and sound effects.

ADVENTURELAND **$26.50**

Wander through an enchanted world trying to recover 13 lost treasures. You'll encounter wild animals, magical beings, and many other perils and puzzles. Can you rescue the Blue Ox from the quicksand? Or find your way out of the maze of pits?

PIRATE'S ADVENTURE **$26.50**

"Yo ho ho and a bottle of rum . . ." Meet the pirate and his daffy bird along with many strange sights as you attempt to get out of your London flat and get to Treasure Island. Can you recover Long John Silver's lost treasures?

MISSION IMPOSSIBLE **$26.50**

Good morning, your mission is to . . . and so it begins. Will you be able to complete your mission in time? Or is the world's first automated nuclear reactor doomed? This is **hard**. There's no magic and no help this time, but plenty of suspense. Good luck!

VOODOO CASTLE **$26.50**

Count Cristo has had a fiendish curse put on him by his enemies. There he lies, with you as his only hope. Will you be able to rescue him or is he forever doomed? Beware the Voodoo Man . . .

THE COUNT **$26.50**

You wake up in a large brass bed in a castle, somewhere in Transylvania. Who are you, what are you doing here, and WHY did the postman deliver a bottle of blood? You'll love this adventure, in fact you might say it's Love at First Byte.

STRANGE ODYSSEY **$26.50**

Marooned at the edge of the galaxy, you've stumbled on the ruins of an ancient alien civilization complete with fabulous treasures and unearthly technologies. Can you collect the treasures and return home or will you be marooned forever?

MISSILE ATTACK **$20.50**

This is a real-time game with sound effects. You must protect your cities against enemy missiles, as your skill increases, so does the level of difficulty making accuracy a must.

SPACE INTRUDERS **$26.50**

A very fast game with the deluxe version of Space Invaders, complete with "spitting" invaders and the SOS of escaping aliens — with sound effects.

MYSTERY FUN HOUSE **$26.50**

Can you even find your way in to the Strangest Fund House in existence let alone find your way completely through it or will you get kicked out when the park closes?

PYRAMID OF DOOM **$26.50**

An Egyptian Treasure Hunt leads you into the dark recesses of a recently uncovered Pyramid. Will you recover all the treasures or more likely will you join its denizens for that long eternal sleep?

GHOST TOWN **$26.50**

Explore a deserted western mining town in search of 13 treasures. From rattlesnakes to runaway horses, this Adventure's got 'em all! (Also includes new bonus scoring system).

SAVAGE ISLAND **$26.50**

Part 1 — A small island in a remote ocean holds an awesome secret. Will you be the first to uncover it? **NOTE:** This is the first part of a larger adventure. It will be necessary to buy further tapes to complete the entire Adventure. **WARNING: FOR EXPERIENCED ADVENTURERS ONLY!**

SAVAGE ISLAND **$26.50**

Part2 — After struggling through Part 1, you have the consolation of knowing it's half over. This concludes the two part Adventure. It requires you have completed Part 1 and received the password to start Part 2.

GOLDEN VOYAGE **$26.50**

WARNING: For Experienced Adventurers Only! The King lies near death in the royal palace—you have only three days to bring back the elixir to cure him. Journey through the lands of magic fountains and sacred temples, stormy seas and gold, gold, GOLD!

BUY YOUR MODEL 3 FROM MICRO-80 AND SAVE $00'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 supplies to avoid screen jitter and overheating. The disk controller boards used incorporate special compensation circuitary for 80 track disk drives and may also be used to run 8 inch disk drives with an appropriate cable and DOS.

MODEL 340 **$3130**

2 40 TRACK SINGLE-HEAD DISK DRIVES GIVING
350K FORMATTED STORAGE, 48K RAM

MODEL 340+ **$3350**

2 40 TRACK DUAL-HEAD DRIVES GIVING
700K FORMATTED STORAGE, 48K RAM

MODEL 500 — 5+ MEGABYTE MODEL 3 **$5895**

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,
DOSPLUS 4.0 DISK OPERATING SYSTEM

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.

WINCHESTER DISK DRIVE SUB-SYSTEM **5MByte $2995**
10MByte $3750

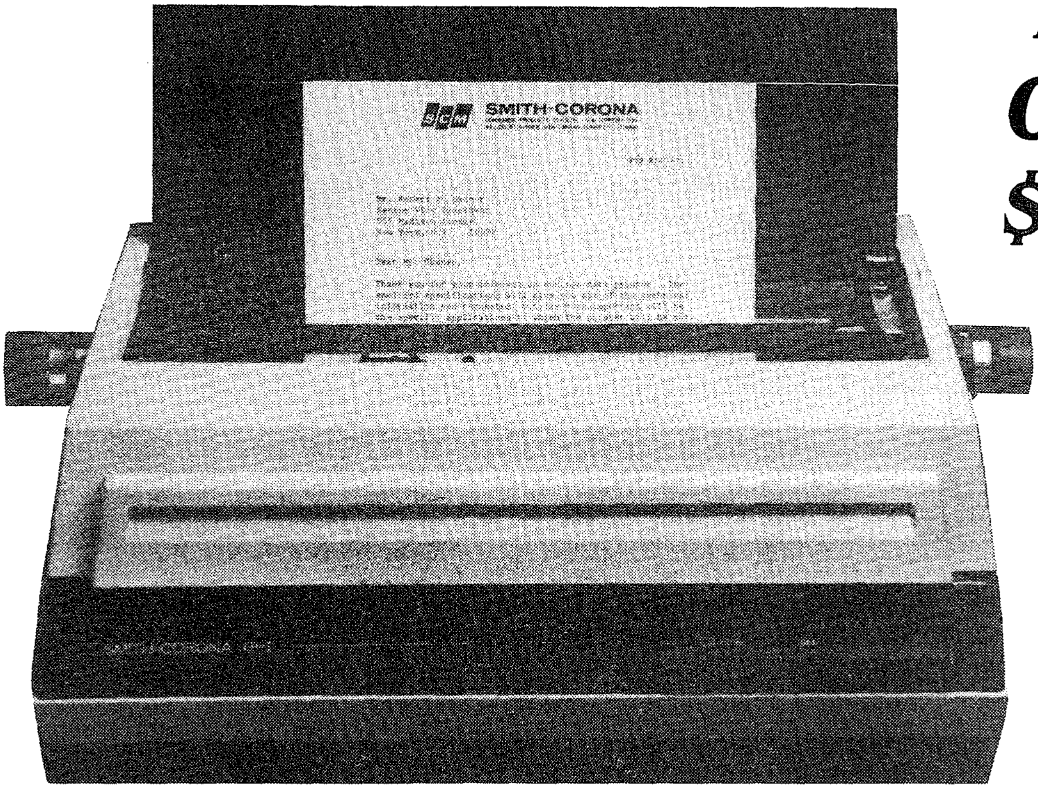
This Winchester Disk Drive sub-system provides either 5 or 10 Megabyte of reliable, high speed storage. It connects to any standard Model 3 equipped with one or more floppy disk drives and does not void the Tandy warranty. Complete with DOSPLUS 4.0 Disk Operating system.

Prices include Sales Tax and are subject to change without notice. Prices are FOB Adelaide. Add $20 road freight anywhere in Australia. All computers and peripherals carry MICRO-80's Australia-wide. 90-day warranty covering parts and labour.

NEW

THE SMITH-CORONA TP-1 DAISY WHEEL PRINTER

A low-priced letter quality printer



For Only $999

Ideally suited for small businesses or the home user, the TP-1 is a microprocessor controlled, correspondence quality printer that prints fully formed characters at an average print speed of 12 characters per second. This simple to operate, compact printer is compatible with most microcomputers and comes with the standard Centronics parallel interface (an optional serial data interface is available) and features:

- 128 ASCII Character Set (88 printable)
- 10 CPI or 12 CPI character spacing
- 105 characters per line (or 126 in 12 pitch)
- Handles letter and legal sized paper (up to 13″ wide)
- Variable line spacing and impression control
- Prints original plus up to three copies

Prices include Sales Tax and are subject to change without notice. Prices are FOB Adelaide. Add $20 road freight anywhere in Australia. All equipment carries MICRO-80's Australia-wide 90-day warranty covering parts and labour.

DISK OPERATING SYSTEMS & DEVELOPMENT SOFTWARE

You can increase your programming productivity, the execution speed and 'user friendliness' of your programs by using an enhanced Disk Operating System (DOS). Together with the other utility software, you can get the most from your disk drives.

DOSPLUS 3.3 $99.95

(Specify Model I single or double density or Model III)

An economic DOS intended for the first-time user and requiring single-sided disk drives. (The TRSDOS & DISK BASIC MANUAL is required to supplement the DOSPLUS manual).

DOSPLUS 3.4 $149.95

(Specify Model I single or double density or Model III)

With a high degree of compatibility with TRSDOS, DOSPLUS 3.4 supports single- or double-sided, single or double density, 5" or 8" disk drives with any track count (up to 96). Suitable for the first-time or experienced user wanting a fuss-free, bug-free, easy to understand but very powerful DOS which supports variable length records up to 255 bytes long. Comes with a stand alone manual.

ENHBAS $52.95

ENHBAS adds over 30 new commands and functions to your BASIC interpreter including high speed SORT, labels in BASIC, RESTORE to any line number, WHILE-WEND for structured programming, SCROLL, LEFT, INVERT, DRAW and PLOT to give you ease of control over graphics, SOUND and PLAY to add realistic sound effects and many more. Makes programming a breeze! Available for Model I or III, disk or cassette — specify which when ordering.

NEWDOS 80 VERSION 2.0 $169.00

(Specify Model I or Model III)

Newdos 80 suits the experienced user who has already used TRSDOS, understands the manual and is prepared to learn the somewhat complicated syntax of one of the most powerful DOS's available. With the correct hardware, Newdos 80 supports any mix of single- or double-sided, single or double density, 5" or 8" disk drives with track counts up to 96. It provides powerful, flexible file handling in BASIC including variable length records up to 4096 bytes. Definitely not for the beginner.

MASTER DISK DIRECTORY $20.95

FIND THE PROGRAM FAST!! PAYS FOR ITSELF BY RELEASING REDUDANT DISK SPACE!! MASTER DIRECTORY records the directories of all your individual disks onto one directory disk. Then it allows you examine them, find an individual file quickly, list files alphabetically, weed out redundant files, identify disks with free space, list files by extension, etc., etc. This program is invaluable for the serious disk user and will pay for itself many times over.

THE FLOPPY DOCTOR/MEMORY DIAGNOSTIC

Model I Disk **$36.50** Model III Disk **$43.50**

THE MICRO CLINIC offers two programs designed to thoroughly check out the two most trouble-prone sections of the TRS-80 — the disk system (controller and drives) and the memory arrays. Both programs are written in Z80 machine code and are supplied together on diskette for a minimum 32K, one disk system. Specify Model I or Model III.

MORE ENTERTAINMENT SOFTWARE

ADVENTURE HINT BOOK $10.95

If you can not go any further this will give you clues that may help — written by Scott Adams for Adventures 1—9.

LABYRINTH $26.50

Labyrinth — you move through a gigantic labyrinth and scattered through this nightmare are a multitude of objects and obstacles. A minotaur prowls the corridors — you must kill it before it kills you, Labyrinth has over 550 locations — be patient.

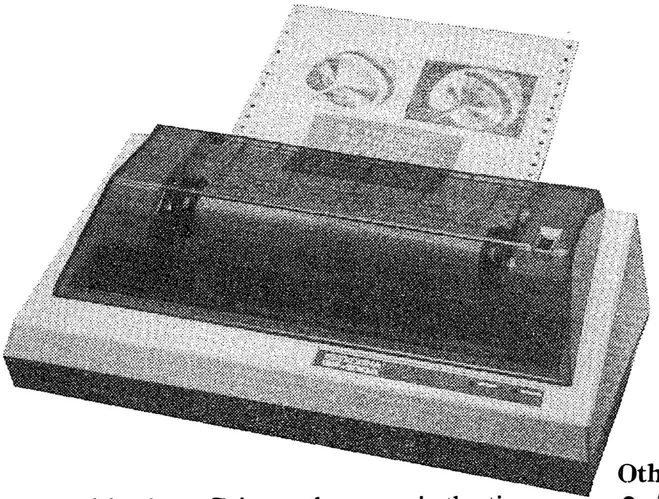
ASYLUM $26.50

Asylum places you in a cell, you have to escape. It's harder than it sounds, lots of hazards will be encountered.

DEATHMAZE 5000 $26.50

Deathmaze 5000 is another 3-D adventure. You move through a 5 storey building — your goal is to leave the deathmaze alive.

New
SEIKOSHA GP-100A GRAPHICS PRINTER



PRICED AT ONLY $475

If you have delayed buying a Printer, then now is the time to reconsider. The Seikosha has been designed for simple operation and puts full dot addressable graphics at your command. You can repeat a column of data as many times as you like with just one command. Double-width character and dot addressable positioning are software controlled.

Other features:

- Automatic Printing avoiding data loss when the maximum line length is exceeded.
- Allows mixing graphics, regular and double width characters on the same line.
- Up to 50 characters per second.
- Standard Centronics type Parallel interface.
- Self-test mode.
- Optional RS-232-C Serial Interface.

PRINTERS GALORE AT UNBEATABLE PRICES

MICRO-80 has a range of printers to suit every requirement from dot-matrix to correspondence quality daisywheel. Choose 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*
EPSON	MX-80III	DM	80	80	Y	FULL	F/T	HI-RES	P	1	$ 999	$ 8.35
EPSON	MX-100III	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	$ 999	$ 8.35
C ITOH	M1550	DM	132	120	Y	FULL	F/T	HI-RES	P	1	$1499	$12.54
OLIVETTI	PRAXIS35	DW	100	6	N	FULL	F	NO	P	1	$ 895	$ 7.49
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

NOTE: The following symbols are used:

TYPE+ DM = DOT MATRIX
DW = DAISYWHEEL
BI DIRECTIONAL Y = YES N = NO
LOWER CASE FULL — means Lowercase descenders to below line
PAPER FEED F — means Friction Feed
T — means Tractor Feed
F/T — means both Friction and Tractor Feed included in 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

MICRO 80 PRODUCTS

Prices include Sales Tax and are subject to change without notice. Prices are FOB Adelaide. Add $20 road freight anywhere in Australia. All equipment carries MICRO-80's Australia-wide 90-day warranty covering parts and labour.

```
PAGE TOTAL FOR 27.3.81
ORD.H  1.5HR  2.0HR   EQ/ORD    RATE    HRLY    WEEKLY  GROSS
 120     5      6     140                               775.13
GROSS       GRP/TAX     DED/A       DED/B    DED/C      NET PAY
  775.13     153.70     20.00        5.00    12.25      584.18
```

As a final reminder, you should remember that each time you change the master file data you are changing the program itself and it will therefore be necessary, in addition to saving your program data, to also save the program itself onto another tape to preserve the changes.

- 0000000000 -

***** JUMP THE RAPIDS LII/16K m.l. - by B. Green *****

The object of the game is to get ten men safely across the rapids. Your men will be positioned one at a time on the nearest bank, using the (J) jump key, jump from log to log until you reach the opposite bank. You have twenty men - each man is worth 10 points. You gain points for each man that gets safely across. The closer a man jumps to the HOME position, the more points he scores, though if he goes off the screen you lose him. Each man lost costs you 10 points. You must get at least ten men across to qualify for a score.

This program uses the machine language program MOVIE (published in the September 1981 issue) to drive the graphics and, for convenience, it has been included on the cassette and disk editions.

To LOAD from cassette:

```
SYSTEM    (ENTER/NEWLINE)
MOVIE     (ENTER/NEWLINE)
/         (ENTER/NEWLINE)
```

and then CLOAD the BASIC program of Jump the Rapids.

For disk users, the MOVIE/CMD file on the disk has an ORG of 8ØØØH and to LOAD it type:

```
LOAD      MOVIE/CMD (ENTER/NEWLINE)
BASIC               (ENTER/NEWLINE)
SYSTEM              (ENTER/NEWLINE)
/32768              (ENTER/NEWLINE)
```

The program will display a copyright message and you can start by typing

```
RUN"RAPIDS/BAS"     (ENTER/NEWLINE)
```

- 0000000000 -

***** FAULT FINDER LII/4K - by N.E.L. Rossiter *****

Fault Finder is a problem in deductive logic. The scenario set is of a machine (fortunately fictitious) which is subject to three particularly undesirable faults when combinations of actions are taken. The computer will act as your agent in performing the actions that you order, and reporting the results. You are required to determine what the common features of combinations of actions which produce the faults are, and hence, which actions must be avoided to avoid the faults. There are three ways of solving the problem. First, by trial and error, second, by pure reasoning and third, by cheating and looking at the program logic. The three faults are:-

THE WIDGET WUMBLES
THE STUGS STICK, and
THE SPROCKETS FALL OFF.

These faults occur intermittently, apparently as a result of actions that you have taken. Each action that can be performed upon the machine has been given a number from 1 to 6. You can order the computer to perform a combination of actions by entering the numbers thus: 135 means that action 1, then 3 and then 5 would be carried out. The actions are:-

1. TAKE OFF THE ENGINE LID.
2. PULL BACK LEVER MARKED 'FORWARD'.
3. PRESS BUTTON B.
4. REMOVE SAFETY CATCH.

5. TAP PLATE C (MARKED 'DO NOT TOUCH').
6. PULL OUT THE BIG RED KNOB.

When you think you know which actions should be avoided in order that the three faults do not occur, then type 999 and give your answer.

- 0000000000 -

***** CHEQUE-ACCOUNT MANAGER LII/16K - by David R. Powell *****

Written on a late model System 80, this program contains a machine code tape handler called 'ARRAY SAVER', written by Mr. Ken Shillito and published in MICRO-80 Issue 22. Note that this routine has been modified to reside in high memory. The location suggested by Ken causes havoc with numeric variables. Memory size and machine routines are handled internally.

The program will store up to 50 transactions of the user's cheque account. It maintains a balance of the account. Being fully interactive, it will check bank statements, edit records, sort records and run record searches.

INITIATION AND USE

The program should be copied onto the start of a cassette. A data file is then generated by the direct command - RUN 1270. (Note that cassette 'index counter' settings have been derived using a C60 audio cassette). The file is then saved onto cassette, and a duplicate back-up also.

After CLOADing and RUNning, the program directs the user to load the data file. Should it encounter the data initiated by itself, it will instruct the user how to proceed.

DATE

Dates must be entered in the format shown by the computer, i.e. DD/MM/YY. Leading zeros should be included, and the computer will reject improper entries. The reason for this strict regimen is that the date forms the major index to a record, and all records are sorted primarily by date.

MAIN MENU

The five elements of the menu are described in detail below.

1. Enter New Cheques

The process by which the user enters account transactions into the records. These include cheques written, deposits and bank fees. The user should enter records following the format shown by the computer, or enter a 'Q' to return to the menu.

Of the format: The date should follow the rule described above. Spaces shown in the format (2 between each field) should be maintained. Under 'CHEQUE' the user should insert a cheque number or the abbreviation DEP for deposit or FEE to cover bank charges or other deductions from the account. The PARTICULARS field may occupy any of the area under the hyphens and title. The value of the transaction should be placed under the dollar and cents characters. The decimal place should be immediately below the example, and trailing zeros should be included. If a normal domestic cheque user needs to write a cheque of value greater than $9999.99 (?!), then it may be entered twice with the value divided into smaller portions. A value of zero is illegal since this is used to check against accidental alpha characters in the value field. No commas should be used anywhere in the record. Automatic verification of the record does take place and most errors are detected, but if one sneaks in, it may be edited out later.

After each record is inserted the computer calculates an adjustment to the user's book balance, + or -.

2. Balance Account

This is one of the major benefits of the program since it automatically balances the book against an incoming bank statement. (Note that any fees on the statement should first be entered into the records).

The computer will display the previous statement's balance and the first item on record. Three special commands are identified: (i) Q - quit and return to menu
(ii) B - bypass item
(iii) C - cancel out

If the user chooses to quit and no balancing has yet been done, the menu is returned. Otherwise steps are taken to verify the stage reached.

The user should bypass an item if it does not appear on the statement. The display will echo a 'B' and show that no change has been recorded in the balance.

To cancel out an item means that the user has matched it against the statement. The balance is amended.

Progress is made through the records until either the user has arrived at the point where his own and the bank records agree, or the last record is reached. In the case of the former, the user should ask to 'Q'uit. If the latter, the computer will ask to 'Q'uit!

If the new balance arrived at is acceptable, the records will be amended and restored, otherwise they will be restored to their original composition.

3. Examine Records

A sub-menu of seven choices is given. This provides a means of searching the records in various ways. Only the first two need extra description here. These will search a range of records specified by the user. If only one date or cheque number is sought after, then enter the same value twice in response to the queries of the computer.

4. Edit Records

The means by which any of the records may be altered. Automatic adjustment is made to other records if the editing dictates that this is required, e.g. if a cheque previously had a given value of $20.00 and this is edited to be $25.00, the book balance will be reduced by five dollars. All of the command keys are shown on the screen.

5. Quit

Checks are made on alterations to the data file. If none were made, the program is terminated, but if there were changes made the user is instructed to store them onto cassette. Because the data file over-writes the same tape sections, the user is given the opportunity to make an 'archival' copy onto another tape.

NOTES

Most operating instructions are included in the program. Although supplied on the Distribution DOS, this program will not work in DISK BASIC. To use it, you will have to save it onto cassette, then enter Level II BASIC and reload it from cassette. A more sensible solution would be to remove the Array saver functions which prevent its disk system use and replace them with functions to save your data to disk.

Printer owners could need a printout function somewhere along the line.

PROGRAM OUTLINE

10	Set MEMSIZE
100	Get and confirm 'NEW LINE' (SUBROUTINE)
110	Get and confirm Y/N answer (SUBROUTINE)
120	Format data string (SUBROUTINE)
130	Dissect data string (SUBROUTINE)
140	Page size monitor for Editor (SUBROUTINE)
160	Get any key input (SUBROUTINE)
170	Error message for Editor (SUBROUTINE)
180	CLEAR space, title page
190	DIMension records
200	Get date. Control data file input
220	Check data file
260	MAIN MENU
290	ENTER NEW CHEQUES, test record size
300	Record full
330	Input records
430	BALANCE ACCOUNT
540	EXAMINE RECORDS
800	QUIT, test save flag
820	Save data file (SUBROUTINE)
870	EDIT RECORDS
1080	Bubble sort by date then cheque number
1160	ARRAY SAVER by K. Shillito
1250	Back-up data file load
1260	Data input for new user
1270	Data file initiator utility

- 0000000000 -

***** MORSE PRACTICE LII/16K m.l. - by K.E. Hicks *****

MORSE PRACTICE is a machine language program designed to teach the Morse code, and to develop speed in copying the code.

The program consists essentially of three parts: an input segment, an output segment, and a text buffer, which will hold up to 2500 characters, which may be alphabetic, numeric, punctuation marks, or spaces between words.

With the input segment, the text buffer may be filled with text or characters (including numerals, punctuation and spaces) from the keyboard, or the buffer may be filled with groups of computer-selected characters. The output segment is used to present the contents of the text buffer as Morse code at any desired speed from less than one word per minute to over 400 wpm.

The output appears as a signal on the external cassette line which is normally plugged into the input of a tape recorder. The program is compatible with TRS-80, SYSTEM-80/VIDEO GENIE, etc. To hear the Morse, simply connect the computer to an amplifier in the same manner as usually used for programs with a sound output.

LOAD the program as a SYSTEM tape. It is not necessary to protect memory, so on power up, respond to MEMORY SIZE?/READY? with ENTER/NEWLINE. Then type SYSTEM and ENTER/NEWLINE and reply to *? with MORSE and ENTER/NEWLINE. When loading is complete, which takes about 15 seconds, respond to *? with / ENTER/NEWLINE. The version supplied on the Distribution DOS has an ORG of 8ØØØH. To load this program on a disk system, type LOAD MORSE/CMD then go into BASIC and type SYSTEM and answer the prompt *? with /32768 and press ENTER/NEWLINE.

The message INPUT/OUTPUT/FINISH? (I/O/F) will be presented at the top of the screen, and the computer will be in what may be termed the waiting state.

If you type "I", the screen will be cleared and you may proceed to enter material into the text buffer by simply typing via the keyboard. When the buffer is full, the message BUFFER FULL is given, and on pressing ENTER/NEWLINE you will be returned to the waiting state. It is not necessary to fill the buffer completely. If you type "*" at any time, further input will be aborted and the computer will revert to the waiting state.

If you type "F" while in the waiting state, you will be returned to BASIC. If, however, you change your mind, or have typed "F" in error, the program may be recovered, with the text buffer intact, by typing SYSTEM ENTER/NEWLINE and responding to *? with /17300 ENTER/NEWLINE. This recovery process may be used if the program crashes for any reason.

If you type "O" while in the waiting state, the screen will be cleared and you will be presented with a series of messages requiring responses.

The first is SPEED? (WPM). Type in the speed you require in words per minute. This must be a positive number, but need not be an integer. Terminate with ENTER/NEWLINE. Mistyping here may cause a system crash, usually with the message ?TM ERROR or ?/0 ERROR. The program may be recovered as described above.

The next message is NORMAL SPACING? (Y/N). If you type "Y" you will receive normal Morse at the speed you have just specified. If you type "N", then the individual characters of each word or group will be spaced a little more than normal (actually, by the duration of one "dash"). This makes Morse easier to copy. It will slow down the overall rate of sending by about 25%, without changing the actual speed at which each character is sent.

The third message is TEXT/RANDOM GROUPS? (T/G). If you press "T", then the screen will be cleared, and on pressing any key, you will start to receive the contents of the text buffer in Morse code as a tone from the speaker of your amplifier, at the speed and spacing you have specified.

If, however, you have typed "G", it indicates to the computer that you want the buffer to be filled with groups of characters selected at random. You will be presented with further questions requiring answers.

The first such question is NUMBERS? (Y/N). Press "Y" if you want numerals to be included in the character groups. The second question (which will only appear if you have answered "Y" to NUMBERS? (Y/N), is PUNCTUATION? (Y/N), which asks if you want punctuation as well as numerals and alphabetic characters in your groups.

Having instructed what the groups are to contain, you will then be asked GROUP SIZE?, which must be answered by a positive integer from 0 to 255, followed by ENTER/NEWLINE. If you have selected zero, then the group size will vary and will consist of groups of from 2 to 10 characters, the size of each group being determined by the random function. Any other number from 1 to 255 will result in the buffer being filled with groups of the specified number of randomly selected characters, each group being separated by a word space.

After answering this last question, the computer will appear to hang up while the buffer is being filled. This takes about 15 seconds, after which the screen will be cleared. Then, on pressing any key you will receive the contents of the buffer in Morse as described above. This will continue until you have run through the whole contents of the buffer, or have come to the abort character (*), when the computer will return to the waiting state. You may interrupt sending at any time by holding down the space bar until a space between words is reached. Sending will then cease. Pressing any key other than BREAK will cause sending to resume where it was interrupted. Pressing BREAK will return you to the waiting stage, when you may then enter new text via the keyboard, or replay the buffer contents, perhaps at a different speed or different character spacing.

As each character is sent in code, it is printed on the screen. This allows you to cheat, of course, but it also allows you to check the accuracy of your copying. You have been copying at say, 15 wpm. To check yourself, replay at say 100 wpm, with the amplifier switched off, and check what you have copied by comparing your effort with the text appearing on the screen.

Note that if you have included carriage return characters in your text, these will be reproduced on the screen, but will be ignored by the Morse output. This applies also to any illegal characters (%,$,£,&, etc).

Note that the hyphen (-) is represented normally on the screen, but the doubledash (--) is represented on the screen by "=". There are also two special characters used commonly in Morse communication which have been included, and which you may use when inputting from the keyboard. The first is the 'start of transmission' which is heard as KA run together (-.-.-) and which is represented on the keyboard (and screen) by '+'; the second is the 'end of transmission' symbol. This is heard as AR run together (.-.-.), and is represented by the character '@'. These two characters are not used by the computer when selecting random characters for groups, but the 'start of transmission' character is always present as the first character (followed by a space) in the text buffer. This is built into the program.

Well, that is about all there is to be said about the operation of the program. The following is a few suggestions on how to get the most out of it if you are seriously trying to learn Morse, or trying to get your speed up.

First, if you are learning the code, it is best to learn only a small group of characters at a time. For example, start with the first six letters of the alphabet, A to F. Learn these, and then input these letters, randomly, separating each by one or more spaces. Then have the computer send at about 5-6 wpm and try to copy the characters. Keep at it until you have mastered these six characters. Then do the same for the next 6 characters, G-L. Master these six in the same way, and then combine the two, and have the computer send the 12 characters A-L, and keep at it until you can copy with 100% accuracy. Then add the next six letters, and so on until you have mastered the whole alphabet.

It is essential that you learn to recognise the character by sound, and not as so many dots and dashes. "A" is d'dah; not dot-dash. "B" is dah di'dit, not dash-dot-dot, and so on. For this reason it is a mistake to have Morse sent at too slow a speed. Although, with this program, you may have it sent at 1 wpm (or even 0.01 wpm), you will never be able to recognise the characters at such a slow speed. Five wpm is about the minimum. If you were learning to be a commercial operator, you would learn the characters at a character speed of 25 wpm. But if you are studying for your ham licence, it is necessary to be able to copy at the speeds you will meet at the examination; that is, 5 wpm for the Novice licence, and 10 wpm for the full licence. It is essential that you feel comfortable with these speeds.

When you can recognise and write down all 26 alphabetic characters, learn the ten numerals, which are very easy. But once again, learn the numeric characters as patterns of sound.

The time has now come to have the computer send you groups of random characters. Set the group size to 3 or 4, and use the increased spacing option, and a speed of 5 wpm. The next stage is to change the spacing to normal. Then gradually increase the group size, and finally try with random group sizes.

When you can copy perfectly at 5 wpm normal spacing, it is time to start going for more speed. It is best to use a speed just slightly faster than you can copy with 100% accuracy, so that you are missing characters from time to time. When you do miss a character, you must learn to ignore it and go straight on to the next character. To stop and think about the one you have missed will most certainly make you miss a few more.

A most helpful habit to adopt right from the start is to learn to write down one or two characters "behind" what you are hearing. This is difficult, but something you should consciously strive for. Commercial operators habitually copy one or two whole words behind what they are actually hearing. One way to develop this ability is to start with very small groups (say 3), and not write the characters until the end of the whole group. Then as you increase the group size, you will find that you are still writing down the characters of one group while hearing the characters of the next.. "Hasten slowly".

Morse comes easily to some people, and only with great difficulty to others. There exists a small percentage of people who simply don't have the type of mind to ever achieve more than 5 wpm. But everybody can get to this speed if they really want to.

This program has been written with EDTASM PLUS. You will need this version of the EDITOR/ASSEMBLER to assemble this program.

If you do not have EDTASM PLUS then you will either have to enter the HEX code using a suitable monitor or change the EDTASM PLUS enhancements to suit your Editor/Assembler.

The EDTASM PLUS file on the distribution cassette and disk WILL load into the ordinary EDTASM but the enhancements of EDTASM PLUS will not be recognised and will be treated as errors when you try to assemble the program with the standard EDTASM, viz. the MACRO functions.

NOTE: As the source listing is quite long, only a HEX dump is printed here, although the cassette and disk editions include the source file. The parameters for the object files are:

START	END	ENTRY	
4394H	47A1H	4394H	4K-16K
8ØØØH	84ØDH	8ØØØH	DISK

- 0000000000 -

```
4394:  AF 32 5E 46 CD C9 01 21 62 46 CD A7 28 CD 49 00
43A4:  CD 33 00 FE 49 28 0E FE 4F 28 6C FE 46 20 EE AF
43B4:  D3 FE C3 CC 06 3E 0E CD 33 00 21 00 00 22 60 46
43C4:  CD C9 01 21 A2 47 CD 49 00 F5 CD 0A 44 F1 CD 33
43D4:  00 FE 08 20 03 2B 18 EE D9 ED 5B 60 46 21 C4 09
43E4:  ED 52 28 0E 13 ED 53 60 46 D9 77 FE 2A 28 0E 23
43F4:  18 D4 D9 2B 3E 2A 77 21 E4 46 CD A7 28 CD 49 00
4404:  FE 0D 20 F9 18 8A 01 14 00 C5 01 3F 00 CD 60 00
4414:  C1 0B 78 B1 C8 18 F2 CD C9 01 CD D3 01 3E 0E CD
4424:  33 00 21 A2 47 22 5A 46 01 1A 00 ED 43 56 46 21
4434:  BF 46 CD A7 28 CD 61 03 23 CD 65 0E CD D9 25 CD
4444:  B2 0A 01 7A 8A 11 00 00 CD A2 08 CD 80 0A 22 58
4454:  46 CD 37 46 21 CD 46 CD A7 28 CD 49 00 CD 33 00
4464:  FE 4E 20 05 3E 01 32 5E 46 CD 37 46 21 80 46 CD
4474:  A7 28 CD 49 00 CD 33 00 FE 54 CA 39 45 FE 47 20
4484:  EB CD 37 46 21 9B 46 CD A7 28 CD 49 00 CD 33 00
4494:  FE 4E 28 21 CD 37 46 01 24 00 ED 43 56 46 21 AB
44A4:  46 CD A7 28 CD 49 00 CD 33 00 FE 4E 28 07 01 31
44B4:  00 ED 43 56 46 CD 37 46 01 00 00 ED 43 60 46 21
44C4:  F5 46 CD A7 28 CD 61 03 23 CD 65 0E CD D9 25 CD
44D4:  80 0A 3A 21 41 32 5C 46 32 55 46 B7 20 0B 21 09
44E4:  00 CD 3D 46 7D 3C 32 5C 46 3A 5C 46 47 C5 2A 56
44F4:  46 CD 3D 46 7D 32 02 45 DD 21 02 47 DD 7E 00 CD
4504:  22 45 28 0F C1 10 E6 3E 20 CD 22 45 28 05 3A 55
4514:  46 18 C8 2A 5A 46 2B 3E 2A 77 C1 C3 39 45 2A 5A
4524:  46 77 23 22 5A 46 ED 5B 60 46 13 21 C4 09 ED 52
4534:  ED 53 60 46 C9 3E 10 D3 FE 3E 04 D3 FF CD C9 01
4544:  3E 0E CD 33 00 CD 49 00 21 9F 47 7E CD 33 00 FE
4554:  0D 20 03 23 18 F5 FE 2A CA E4 45 23 E5 21 36 47
4564:  01 35 00 ED B9 E2 9B 45 DD 21 36 47 FD 21 6A 47
4574:  DD 09 FD 09 DD 7E 00 FD 46 00 F5 78 A0 28 1B F1
4584:  17 F5 C5 DC 12 46 D4 EE 45 C1 10 F3 CD 2D 46 3A
4594:  5E 46 B7 C4 2D 46 F1 E1 18 B1 CD 2D 46 CD 2D 46
45A4:  CD 2D 46 3A 40 38 E6 80 28 EC DD 21 00 38 DD 7E
45B4:  01 B7 C2 9A 45 DD 7E 02 B7 C2 9A 45 DD 7E 04 B7
45C4:  C2 9A 45 DD 7E 08 B7 C2 9A 45 DD 7E 10 B7 C2 9A
45D4:  45 DD 7E 20 B7 C2 9A 45 DD 7E 40 E6 04 28 CB F1
45E4:  AF D3 FE C3 94 43 F1 C3 E4 45 ED 4B 58 46 00 C5
45F4:  3E 06 D3 FF 01 20 00 CD 60 00 3E 04 D3 FF 01 20
4604:  00 CD 60 00 C1 0B 78 B1 20 E5 CD 26 46 C9 ED 4B
4614:  58 46 DD 2A 58 46 DD 09 DD 09 DD E5 C1 CD F3 45
4624:  37 C9 ED 4B 58 46 C3 0D 44 CD 26 46 CD 26 46 CD
4634:  26 46 C9 3E 0D CD 33 00 C9 CD 9A 0A 3E 02 32 0F
4644:  4A CD D9 25 CD 80 0A CD CC 14 CD D9 25 CD 80 0A
4654:  C9 00 00 00 00 00 00 00 00 00 00 00 00 00 49 4E
4664:  50 55 54 2F 4F 55 54 50 55 54 2F 46 49 4E 49 53
4674:  48 3F 20 3C 49 2F 4F 2F 46 3E 20 22 54 45 58 54
4684:  2F 52 41 4E 44 4F 4D 20 47 52 4F 55 50 53 3F 20
4694:  3C 54 2F 47 3E 20 22 4E 55 4D 42 45 52 53 3F 20
46A4:  3C 59 2F 4E 3E 20 22 50 55 4E 43 54 55 41 54 49
46B4:  4F 4E 3F 20 3C 59 2F 4E 3E 20 22 53 50 45 45 44
46C4:  3F 20 3C 57 50 4D 3E 20 22 4E 4F 52 4D 41 4C 20
46D4:  53 50 41 43 49 4E 47 3F 20 3C 59 2F 4E 3E 20 22
46E4:  54 45 58 54 20 42 55 46 46 45 52 20 46 55 4C 4C
46F4:  22 47 52 4F 55 50 20 53 49 5A 45 3F 20 22 20 41
4704:  42 43 44 45 46 47 48 49 4A 4B 4C 4D 4E 4F 50 51
4714:  52 53 54 55 56 57 58 59 5A 31 32 33 34 35 36 37
```

```
4724: 38 39 30 2E 2C 28 29 3F
472C: 3D 2D 22 3A 3B 21 27 2F
4734: 2B 40 20 40 80 A0 80 00
473C: 20 C0 00 00 70 A0 40 C0
4744: 80 E0 60 D0 40 00 80 20
474C: 10 60 90 B0 C0 78 38 18
4754: 08 00 80 C0 E0 F0 F8 54
475C: CC B0 B4 30 88 84 48 E0
4764: A8 E0 78 90 A8 50 00 02
476C: 04 04 03 01 04 03 04 02
4774: 04 03 04 02 02 03 04 04
477C: 03 03 01 03 04 03 04 04
4784: 04 05 05 05 05 05 05 05
478C: 05 05 05 06 06 05 06 06
4794: 05 06 06 06 06 04 06 05
479C: 05 05 00 20 2B 20 00 00
```

**** CHECK BOOK DATAFILE ****

COLOUR COMPUTER

```
10 DIM TA(100),B$(100):CC=1'THES
E SET THE NO. OF ENTRIES. -CAN B
E INCREASED DEPENDING ON MEMORY
SIZE
20 CLS:PRINT@38," CHEQUE BOOK DA
TA"
30 PRINT@134,"1==> ENTER CHEQUES
"
40 PRINT@198,"2==> SAVE NEW FILE
"
50 PRINT@262,"3==> LOAD OLD FILE
"
60 PRINT@326,"4==> VIEW ENTRIES"
70 PRINT@390,"5==> EXIT PROGRAM"
80 PRINT@451,"SELECT REQUIRED FU
NCTION";
90 IN$=INKEY$:IF IN$="" THEN 90
ELSE IN=VAL(IN$):IF IN<1 OR IN>5
 THEN 90
100 ON IN GOTO 110,250,390,490,2
30
110 CLS:INPUT"CURRENT BALANCE";B
O:B=BO
120 INPUT"CHEQUE DETAIL,CREDIT (
X,X EXITS)";B$,D$
130 IF B$="X" THEN 190
140 IF LEN(B$)>16 THEN B$=LEFT$(
B$,16)
150 TA=VAL(D$):B=B+TA
160 IF B<1 AND B>-1 THEN B=INT((
B*100)+.5)/100
170 B$(CC)=B$:TA(CC)=TA
180 CC=CC+1:IF CC>100 THEN 200 E
LSE 120
190 PRINT"YOUR BALANCE IS $";B:G
OTO600
200 PRINT:PRINT"TRANSACTION LIMI
T REACHED.        PRESS <ENTER> T
O CONTINUE..."
210 IF INKEY$<>CHR$(13) THEN 210
220 GOTO20
230 CLS:PRINT"YOUR CURRENT BALAN
CE IS $";B
240 END
250 CLS:PRINT@160,"SET TAPE TO E
ND OF LAST CHEQUE
260 PRINT"ENTERED,& PRESS PLAY &
 RECORD"
270 PRINT"BUTTONS.  THE COMPUTER
 WILL NOW"
280 PRINT"RECORD YOUR LATEST FIG
URES."
290 PRINT:INPUT"PRESS ENTER WHEN
 RECORDER READY";A$
300 PRINT:PRINT"NOW RECORDING":P
RINT
310 'PRINTS DATA TO TAPE
320 OPEN"O",-1,"CAS0"
330 PRINT#-1,BO;CC
340 FOR T=1 TO CC
350 PRINT#-1,B$(T),TA(T)
360 NEXT T
370 CLOSE
380 GOTO 20
390 CLS:PRINT@192,"REWIND TAPE T
O START OF DATA    FILE & PRESS
PLAY BUTTON."
400 PRINT"THE COMPUTER WILL NOW
RETRIEVE  ALL CHEQUE BOOK DATA."
410 INPUT"PRESS ENTER WHEN READY
";A$
420 CLS:PRINT@40,"SEARCHING...":
OPEN"I",-1,"CAS0"
430 PRINT@106,"RETRIEVING DATA"
440 INPUT#-1,BO,CC
450 FOR T=1 TO CC
460 INPUT#-1,B$(T),TA(T)
470 NEXT T
480 CLOSE
490 'DISPLAY TAPE LOAD
500 B=BO:S=1:IF CC>14 THEN V=14
ELSE V=CC
510 CLS:PRINT"  OPENING BALANCE:
 $";BO
520 PRINT"  DETAILS           CRED
IT BALANCE";
530 FOR I=S TO V
540 B$=B$(I):TA=TA(I):B=B+TA:IF
B<1 AND B>-1 THEN B=INT((B*100)+
.5)/100
550 PRINTUSING"%                %
####.## ####.##";B$(I);TA(I);B;
560 NEXT I
570 IF V=CC THEN 600
580 INPUT"PRESS ENTER TO CONTINU
E...";A$
590 S=S+14:V=V+14:IF V>CC THEN V
=CC:GOTO 510
600 INPUT"HIT ENTER TO RETURN TO
 MENU..";A$
610 GOTO20
```

**** PAYROLL ****

COLOUR COMPUTER

```
10 'PAYROLL  COPYRIGHT L J LAWES
 21 RODNEY ST LINDUM 3962998
20 CLS:CLEAR1500
30 DIMA(20,11):' WILL HANDLE UP
TO 20 EMPLOYEE RECORDS
40 IS=12345:GOTO1070:'INITIALISE
 SYSTEM
50 IS=0
60 'MENU - TO RECOVER PROGRAM TY
PE <GOTO 100>
70 PRINT" PAYROLL CASSETTE BASED
 SYSTEM"
80 PRINT:PRINT@68,"1==> UPDATE M
ASTER FILE"
90 PRINT@132,"2==> LOAD OLD FILE
"
100 PRINT@196,"3==> PROCESS PAYR
OLL"
110 PRINT@260,"4==> SAVE NEW FIL
E"
120 PRINT@324,"5==> EMPLOYEE INQ
UIRY"
130 PRINT@419,"SELECT REQUIRED F
UNCTION"
140 IN$= INKEY$:IF IN$=""THEN140
 ELSEIN= VAL(IN$)
```

```
150 ON IN GOTO160,320,420,1530,1
630
160 CLS:PRINT"TO ADD, DELETE, CH
ANGE EMPLOYEE DETAILS,USE DATA S
TATEMENTS AND EDIT IN THE NORMAL
 MANNER AS    REQUIRED."
170 PRINT"USE LINES 200-290 FOR
MASTER    FILE DATA"
180 INPUT"PRESS <ENTER> TO CONTI
NUE ";IN$:CLS:LIST190-290
190 'DATA LINES BEGIN HERE
200 DATA 1,A APPLE,6.25,0,1
210 DATA 2,B BRAVO,5.50,0,2
220 DATA 3,C CHARLIE,4.75,0,3
230 DATA 4,D DELTA,6.50,0,6
240 DATA 5,E ECHO,0,250.00,1
250 DATA 6,F FOXTROT,4.50,20.00,
2
260 DATA 7,G GOLF,6.25,20.00,3
270 DATA 8,H HOTEL,8.25,0,4
280 DATA 9,I INDIA,4.50,0,6
290 DATA10,J JULIET,6.25,0,1
300 DATA 99," ",0,0,0
310 'DATA LINES END HERE
320 'LOAD OLD FILE FROM CASSETTE
330 CLS
340 INPUT"         PREPARE CASSET
TE        PRESS <ENTER> WHEN REA
DY TO LOAD ";IN$:PRINT"DATA NOW
LOADING..."
350 OPEN "I",-1,"PAY"
360 INPUT#-1,RC:'NO. OF EMPLOYEE
 RECORDS ON TAPE
370 FOR I=1 TO RC
380 INPUT#-1,A(I,1),A(I,2),A(I,3
),A(I,4),A(I,5),A(I,6),A(I,7),A(
I,8),A(I,9),A(I,10),A(I,11)
390 NEXTI
400 CLOSE-1
410 PRINT"DATA LOADING COMPLETE"
:FORT=1TO1000:NEXTT:CLS:GOTO70
420 'INPUT PAYROLL DATA
430 CLS:INPUT"PAYROLL FOR WEEK E
NDING  ";DA$
440 INPUT"IS PRINTER REQUIRED (Y
/N) ";LP$
450 IF LP$< >"Y"AND  LP$< >"N" T
HEN440
460 IF LP$="Y" THEN LP=1 ELSE LP
=0:'(TRUE=1, FALSE=0 IN 'IF' TES
T)
470 IF LP THEN IF PEEK(&HFF22) A
ND 1 THEN PRINT"PRINTER NOT READ
Y":GOTO440
480 CLS
490 H1=0:H2=0:H3=0:HR=0:RT=0:GA=
0:GB=0:GC=0:GD=0:TX=0:TY=0:TZ=0:
DA=0:DB=0:DC=0:NP=0
500 INPUT"PAY NUMBER ";PN
510 RESTORE:I=1
520 READ MF(I),MF$(I),MR(I),MS(I
),MT(I)
530 IF MF(I)=PN THEN PRINT"EMPLO
YEE NAME  :";MF$(I):RESTORE:GOTO
560
540 IF MF(I)=99 THEN PRINT"INVAL
ID PAY NUMBER":RESTORE:GOTO500
550 I=I+1:GOTO520
560 PRINT"SELECT:-     1==> HOURL
Y RATE
570 PRINT" 2==> WEEKLY RATE  3==
> BOTH
580 IN$= INKEY$:IF IN$=""THEN580
 ELSEIN= VAL(IN$)
590 ON IN GOTO600,680,600
600 INPUT"HRS-ORD,TIME&HALF,DOUB
LE ";H1,H2,H3
610 HR=H1+(1.5*H2)+(2*H3):PRINT"
EQUIV ORD TIME  :";HR
620 PRINT"SELECT 1==>AUTO: 2==>
MANUAL RATE INPUT :"
630 JN$= INKEY$:IF JN$="" THEN63
0 ELSEJN= VAL(JN$)
640 ON JN GOTO650,660
650 RT=MR(I):PRINT"HOURLY RATE
:";RT:GOTO670
660 INPUT"HOURLY RATE  :";RT
670 IF IN=3 THEN680 ELSE730
680 PRINT"SELECT 1==> AUTO: 2==>
 MANUAL EARNINGS INPUT"
690 JN$= INKEY$:IF JN$=""THEN690
 ELSEJN= VAL(JN$)
700 ON JN GOTO710,720
710 GB=MS(I):PRINT"WEEKLY EARNIN
GS  :";GB:GOTO730
720 INPUT"WEEKLY EARNINGS  :";GB
730 GA=HR*RT:GC=GA+GB:GD= INT(GC
)+1:PRINT"HOURLY EARNINGS  :";GA
740 PRINT"GROSS EARNINGS  :";GC
750 'TAX CALCULATIONS
760 PRINT"TAX CALCULATIONS"
770 PRINT"1==> NO REBATE    3==>
 $800 REBATE    5==> AUTO CALC."
780 PRINT"2==> $559 REBATE  4==>
 MANUAL OVERIDE 6==> NO EXEM/REB
"
790 IM$= INKEY$:IF IM$=""THEN790
 ELSEIM= VAL(IM$)
800 ON IM GOTO810,810,810,920,87
0,890
810 IF GD<75 THENTX=0:GOTO930
820 IF GD<326 THENTX=GD*.325-24.
35:GOTO840
830 IF GD<657 THENTX=GD*.46-68.4
2 ELSETX=GD*.6-160.53
840 IF IM=1THEN930
850 IF IM=2THENTX=TX-10.621:GOTO
930
860 IF IM=3THENTX=TX-15.2:GOTO93
0
870 IM=MT(I):IF IM=1 OR  IM=2 OR
  IM=3 THEN800
880 IF IM=4THEN920 ELSE IFIM=6 T
HEN890
890 IF GD<202 THENTX=GD*.32-.32:
GOTO930
900 IF GD<533 THENTX=GD*.46-28.6
 ELSETX=GD*.6-103.29
910 GOTO930
920 INPUT"MANUAL TAX CALCULATION
  :";TX
930 :REM 'ROUNDING OFF TO NEARES
T 5 CENTS
940 TX=TX*10:TY= INT(TX):TZ=TX-T
Y
950 IF TZ<.25THENTZ=0:GOTO970
960 IF TZ<.75THENTZ=.5 ELSETZ=1
970 TX=(TY+TZ)/10
980 PRINT"GROUP TAX DEDUCTED  :"
;TX
990 INPUT"DEDUCTION A  :";DA
1000 INPUT"DEDUCTION B  :";DB
1010 INPUT"DEDUCTION C  :";DC
1020 NP=GC-TX-DA-DB-DC
1030 PRINT"NETT TAKE HOME PAY  :
";NP
1040 PRINT"SELECT 1==> ACCEPT: 2
==> REJECT"
1050 IP$= INKEY$:IF IP$=""THEN10
50 ELSEIP= VAL(IP$)
1060 ON IP GOTO1070,1200
1070 F1$="###             %
             %              %
   %"
1080 F2$="##.##  ##.##  ##.##  #
#.##  ##.####  ####.## ####.## #
###.##"
1090 SA$="###.#  ###.#  ###.#
 ###.#"
1100 SB$="##.#### ####.## ####.#
# ####.##"
1110 SC$="####.##  ####.##  ####
.##"
1120 SD$="####.##       ####.##"
1130 F3$="#####.##  ######.##
####.##   ####.##   ####.##   ##
###.##"
```

```
1140 F4$="#####  #####  #####  #
####                          ##
###.##"
1150 H1$="P/NO                EMPLOY
EE NAME                       DATE
"
1160 H2$="ORD.H  1.5HR  2.0HR  E
Q/ORD  RATE     HRLY     WEEKLY
 GROSS"
1170 H3$="GROSS        GRP/TAX
  DED/A      DED/B      DED/C
NET PAY"
1180 IF IS=12345 THEN50
1190 GOTO1210
1200 CLS:GOTO480
1210 'UPDATE FILE TOTALS
1220 I=PN:A(I,2)=A(I,2)+H1:A(I,3
)=A(I,3)+H2:A(I,4)=A(I,4)+H3:A(I
,5)=A(I,5)+HR:A(I,6)=A(I,6)+GC:A
(I,7)=A(I,7)+TX:A(I,8)=A(I,8)+DA
:A(I,9)=A(I,9)+DB:A(I,10)=A(I,10
)+DC:A(I,11)=A(I,11)+NP:DV=0
1230 'SCREEN/PRINTER OUTPUT
1240 CLS
1250 PRINTUSING"%   %###%       %
 ";"P/NO:";PN;"   DATE:";DA$
1260 PRINT"NAME:  ";MF$(I):PRINT
1270 PRINT"ORD.H  1.5HR  2.0HR
 EQ/ORD":PRINTUSING SA$;H1;H2;H3
;HR:PRINT
1280 PRINT"  RATE   HOURLY  WEEK
LY   GROSS":PRINTUSING SB$;RT;GA
;GB;GC:PRINT
1290 PRINT"  DED/A    DED/B    D
ED/C":PRINTUSING SC$;DA;DB;DC:PR
INT
1300 PRINT" GROSS  GROUP TAX  NE
T PAY":PRINTUSING SC$;GC;TX;NP:P
RINT
1310 IF LP THEN PRINT#-2,H1$
1320 IF LP THEN PRINT#-2,USING F
1$;PN,MF$(I),DA$
1330 IF LP THEN PRINT#-2,H2$
1340 IF LP THEN PRINT#-2,USING F
2$;H1,H2,H3,HR,RT,GA,GB,GC
1350 IF LP THEN PRINT#-2,H3$
1360 IF LP THEN PRINT#-2,USING F
3$;GC,TX,DA,DB,DC,NP:PRINT#-2,"
"
1370 A(0,2)=A(0,2)+H1:A(0,3)=A(0
,3)+H2:A(0,4)=A(0,4)+H3:A(0,5)=A
(0,5)+HR:A(0,6)=A(0,6)+GC:A(0,7)
=A(0,7)+TX:A(0,8)=A(0,8)+DA:A(0,
9)=A(0,9)+DB:A(0,10)=A(0,10)+DC:
A(0,11)=A(0,11)+NP :'ACCUMULATE
PAGE TOTALS
1380 IF LP THEN PRINT#-2,STRING$
(63,95):PRINT#-2," "
1390 PRINT"SELECT 1==> NEXT: 2==
>END:";
1400 IN$= INKEY$:IF IN$=""THEN14
00 ELSEIN= VAL(IN$)
1410 ON IN GOTO1420,1430
1420 CLS:GOTO480
1430 CLS
1440  PRINT"PAGE TOTAL FOR  ";DA
$:PRINT:IF LP THEN PRINT#-2,"PAG
E TOTAL FOR  ";DA$
1450 PRINT"ORD.H  1.5HR  2.0HR
 EQ/ORD":PRINTUSING SA$;A(0,2);A
(0,3);A(0,4);A(0,5):PRINT
1460 IF LP THEN PRINT#-2,H2$
1470 PRINT"  DED/A    DED/B    D
ED/C":PRINTUSING SC$;A(0,8);A(0,
9);A(0,10):PRINT
1480 IF LP THEN PRINT#-2,USING F
4$;A(0,2),A(0,3),A(0,4),A(0,5),A
(0,6)
1490 PRINT"  GROSS  GROUP TAX  N
ET PAY":PRINTUSING SC$;A(0,6);A(
0,7);A(0,11):PRINT
1500 IF LP THEN PRINT#-2,H3$
1510 IF LP THEN PRINT#-2,USING F
3$;A(0,6),A(0,7),A(0,8),A(0,9),A
(0,10),A(0,11)
1520 PRINT"END OF RUN":PRINT:INP
UT"PRESS <ENTER> TO CONTINUE ";I
C:CLS:GOTO70
1530 :REM 'SAVE NEW FILE
1540 INPUT"PREPARE CASSETTE. TYP
E NUMBER OF EMPLOYEE RECORDS ON
FILE, THEN PRESS <ENTER> WHEN RE
ADY :";RC:PRINT"DATA NOW BEING W
RITTEN TO TAPE....."
1550 OPEN "O",-1,"PAY"
1560 PRINT#-1,RC
1570 FOR I=1TO RC
1580 PRINT#-1,A(I,1),A(I,2),A(I,
3),A(I,4),A(I,5),A(I,6),A(I,7),A
(I,8),A(I,9),A(I,10),A(I,11)
1590 NEXTI
1600 PRINT".....DATA SAVE COMPLE
TE"
1610 CLOSE-1
1620 FORT=1TO1000:NEXTT:CLS:GOTO
70
1630 'EMPLOYEE ENQUIRY (YEAR TO
DATE TOTALS)
1640 CLS:PRINT:PRINT
1650  INPUT"IS PRINTER REQUIRED
(Y/N)";LP$
1660 IF LP$="Y" THEN LP=1 ELSE L
P=0
1670 IF LP THEN IF PEEK(&HFF22)
AND 1 THEN PRINT"PRINTER NOT REA
DY":GOTO1650
1680 INPUT"ENTER REQUIRED PAY NU
MBER :";PN
1690 RESTORE:I=1
1700 READ MF(I),MF$(I),MR(I),MS(
I),MT(I)
1710 IF MF(I)=PN THEN1720 ELSE I
FMF(I)=99 THEN1680 ELSEI=I+1:GOT
O1700
1720 DA$="Y.T.D."
1730 IF LP THEN PRINT#-2,H1$
1740 PRINTUSING"%   %###%
% ";"P/NO:";PN;"   DATE:";DA$
1750 PRINT"NAME:  ";MF$(I):PRINT
1760 IF LP THEN PRINT#-2,USING F
1$;PN,MF$(I),DA$
1770 PRINT"ORD.H  1.5HR  2.0HR
 EQ/ORD":PRINTUSING SA$;A(I,2);A
(I,3);A(I,4);A(I,5):PRINT
1780 IF LP THEN PRINT#-2,H2$
1790 IF LP THEN PRINT#-2,USING F
4$;A(I,2),A(I,3),A(I,4),A(I,5),A
(I,6)
1800 PRINT"  DED/A    DED/B    D
ED/C":PRINTUSING SC$;A(I,8);A(I,
9);A(I,10):PRINT
1810 PRINT" GROSS  GROUP TAX  NE
T PAY":PRINTUSING SC$;A(I,6);A(I
,7);A(I,11)
1820 PRINT:PRINTH3$:IF LP THEN P
RINT#-2,H3$
1830 IF LP THEN PRINT#-2,USING F
3$;A(I,6),A(I,7),A(I,8),A(I,9),A
(I,10),A(I,11)
1840 PRINT:PN=PN+1:INPUT"NEXT PA
Y NUMBER (99 EXITS)";PN:CLS
1850  IF PN=99 THEN CLS:GOTO70
1860 GOTO1690
```

**** CHECK BOOK DATAFILE ****

HITACHI PEACH

```
10 'MICRO-80 SOFTWARE
20 REM  ORIGINAL PROGRAM BY K.L.FORD
30 REM  PUBLISHED IN ISSUE 20,JULY 1981
40 REM  EXTENSIVELY MODIFIED FOR THE PEA
CH
```

```
50 REM  BY A.G.FIELKE FOR MICRO-80
60 REM
70 DIM TA(38),B$(30):R=0:B=0:M=0:C=0:Y=1
:N=0
80 CLS:LOCATE 17,3:PRINT"CHEQUE BOOK DAT
A"
90 LOCATE17,5:PRINT"1===> ENTER CHEQUES"
100 LOCATE17,6:PRINT"2===> SAVE NEW FILE
"
110 LOCATE17,7:PRINT"3===> LOAD OLD FILE
"
120 LOCATE17,8:PRINT"SELECT REQUIRED FUN
CTION"
130 IN$=INKEY$:IF IN$="" THEN 130 ELSE I
N =VAL (IN$):IF IN<1 OR IN >3 THEN130
140 ON IN GOTO 150,430,610
150 CLS:INPUT"WHAT IS THE CURRENT BALANC
E.";BO:B=BO
160 PRINT:INPUT"CHEQUE NUMBER";C:CC=0
170 CONSOLE3,15
180 GOSUB820:'PRINT HEADINGS
190 GOSUB 930 'PRINT OPENING BALANCE
200 GOSUB330:'GET A CHEQUE
210 B$(CC)=B$:TA(CC)=TA
220 GOSUB 840:'PRINT A LINE
230 IF INSTR(B$,"%FEE")>0 THEN 250
240 IF TA<0 THEN C=C+1
250 LOCATE2,22:INPUT"HAVE YOU ANY MORE C
HEQUES TO ENTER.";A$
260 IFA$="Y" THEN IF CC<30 THEN CC=CC+1:
GOTO200 ELSE 280
270 IF A$<>"N" THEN 250 ELSE 300
280 LOCATE2,22:PRINTCHR$(26);"TRANSACTIO
N LIMIT REACHED!  PRESS <RETURN> TO CONT
INUE ..."
290 A$=INKEY$:IF A$<>CHR$(13) THEN 290
300 CONSOLE0,24:GOTO 80
310 CLS:PRINT"YOUR CURRENT BALANCE IS $"
;B
320 END
330 REM INPUT DATA
340 LOCATE2,18:PRINTCHR$(26);"DATE AND P
ARTICULARS OF LAST CHEQUE (DD/MM/YY DRAW
ER)"
350 LOCATE2,19:INPUT"(MAX.  16 CHARACTER
S)";B$:IF LEN(B$)>16 THEN B$=LEFT$(B$,14
)
360 LOCATE2,19:PRINT:INPUT"DEPOSIT IN DO
LLARS AND CENTS.  <RETURN> IF A WITHDRAW
AL.";D$
370 IF VAL(D$)=0 THEN 380 ELSE TA=VAL(D$
):GOTO 400
380 LOCATE2,21:PRINTCHR$(26);:INPUT"WITH
DRAWAL IN DOLLARS AND CENTS";M$:M=VAL(M$
):IF M=0 THENPRINTCHR$(7);:GOTO380
390 TA=-M
400 B=B+TA
410 IF B<1 AND B>-1 THEN B=INT((B*100)+.
5)/100
420 RETURN
430 PRINT"SET CASSETTE TO END OF LAST CH
EQUE"
440 PRINT
450 PRINT"ENTERED AND PRESS PLAY AND REC
ORD"
460 PRINT
470 PRINT"BUTTONS.  THE COMPUTER WILL NO
W"
480 PRINT
490 PRINT"RECORD YOUR LATEST FIGURES"
500 PRINT
510 INPUT"PRESS RETURN WHEN READY TO REC
ORD";A$
520 PRINT"NOW RECORDING"
530 REM"PRINTS DATA TO TAPE"
540 OPEN "O",1,"CAS0:DATA"
550 PRINT#1,BO,C,CC
560 FOR T= 0 TO CC
570 PRINT#1,B$(T),TA(T)
580 NEXT T
590 CLOSE
600 GOTO80
610 PRINT"REWIND CASSETTE RECORDER TO ST
ART OF"
620 PRINT"DATA FILE AND PRESS PLAY BUTTO
N "
630 PRINT"THE COMPUTER WILL NOW RETRIEVE
"
640 PRINT"ALL CHEQUE BOOK DATA."
650 INPUT"PRESS RETURN WHEN READY TO RET
RIEVE.";A$
660 CLS:PRINT"SEARCHING....":OPEN "I",1,
"CAS0:DATA"
670 PRINT"RETRIEVING DATA"
680 INPUT#1,B,C,CC
690 FOR T=0 TO CC
700 INPUT#1,B$(T),TA(T)
710 NEXT T
720 CLOSE
730 'DISPLAY TAPE LOAD
740 CLS:CONSOLE3,15:GOSUB 820
750 GOSUB 930:'PRINT OPENNING BAL
760 T=CC:FOR CC=0 TO T
770 B$=B$(CC):TA=TA(CC):GOSUB 400:'WORK
OUT BAL
780 GOSUB 840:'PRINT A LINE
790 NEXT CC
800 LOCATE2,23:INPUT"PRESS 'RETURN' TO C
ONTINUE...";A$
810 BO=B:CC=0:GOTO 180
820 CLS:LOCATE0,0:PRINT"NUMBER" TAB(11)
"PARTICULARS" TAB(31) "CREDIT" TAB(41) "
 DEBIT " TAB(57) "BALANCE"
830 RETURN
840 YP=CC+3:IF YP>16 THEN YP=16
850 LOCATE2,YP
860 IF INSTR(B$,"%FEE")>0 THEN 880
870 IF TA<0 THEN PRINTTAB(2)C;
880 PRINTTAB(11)B$;
890 D$="            #####.##     ######.
##"
900 W$="#####.##                 ######.
##"
910 PRINTTAB(28)" ";:IFTA<=0 THEN PRINT
USINGD$;TA,B ELSE PRINT USINGW$;ABS(TA),
B
920 PRINT:RETURN
930 L$="#####.##":PRINTTAB(9)"OPENING BA
LANCE";TAB(54)" ";:PRINTUSINGL$;B
940 RETURN
```

**** PAYROLL ****

HITACHI PEACH

```
5 'V2.0 2/2/83 MICRO-80 SOFTWARE
10 'PAYROLL  COPYRIGHT L J LAWES 21 RODN
EY ST LINDUM 3962998
20 CLS:CLEAR200:ON ERROR GOTO1680
30 DEFINTI-J:DEFSTRF:DEFDBLA,B
40 DIMA(20,11)  'WILL HANDLE UP TO 20 EM
PLOYEE RECORDS
50 IS=12345:GOTO1060 'INITIALISE SYSTEM
60 IS=0
70 'MENU - TO RECOVER PROGRAM TYPE <GOTO
 100>
80 LOCATE7,3:PRINT"PAYROLL CASSETTE BASE
D SYSTEM - L2 16K"
90 PRINT:LOCATE17,5:PRINT"1==> UPDATE MA
STER FILE"
100 LOCATE17,6:PRINT"2==> LOAD OLD FILE"
110 LOCATE17,7:PRINT"3==> PROCESS PAYROL
L"
120 LOCATE17,8:PRINT"4==> SAVE NEW FILE"
130 LOCATE17,9:PRINT"5==> EMPLOYEE ENQUI
RY"
140 LOCATE7,14:PRINT"SELECT REQUIRED FUN
CTION"
150 IN$=INKEY$:IF IN$=""THEN150 ELSEIN=V
AL(IN$):IF IN<1 OR IN>5 THEN 150
160 ON IN GOTO170,330,430,1390,1500
```

```
170 CLS:PRINT"TO ADD, DELETE OR CHANGE E
MPLOYEE DETAILS, USE DATA STATEMENTS":PR
INT" AND EDIT IN THE NORMAL MANNER AS RE
QUIRED"
180 PRINT"USE LINES 210-300 FOR MASTER F
ILE DATA"
190 INPUT"PRESS <ENTER> TO CONTINUE ";IN
$:CLS:LIST210-300
200 'DATA LINES BEGIN HERE
210 DATA 1,A APPLE,6.25,0,1
220 DATA 2,B BRAVO,5.50,0,2
230 DATA 3,C CHARLIE,4.75,0,3
240 DATA 4,D DELTA,6.50,0,6
250 DATA 5,E EVEREST,0,250.00,1
260 DATA 6,F FROGG,4.50,20.00,2
270 DATA 7,G GOLF,6.25,20.00,3
280 DATA 8,H HOTEL,8.25,0,4
290 DATA 9,I INDIGO,4.50,0,6
300 DATA10,J JULIET,6.25,0,1
310 DATA 99," ",0,0,0
320 'DATA LINES END HERE
330 'LOAD OLD FILE FROM CASSETTE
340 INPUT"PREPARE CASSETTE PRESS <ENTER>
 WHEN READY TO LOAD ";IN$
350 OPEN "I",1,"CAS0:DATA"
360 PRINT"DATA NOW LOADING..."
370 INPUT#1,RC%  'NO OF EMPLOYEE RECORDS
 ON TAPE
380 FOR I=1 TO RC%
390 INPUT#1,A(I,1),A(I,2),A(I,3),A(I,4),
A(I,5),A(I,6),A(I,7),A(I,8),A(I,9),A(I,1
0),A(I,11)
400 NEXTI
410 CLOSE
420 PRINT"DATA LOADING COMPLETE":FORT=1T
01000:NEXTT:CLS:GOTO80
430 'INPUT PAYROLL DATA
440 CLS:INPUT"PAYROLL FOR THE WEEK ENDIN
G  ";DA$
450 INPUT"IS THE PRINTER REQUIRED (Y/N)
";LP$
460 IF LP$<>"Y"AND  LP$<>"N" THEN450
470 IF LP$="Y" THEN OPEN"O",1,"LPT0:":PR
INT"SWITCH PRINTER ON"
480 CLS
490 H1=0:H2=0:H3=0:HR=0:RT=0:GA=0:GB=0:G
C=0:GD=0:TX=0:TY=0:TZ=0:DA=0:DB=0:DC=0:N
P=0
500 INPUT"PAY NUMBER ";PN%
510 RESTORE:I=1
520 READ MF%(I),MF$(I),MR(I),MS(I),MT(I)
530 IF MF%(I)=99 THEN PRINT"INVALID PAY
NUMBER":GOTO500
540 IF MF%(I)=PN% THEN PRINT"EMPLOYEE NA
ME  :";MF$(I):RESTORE:GOTO560
550 I=I+1:GOTO520
560 PRINT"SELECT 1==> HRLY RATE: 2==> WK
LY RATE: 3==> BOTH"
570 IN$=INKEY$:IF IN$=""THEN570 ELSEIN=V
AL(IN$)
580 ON IN GOTO590,670,590
590 INPUT"HRS-ORD,TIME&HALF,DOUBLE ";H1,
H2,H3
600 HR=H1+(1.5*H2)+(2*H3):PRINT"EQUIV OR
D TIME  :";HR
610 PRINT"SELECT 1==>AUTO: 2==> MANUAL R
ATE INPUT :"
620 JN$=INKEY$:IF JN$="" THEN620 ELSEJN=
VAL(JN$)
630 ON JN GOTO640,650
640 RT=MR(I):PRINT"HOURLY RATE  :";RT:GO
TO660
650 INPUT"HOURLY RATE  :";RT
660 IF IN=3 THEN670 ELSE720
670 PRINT"SELECT 1==> AUTO: 2==> MANUAL
EARNINGS INPUT"
680 JN$=INKEY$:IF JN$=""THEN680 ELSEJN=V
AL(JN$)
690 ON JN GOTO700,710
700 GB=MS(I):PRINT"WEEKLY EARNINGS  :";G
B:GOTO720
710 INPUT"WEEKLY EARNINGS  :";GB
720 GA=HR*RT:GC=GA+GB:GD=INT(GC)+1:PRINT
"HOURLY EARNINGS  :";GA
730 PRINT"GROSS EARNINGS  :";GC
740 'TAX CALCULATIONS
750 PRINT"TAX CALCULATIONS"
760 PRINT"1==> NO REBATE     3==> $800 RE
BATE    5==> AUTO CALC."
770 PRINT"2==> $559 REBATE  4==> MANUAL
OVERIDE 6==> NO EXEM/REB"
780 IM$=INKEY$:IF IM$=""THEN780 ELSEIM=V
AL(IM$)
790 ON IM GOTO800,800,800,910,860,880
800 IF GD<75 THENTX=0:GOTO920
810 IF GD<326 THENTX=GD*.325-24.35:GOTO8
30
820 IF GD<657 THENTX=GD*.46-68.42 ELSETX
=GD*.6-160.53
830 IF IM=1THEN920
840 IF IM=2THENTX=TX-10.621:GOTO920
850 IF IM=3THENTX=TX-15.2:GOTO920
860 IM=MT(I):IF IM=1 OR  IM=2 OR  IM=3 T
HEN790
870 IF IM=4THEN910 ELSEIFIM=6 THEN880
880 IF GD<202 THENTX=GD*.32-.32:GOTO920
890 IF GD<533 THENTX=GD*.46-28.6 ELSETX=
GD*.6-103.29
900 GOTO920
910 INPUT"MANUAL TAX CALCULATION  :";TX
920 'ROUNDING OFF TO NEAREST 5 CENTS
930 TX=TX*10:TY=INT(TX):TZ=TX-TY
940 IF TZ<.25THENTZ=0:GOTO960
950 IF TZ<.75THENTZ=.5 ELSETZ=1
960 TX=(TY+TZ)/10
970 PRINT"GROUP TAX DEDUCTED   :";TX
980 INPUT"DEDUCTION A   :";DA
990 INPUT"DEDUCTION B   :";DB
1000 INPUT"DEDUCTION C   :";DC
1010 NP=GC-TX-DA-DB-DC
1020 PRINT"NETT TAKE HOME PAY  :";NP
1030 PRINT"SELECT 1==> ACCEPT: 2==> REJE
CT"
1040 IP$=INKEY$:IF IP$=""THEN1040 ELSEIP
=VAL(IP$)
1045 IF (IP-1)*(IP-2)<>0 THEN 1040
1050 ON IP GOTO1060,1150
1060 F1="###              &
    &                  &       &"
1070 F2="##.##  ##.##  ##.##  ##.##  ##.
####  ####.## ####.## ####.##"
1080 F3="#####.##  ######.##   #####.##
 #####.##   #####.##   ######.##"
1090 F4="#####  #####  #####  #####
                         ######.##"
1100 H1$="P/NO             EMPLOYEE NAME
                      DATE"
1110 H2$="ORD.H  1.5HR  2.0HR  EQ/ORD  R
ATE      HRLY      WEEKLY  GROSS"
1120 H3$="GROSS        GRP/TAX     DED/A
    DED/B      DED/C      NET PAY"
1130 IF IS=12345 THEN60 ELSE1140
1140 GOTO1160
1150 CLS:GOTO480
1160 'UPDATE FILE TOTALS
1170 I=PN%:A(I,2)=A(I,2)+H1:A(I,3)=A(I,3
)+H2:A(I,4)=A(I,4)+H3:A(I,5)=A(I,5)+HR:A
(I,6)=A(I,6)+GC:A(I,7)=A(I,7)+TX:A(I,8)=
A(I,8)+DA:A(I,9)=A(I,9)+DB:A(I,10)=A(I,1
0)+DC:A(I,11)=A(I,11)+NP:DV=0
1180 'SCREEN/PRINTER OUTPUT
1190 CLS:PRINT
1200 PRINTH1$:IF LP$="Y" THEN PRINT#1,H1
$
1210 PRINTUSINGF1;PN%,MF$(I),DA$:IF LP$=
"Y" THEN PRINT#1,USINGF1;PN%,MF$(I),DA$
1220 PRINT:PRINTH2$:IF LP$="Y" THENPRINT
#1,H2$
1230 PRINTUSINGF2;H1,H2,H3,HR,RT,GA,GB,G
C:IF LP$="Y" THENPRINT#1,USINGF2;H1,H2,H
3,HR,RT,GA,GB,GC
1240 PRINT:PRINTH3$:IFLP$="Y" THENPRINT#
1,H3$
1250 PRINTUSINGF3;GC,TX,DA,DB,DC,NP:IF L
P$="Y" THENPRINT#1,USINGF3;GC,TX,DA,DB,D
C,NP:PRINT#1" "
```

```
1260 A(0,2)=A(0,2)+H1:A(0,3)=A(0,3)+H2:A
(0,4)=A(0,4)+H3:A(0,5)=A(0,5)+HR:A(0,6)=
A(0,6)+GC:A(0,7)=A(0,7)+TX:A(0,8)=A(0,8)
+DA:A(0,9)=A(0,9)+DB:A(0,10)=A(0,10)+DC:
A(0,11)=A(0,11)+NP 'ACCUMULATE PAGE TOTA
LS
1270 IF LP$="Y"THEN PRINT#1,STRING$(63,9
5):PRINT#1," "
1280 PRINT:PRINT"SELECT 1==> NEXT: 2==>E
ND:"
1290 IN$=INKEY$:IF IN$=""THEN1290 ELSEIN
=VAL(IN$)
1300 ON IN GOTO1310,1320
1310 CLS:GOTO480
1320 CLS:PRINT:PRINT
1330 PRINT"PAGE TOTAL FOR  ";DA$:IF LP$=
"Y"!"PAGE TOTAL FOR  ";DA$
1340 PRINTH2$:IF LP$="Y"!H2$
1350 PRINTUSINGF4;A(0,2),A(0,3),A(0,4),A
(0,5),A(0,6):IF LP$="Y"!USINGF4;A(0,2),A
(0,3),A(0,4),A(0,5),A(0,6)
1360 PRINTH3$:IF LP$="Y" THENPRINT#1,H3$
1370 PRINTUSINGF3;A(0,6),A(0,7),A(0,8),A
(0,9),A(0,10),A(0,11):IF LP$="Y" THENPRI
NT#1,USINGF3;A(0,6),A(0,7),A(0,8),A(0,9)
,A(0,10),A(0,11)
1380 PRINT"END OF RUN":PRINT:INPUT"PRESS
 <ENTER> TO CONTINUE ";IN$:CLS:GOTO80
1390 'SAVE NEW FILE
1400 PRINT"PREPARE CASSETTE. TYPE NUMBER
 OF EMPLOYEE RECORDS ON FILE,":INPUT"THE
N PRESS <ENTER> WHEN READY :";RC%:PRINT"
DATA NOW BEING WRITTEN TO TAPE....."
1410 OPEN"O",#1,"CAS0:DATA"
1430 PRINT#1,RC%
1440 FOR I=1 TO RC%
1450 PRINT#1,A(I,1),A(I,2),A(I,3),A(I,4)
,A(I,5),A(I,6),A(I,7),A(I,8),A(I,9),A(I,
10),A(I,11)
1460 NEXTI
1470 CLOSE
1480 PRINT".....DATA SAVE COMPLETE"
1490 FORT=1TO1000:NEXTT:CLS:GOTO80
1500 'EMPLOYEE ENQUIRY (YEAR TO DATE TOT
ALS)
1510 CLS:PRINT:PRINT
1520 INPUT"IS THE PRINTER REQUIRED (Y/N)
";LP$
1530 IF LP$="Y" THEN OPEN "O",1,"LPT0:":
PRINT"SWITCH PRINTER ON"
1540 INPUT"ENTER REQUIRED PAY NUMBER :";
PN%
1550 RESTORE:I=1
1560 READ MF%(I),MF$(I),MR(I),MS(I),MT(I
)
1565 IF MF%(I)=99 THEN PRINT"INVALID PAY
 NUMBER":GOTO 1540
1570 IF MF%(I)=PN% THEN 1580 ELSEI=I+1:G
OTO1560
1580 DA$="Y. T. D."
1590 PRINT:PRINTH1$:IF LP$="Y" THENPRINT
#1,H1$
1600 PRINTUSINGF1;PN%,MF$(I),DA$:PRINT:I
F LP$="Y" THENPRINT#1,USINGF1;PN%,MF$(I)
,DA$
1610 PRINTH2$:IF LP$="Y" THENPRINT#1,H2$
1620 PRINTUSINGF4;A(I,2),A(I,3),A(I,4),A
(I,5),A(I,6):IF LP$="Y" THENPRINT#1,USIN
GF4;A(I,2),A(I,3),A(I,4),A(I,5),A(I,6)
1630 PRINT:PRINTH3$:IF LP$="Y" THENPRINT
#1,H3$
1640 PRINTUSINGF3;A(I,6),A(I,7),A(I,8),A
(I,9),A(I,10),A(I,11):IF LP$="Y" THENPRI
NT#1,USINGF3;A(I,6),A(I,7),A(I,8),A(I,9)
,A(I,10),A(I,11)
1650 PRINT:PN%=PN%+1:INPUT"ENTER NEXT PA
Y NUMBER (999 TO EXIT) ";PN%:CLS
1660 IF PN%=999 THEN CLS :GOTO 80
1670 GOTO1550
1680 PRINT" ***** INVALID PAY NUMBER":RE
SUME NEXT
```

**** LII/16K   JUMP THE RAPIDS ****

TRS-80/SYSTEM-80

```
10 REM                    *** JUMP THE RAPIDS ***
20 REM BRIAN GREEN. 16 JAMES ST. WHYALLA. S.A.5608. PHONE 452450
ASSISTED TONY WAUGH.
30 REM THIS PROGRAM NEEDS <MOVIE> ROUTINE AS A GRAPHICS DRIVER
40 REM TITLE GRAPHICS ROUTINE FROM 80 MICROCOMPUTING. SEP 1981
50 CLS:CLEAR75:GOSUB1160
60 GOSUB1070
70 HS=0
80 CLS:PRINT@25,"HIGH - SCORE";:PRINT@91,HS;
90 C1=200:M=0:V=20:P=47:NS=HS
100 PRINT@22,CHR$(191);:PRINT@39,CHR$(191);:PRINT@86,CHR$(143);:
PRINT@103,CHR$(143);
110 PRINT@128,STRING$(64,175);
120 PRINT@960,STRING$(3,190);:PRINT@968,STRING$(55,190);
130 FORX=8TO45:FORY=42TO43:SET(X,Y):NEXT:NEXT
140 FORX=80TO110:FORY=42TO43:SET(X,Y):NEXT:NEXT
150 FORX=70TO90:FORY=36TO38:SET(X,Y):NEXT:NEXT
160 FORX=0TO20:FORY=36TO37:SET(X,Y):NEXT:NEXT
170 FORX=15TO30:FORY=30TO32:SET(X,Y):NEXT:NEXT
```

```
180 FORX=80TO90:Y=30:SET(X,Y):NEXTX
190 FORX=20TO25:FORY=24TO26:SET(X,Y):NEXT:NEXT
200 FORX=60TO65:Y=24:SET(X,Y):NEXTX
210 FORX=110TO115:FORY=24TO25:SET(X,Y):NEXT:NEXT
220 FORX=0TO5:FORY=18TO19:SET(X,Y):NEXT:NEXT
230 FORX=40TO43:Y=18:SET(X,Y):NEXTX
240 FORX=100TO103:FORY=18TO19:SET(X,Y):NEXT:NEXT
250 FORX=20TO22:FORY=12TO13:SET(X,Y):NEXT:NEXT
260 FORX=60TO62:Y=12:SET(X,Y):NEXTX
270 FORX=100TO102:Y=12:SET(X,Y):NEXTX
280 PRINT@40,"SAFE >";:PRINT@0,"SCORE    =";:PRINT@11,C1;:PRINT@
64,"MEN LEFT =";:PRINT@75,V;
290 PRINT@58,"HOME";:PRINT@121,CHR$(143)+"     "+CHR$(143);
300 W=10:Z=47:SET(W,Z)
310 GOSUB600:GOSUB620:GOSUB640:GOSUB660:GOSUB680:GOSUB700
320 A$=INKEY$
330 IFA$="J":Z=Z-6:GOSUB430
340 IFZ=41S=832:GOSUB480
350 IFZ=35S=704:GOSUB490
360 IFZ=29S=576:GOSUB480
370 IFZ=23S=448:GOSUB490
380 IFZ=17S=320:GOSUB480
390 IFZ=11S=192:GOSUB490
400 IFZ=5S=64:GOSUB720
410 IFC1<5GOTO890
420 GOTO310
430 '******
440 RESET(W,Z+6)
450 SET(W,Z)
460 IFPOINT(W,Z+1)RETURNELSEGOTO550
470 GOTO310
480 W=W+2:GOTO500
490 W=W-2
500 IFW>126RESET(W-2,Z):GOTO530
510 IFW<1RESET(W+2,Z):GOTO530
520 RETURN
530 C1=C1-10:PRINT@11,C1;:PRINT@468,"GOOD-BYE MATE !";:FORI=1TO1
000:NEXT:PRINT@468,STRING$(15,128);:W=10:Z=47:SET(W,Z):GOSUB800:
RETURN
540 RETURN
550 IFZ=41S=960
560 T=INT(W/2)-130:PRINT@S+T,"SPLASH";:GOSUB800
570 C1=C1-10:PRINT@11,C1;
580 FORI=1TO300:NEXT:PRINT@S+T,"      ";
590 W=10:Z=47:SET(W,Z):GOTO310
600 RSET(13,15,0)
610 RETURN
620 LSET(11,13,0)
630 RETURN
640 RSET(9,11,0)
650 RETURN
660 LSET(7,9,0)
670 RETURN
680 RSET(5,7,0)
690 RETURN
```

```
700 LSET(3,5,0)
710 RETURN
720 F1$=CHR$(132)
730 PRINT@P,F1$;:GOSUB800
740 U=W:RESET(W,Z)
750 C1=C1+U:M=M+1:PRINT@122,M;:IFM=10GOTO830
760 PRINT@11,C1;:PRINT@75,V;
770 W=10:Z=47:SET(W,Z)
780 P=P+1
790 RETURN
800 IFV=0ANDM<10GOTO890
810 IFV+M<11GOTO890
820 V=V-1:PRINT@75,V;:RETURN
830 FORI=1TO200:CLS:PRINT@465,"!! HOORAY - YOU MADE IT !!"
840 HS=C1:PRINT@598,"SCORE > ";:PRINT@607,HS:PRINT:PRINT"
     PRESS <ENTER> FOR ANOTHER TRY."
850 IFNS>HSTHENHS=NS
860 A$=INKEY$
870 IFA$=""GOTO860
880 GOTO80
890 CLS:PRINT@468,"YOU BLEW IT !!!":FORI=1TO2000:NEXT
900 CLS:PRINT:PRINT,"PRESS <ENTER> FOR ANOTHER TRY."
910 A$=INKEY$
920 IFA$=""GOTO910
930 GOTO80
940 CLS:PRINT@20,"*** JUMP THE RAPIDS ***";
950 PRINT@200,"THE OBJECT OF THE GAME IS TO GET 10 MEN SAFELY AC
ROSS   THE RAPIDS."
960 PRINT"YOUR MEN WILL BE POSITIONED ONE AT A TIME ON THE NEARE
ST BANK."
970 PRINT"USING THE JUMP BUTTON <J> JUMP FROM LOG TO LOG UNTIL Y
OU REACH  THE OPPOSITE BANK."
980 PRINT"YOU HAVE TWENTY MEN. EACH MAN IS WORTH 10 POINTS."
990 PRINT"YOU GAIN POINTS FOR EVERY MAN YOU GET SAFELY ACROSS."
1000 PRINT"THE CLOSER A MAN JUMPS TO THE HOME POSITION, THE MORE
 POINTS HE SCORES."
1010 PRINT"EACH MAN LOST COSTS YOU 10 POINTS. "
1020 PRINT"YOU MUST GET  TEN MEN ACROSS TO QUALIFY FOR A SCORE."
1030 PRINT"PRESS ANY KEY TO START GAME ---"
1040 A$=INKEY$
1050 IFA$=""GOTO1040
1060 GOTO70
1070 CLS:PRINT"DID YOU LOAD MOVIE ?.....
IF NOT, PRESS <BREAK>...TYPE NEW...LOAD <MOVIE>
RELOAD <JUMP THE RAPIDS>
IF <MOVIE> IS LOADED...PRESS ANY KEY TO CONTINUE----"
1080 A$=INKEY$:IFA$=""GOTO1080
1090 CLS:PRINT@460,"DO YOU NEED INSTRUCTIONS ?  <Y/N>"
1100 A$=INKEY$
1110 IFA$="Y"GOTO940
1120 IFA$="N"RETURN
1130 IFA$=""GOTO1100
1140 GOTO1140
1150 REM *** TITLE GRAPHICS ***
1160 T=854:R=74
```

```
1170 T=T+1:R=R+1:Q=Q+1:ONQGOTO1190,1200,1210,1180,1190,1200,1210
,1180,1190,1200,1210
1180 B$=CHR$(150)+CHR$(141)+CHR$(172)+CHR$(178)+CHR$(174):GOTO12
20
1190 B$=CHR$(137)+CHR$(140)+CHR$(174)+CHR$(171)+CHR$(131):GOTO12
20
1200 B$=CHR$(151)+CHR$(171)+CHR$(186)+CHR$(162)+CHR$(167):GOTO12
20
1210 B$=CHR$(149)+CHR$(176)+CHR$(176)+CHR$(152)+CHR$(131)
1220 FORU=0TO65STEP5:PRINT@U,B$;:NEXT
1230 FORU=110TO845STEP65:PRINT@U,B$+B$+B$+B$+B$;:NEXT
1240 FORU=890TO955STEP5:PRINT@U,B$;:NEXT
1250 FORU=TTORSTEP-65:PRINT@U," JUMP THE RAPIDS !";:NEXT
1260 IFT<866GOTO1170
1270 FORU=1TO200:NEXT:RETURN
```

**** LII/16K   FAULT FINDER ****

TRS-80/SYSTEM-80

```
10 CLS:PRINT@26,"FAULT FINDER":PRINT@90,STRING$(12,"="):PRINT"
   I HAVE AN ENGINE THAT IS SUBJECT TO THREE FAULTS:-":PRINTCHR$
(204);"THE WIDGET WUMBLES,":PRINTCHR$(204);"THE STUGS STICK,":PR
INTCHR$(200)"AND THE SPROCKETS FALL OFF.":PRINT
20 PRINT"     THESE FAULTS OCCUR INTERMITTENTLY, APPARENTLY AS A
":PRINT"RESULT OF ACTIONS I HAVE TAKEN.":PRINT"     WHAT I WOULD
 LIKE YOU TO DO IS TO DIRECT ME TO TAKE VARIOUS";
30 PRINT"COMBINATIONS OF THESE ACTIONS, AND I SHALL REPORT THE R
ESULTS.":PRINT"     FROM THIS, YOU MAY BE ABLE TO TELL ME WHAT I
 SHOULD AVOID":PRINT"DOING, IN ORDER THAT THE THREE FAULTS SHOUL
D NOT OCCUR."
40 PRINT@973,"PRESS ANY KEY TO CONTINUE.";
50 Z$="":Z$=INKEY$:IFZ$=""THEN50ELSECLS
60 PRINT@256,"     I HAVE GIVEN EACH ACTION A NUMBER (1-6) AND Y
OU CAN":PRINT"ORDER ME TO PERFORM A COMBINATION BY ENTERING THE
NUMBERS.":PRINT"THUS, 135 WOULD MEAN I WAS TO PERFORM ACTION 1,
THEN ACTION 3,"
70 PRINT"THEN ACTION 5.":PRINT"     IF YOU THINK YOU HAVE DEDUCE
D THE ANSWER, YOU JUST ENTER   999.":PRINT"     FOR TECHNICAL RE
ASONS, I AM UNABLE TO CARRY OUT MORE THAN":PRINT"FOUR ACTIONS AT
 ANY ONE TIME."
80 PRINT@973,"PRESS ANY KEY TO CONTINUE.";
90 Z$="":Z$=INKEY$:IFZ$=""THEN90ELSECLS
100 CLS:Z$="":DEFINTJ-N:PRINT"THE ACTIONS ARE:-1. TAKE OFF THE E
NGINE LID.":PRINTCHR$(209);"2. PULL BACK LEVER MARKED 'FORWARD'.
":PRINTCHR$(209);"3. PRESS BUTTON B.":PRINTCHR$(209);"4. REMOVE
SAFETY CATCH."
110 PRINTCHR$(209);"5. TAP PLATE C (MARKED 'DO NOT TOUCH').":PRI
NTCHR$(209);"6. PULL OUT THE BIG RED KNOB."
120 INPUT"PLEASE ENTER A COMBINATION (OR 999)";Z$
130 IFZ$=""THEN100ELSEIFZ$="999"THEN310
140 IFLEN(Z$)>4THENPRINT:PRINT"I'M SORRY: I CAN'T CARRY OUT MORE
 THAN FOUR ACTIONS!":GOTO360
150 J1=0:J=0:K=1:L=0:M=464:FORN=1TOLEN(Z$):X$=MID$(Z$,N,1):IFX$=
"2"THENJ1=1:K=K+1:GOSUB270ELSEIFX$="1"THENGOSUB220ELSEIFX$="6"TH
ENGOSUB220ELSEIFX$="3"THENGOSUB270ELSEIFX$="4"THENGOSUB210ELSEIF
X$="5"THENGOSUB220
160 NEXT:IFJ1>0THEN180ELSEIFL>0ANDK>1THENGOSUB280
170 IFJ>0ANDK>1THENGOSUB290
180 IFL>0ANDJ>0THENGOSUB300
190 PRINT@973,"PRESS ANY KEY TO CONTINUE.";
200 Z$="":Z$=INKEY$:IFZ$=""THEN200ELSECLS:GOTO100
210 IFL>0THENRETURNELSEPRINT@M,"THE SNIFTING VALVE VENTS":M=M+64
:L=L+1:IFK=2THEN240ELSERETURN
220 ONKGOTO230,240,250,260,260,260,260
230 PRINT@M,"THE INDEX REGISTER TURNS BLUE":K=K+1:M=M+64:RETURN
240 PRINT@M,"THE GUDGEONS JAMBLE":K=K+1:M=M+64:RETURN
250 PRINT@M,"THE ENGINE BOILS":K=K+1:M=M+64
260 RETURN
270 IFJ>0THENRETURNELSEPRINT@M,"THE ANEMOMETER QUIVERS":J=J+1:M=
M+64:IFK=2THEN240ELSERETURN
280 PRINT@M,"THE WIDGET WUMBLES":M=M+64:RETURN
290 PRINT@M,"THE STUGS STICK":M=M+64:RETURN
300 PRINT@M,"THE SPROCKETS FALL OFF":M=M+64:RETURN
310 INPUT"WHICH ACTION MUST I AVOID TO STOP THE WIDGET WUMBLING"
;Z$:IFZ$<>"4"THEN350
320 INPUT"WHICH ACTION MUST I AVOID TO STOP THE STUGS STICKING";
Z$:IFZ$<>"3"THEN350
330 INPUT"AND THE SPROCKETS FALLING OFF";Z$:IFZ$<>"4"THEN350
340 PRINT:PRINT"CONGRATULATIONS! YOU HAVE IT ALL CORRECT.":PRINT
:END
350 PRINT:PRINT"THAT IS NOT CORRECT! TRY AGAIN."
360 FORN=1TO800:NEXT:CLS:GOTO100
```

**** LII/16K   CHEQUE ACCOUNT MANAGER ****

TRS-80/SYSTEM-80

```
10 POKE16561,186:POKE16562,127:GOTO180
20 '***    CHEQUE ACCOUNT MANAGER   VERSION 1.0         ***
30 '***    BY DAVE POWELL                  DEC 81       ***
40 '***    MORPHETT VALE S.A.                            ***
50 '***    UTILITY 'ARRAY SAVER' WRITTEN BY K.SHILLITO,  ***
60 '***    MICRO-80 ISSUE 22. NOTE RE-LOCATION TO HIGH   ***
70 '***    MEMORY TO AVOID DAMAGE TO VARIABLES.          ***
80 '***    CSAVE PROGRAM ONTO START  OF CASSETTE, THEN   ***
90 '***    TYPE 'RUN 1270' TO INITIATE DATA FILES.       ***
100 K$=INKEY$:IFK$=""THEN100ELSEIFASC(K$)<>13THEN100ELSERETURN
110 K$=INKEY$:IFK$=""THEN110ELSEIFK$<>"N"ANDK$<>"Y"THEN110ELSEPR
INTK$;:RETURN
```

```
120 LB$=STR$(LB):I%=8-LEN(LB$):BB$=STR$(BB):J%=8-LEN(BB$):CH$(0)
=D$+STRING$(I%,32)+LB$+STRING$(J%,32)+BB$:CH$(0)=CH$(0)+STRING$(
50-LEN(CH$(0)),35):RETURN
130 LB=VAL(MID$(CH$(0),9,8)):BB=VAL(MID$(CH$(0),17,8)):RETURN
140 J=J+1:IFJ/10=INT(J/10)PRINT@192,CHR$(31);
150 RETURN
160 K$=INKEY$:IFK$=""THEN160ELSEK=VAL(K$):RETURN
170 PRINT"IMPROPER FORMAT....MATCH ABOVE FIELDS":FORZ=0TO1000:NE
XT:J=9:GOSUB140:PRINT"  "CH$(I)"  <"I">":RETURN
180 CLEAR3500:CLS:FORI=0TO14:PRINTSTRING$(1,191);:PRINTSTRING$(6
2,32);:PRINTSTRING$(1,191);:NEXT:PRINT@1,STRING$(62,131);:PRINT@
961,STRING$(62,176);:POKE16320,191:POKE16383,191
190 PRINT@269,"CHEQUE ACCOUNT MANAGER  -  VERSION 1.0";:PRINT@84
8,"( PRESS 'NEW LINE' TO CONTINUE )";:DIMCHEQUE$(50):GOSUB100
200 CLS:INPUT"ENTER DATE (DD/MM/YY)";D$:IFLEN(D$)<>8ORMID$(D$,3,
1)<>"/"ORMID$(D$,6,1)<>"/"THEN200ELSEPRINT@128,"WIND TAPE TO (03
0) FOR DATA INPUT, PRESS PLAY THEN 'NEW LINE'"
210 FORI%=0TO50:CH$(I%)=STRING$(50,35):NEXT:J%=0:BY%=51:GOSUB100
:PRINT"READING FROM TAPE....":PRINT"( IN THE EVENT OF FAILURE PR
ESS 'RESET', TYPE 'GOTO 1250' )":AD%=VARPTR(CH$(0)):GOSUB1190
220 IFMID$(CH$(0),3,1)<>"/"ORMID$(CH$(0),6,1)<>"/"THEN1250
230 PRINT:PRINT"DATA LOADING COMPLETE.  DATE OF SAVING DATA WAS:
 ";:IFLEFT$(CH$(0),1)<>"#"THENPRINTLEFT$(CH$(0),8)ELSEPRINT"( NE
W USER )":GOSUB1260:GOTO250
240 GOSUB130
250 PRINT:PRINT"PRESS 'NEW LINE' FOR MENU":GOSUB100:U$="+$$####.
##":SF=0
260 CLS:PRINT@18,"CHEQUE ACCOUNT MANAGER":PRINT@146,"1. ENTER NE
W CHEQUES":PRINT@210,"2. BALANCE ACCOUNT":PRINT@274,"3. EXAMINE
RECORDS":PRINT@338,"4. EDIT RECORDS":PRINT@402,"5. QUIT":PRINT@6
58,"SELECT (1 - 5)?";
270 GOSUB160:IFK<1ORK>5THEN270
280 CLS:ONKGOTO290,430,540,870,800
290 FORI=1TO50:IFCH$(I)=STRING$(50,35)THENJ=I:GOTO310ELSENEXT
300 PRINT"RECORDS CONTAIN 50 CHEQUES......BALANCE ACCOUNT TO RED
UCE.":FORI=0TO2000:NEXT:GOTO260
310 GOSUB400
320 FORP=0TO9
330 INPUTCH$(J):IFCH$(J)="Q"THENCH$(J)=STRING$(50,35):GOTO380
340 IFMID$(CH$(J),3,1)<>"/"ORMID$(CH$(J),6,1)<>"/"ORMID$(CH$(J),
9,2)<>"  "ORMID$(CH$(J),17,2)<>"  "ORMID$(CH$(J),42,2)<>"  "ORLE
N(CH$(J))<>50ORVAL(RIGHT$(CH$(J),7))=0THENPRINT"IMPROPER FORMAT.
...MATCH ABOVE FIELDS":P=P+1:GOTO370
350 CH=VAL(RIGHT$(CH$(J),7)):IFMID$(CH$(J),11,3)<>"DEP"THENCH=-C
H
360 CU=PEEK(16417)*256+PEEK(16416)-10:POKE16417,INT(CU/256):POKE
16416,CU-INT(CU/256)*256:BB=BB+CH:GOSUB410:PRINTUSINGU$;BB;:J=J+
1:IFJ>50THENPRINT"WAIT...";:SF=1:GOSUB1080:PRINT:GOTO300
370 NEXTP:GOSUB400:PRINT"  "CH$(J-1):GOTO320
380 IFJ>ITHENSF=1:GOSUB1080
390 GOTO260
400 CLS:PRINT"*** ENTER NEW CHEQUES ***  (TYPE 'Q' FOR QUIT)":PR
INT"'CHEQUE' = '123456' OR 'DEP   ' OR 'FEE   '":PRINT@60,"BOOK"
:PRINT@121,"BALANCE  DD/MM/YY  CHEQUE  ----- PARTICULARS -----
$$$$.CC";:PRINT@182,USINGU$;BB;:RETURN
```

```
410 IFBB>-.01ANDBB<.01THENBB=0
420 RETURN
430 SO=0:V3=0:J=0:CLS:PRINT"*** BALANCE ACCOUNT ***":PRINT@49,"B
 - BYPASS ITEMLAST BALANCE "USINGU$;LB:PRINT@103,"C - CANCEL OUT
   Q - QUITDD/MM/YY  CHEQUE  ----- PARTICULARS -----  $$$$.CC":P
RINT@185,"BALANCE";:V1=LB
440 FORI=1TO50:IFCH$(I)=STRING$(50,35)THENI=50:GOTO490ELSEGOSUB1
40:PRINTCH$(I);
450 GOSUB160:IFK$="Q"THENI=50:V3=1:GOTO480ELSEIFK$="B"THEN480ELS
EIFK$<>"C"THEN450
460 SO=1:V2=VAL(RIGHT$(CH$(I),7)):CH$(I)=CH$(I)+K$:IFMID$(CH$(I)
,11,3)="DEP"THENV2=-V2
470 V1=V1-V2
480 PRINT"  "K$" "USINGU$;V1;
490 NEXT:IFV3=0THENPRINT:PRINT"NO MORE ITEMS"
500 IFSO=0THEN260ELSEPRINT:PRINT"NEW BALANCE "USINGU$;V1
510 PRINT:PRINT"ACCEPT BALANCE (Y/N)?";:GOSUB110:PRINT:IFK$="N"T
HENPRINT"  RESTORING RECORDS-":FORI=1TO50:CH$(I)=LEFT$(CH$(I),50
):NEXT:GOTO430
520 PRINT:PRINT"AMENDING RECORDS-";:SF=1:LB=V1:FORI=1TO50:IFLEN(
CH$(I))>50THENCH$(I)=STRING$(50,35)
530 NEXT:PRINT" DONE":GOSUB1080:GOTO260
540 K=0:PRINT"*** EXAMINE RECORDS ***":PRINT:PRINT"1. BY DATE":P
RINT"2. BY CHEQUE #":PRINT"3. DEPOSITS":PRINT"4. FEES":PRINT"5.
BALANCES":PRINT"6. ALL":PRINT"7. QUIT":PRINT:PRINT"SELECT (1-7)?
"
550 GOSUB160:IFK<1ORK>7THEN550ELSECLS:D=0:P=0:ONKGOTO560,660,690
,700,790,720,260
560 INPUT"FROM DATE (DD/MM/YY)";D2$:IFMID$(D2$,3,1)<>"/"ORMID$(D
2$,6,1)<>"/"ORLEN(D2$)<>8THEN560
570 INPUT"TO DATE (DD/MM/YY)";D3$:IFMID$(D3$,3,1)<>"/"ORMID$(D3$
,6,1)<>"/"ORLEN(D3$)<>8THEN570ELSEIFD2$=D3$THEND=1:GOTO590
580 Y2$=RIGHT$(D2$,2):Y3$=RIGHT$(D3$,2):M2$=MID$(D2$,4,2):M3$=MI
D$(D3$,4,2):D2$=LEFT$(D2$,2):D3$=LEFT$(D3$,2)
590 PRINT:FORI=1TO50:IFCH$(I)=STRING$(50,35)THENI=50:GOTO770
600 IFD=1ANDLEFT$(CH$(I),8)=D2$THEN650ELSEIFD=1THEN770
610 Y1$=MID$(CH$(I),7,2):M1$=MID$(CH$(I),4,2):D1$=LEFT$(CH$(I),2
):IFY1$<Y2$ORY1$>Y3$THEN770ELSEIFY1$=Y2$ANDY1$=Y3$THEN640ELSEIFY
1$=Y2$THEN630
620 IFM1$>M3$THEN770ELSEIFM1$<M3$THEN650ELSEIFD1$>D3$THEN770ELSE
650
630 IFM1$<M2$THEN770ELSEIFM1$>M2$THEN650ELSEIFD1$<D2$THEN770ELSE
650
640 IFM1$<M2$ORM1$>M3$THEN770ELSEIFM1$=M2$ANDD1$<D2$THEN770ELSEI
FM1$=M3$ANDD1$>D3$THEN770ELSE650
650 PRINTCH$(I)"   <"I">":P=1:GOTO770
660 INPUT"FROM #";V1:INPUT"TO #";V2:IFV2<V1THEN660ELSEIFV2=V1THE
ND=1
670 PRINT:FORI=1TO50:IFCH$(I)=STRING$(50,35)THENI=50:GOTO770ELSE
K=VAL(MID$(CH$(I),11,6)):IFD=1ANDK=V1THEN650ELSEIFD=1THEN770
680 IFK>=V1ANDK=<V2THEN650ELSE770
690 FORI=1TO50:IFCH$(I)=STRING$(50,35)THENI=50:GOTO770ELSEIFMID$
(CH$(I),11,3)="DEP"THEN650ELSE770
700 FORI=1TO50:IFCH$(I)=STRING$(50,35)THENI=50:GOTO770
710 IFMID$(CH$(I),11,3)="FEE"THEN650ELSE770
```

```
720 PRINT"DURING LISTING - PRESS 'SPACE-BAR' TO PAUSE/CONTINUE..
..":PRINT
730 K$=INKEY$:IFK$=""THEN730ELSEIFASC(K$)<>32THEN730ELSEFORI=1TO
50:IFCH$(I)=STRING$(50,35)THENI=50:GOTO770
740 K$=INKEY$:IFK$=""THEN750ELSEIFASC(K$)=32THEN760
750 GOTO650
760 K$=INKEY$:IFK$=""THEN760ELSEIFASC(K$)<>32THEN760ELSE750
770 NEXTI:IFP=0PRINT"NO RECORD FOUND"
780 PRINT:PRINT"END OF SEARCH.  PRESS 'NEW LINE' FOR MENU":GOSUB
100:CLS:GOTO540
790 PRINT"LAST BANK STATEMENT BALANCE"USINGU$;LB:PRINT:PRINT"BOO
K BALANCE"USINGU$;BB:PRINT:GOTO780
800 GOSUB810:CLS:PRINT"CHEQUE ACCOUNT MANAGER TERMINATED.":PRINT
"( TYPE 'GOTO 260' TO RESUME )":PRINT:PRINT:END
810 IFSF<>1RETURN
820 GOSUB120:PRINT"SAVE DATA TO TAPE BEFORE QUITTING.":PRINT:PRI
NT"WIND TAPE TO 030"
830 GOSUB860:PRINT"WIND TAPE TO 050 FOR BACK-UP COPY":GOSUB860:S
F=0
840 PRINT"DO YOU WISH TO MAKE ANOTHER COPY (Y/N)?";:GOSUB110:PRI
NT:IFK$="N"RETURN
850 PRINT:PRINT"INSERT TAPE, ";:GOSUB860:GOTO840
860 BY%=51:PRINT"PRESS PLAY & RECORD, THEN 'NEW LINE'":GOSUB100:
PRINT"WRITING TO TAPE....":AD%=VARPTR(CH$(0)):GOSUB1180:PRINT:RE
TURN
870 GOSUB1050:PRINT"*** EDIT RECORDS ***":PRINT@33,"D - DELETE I
TEM   E - EDIT ITEM":PRINT@84,"N - NEXT ITEM   P - PREVIOUS ITEM
   Q - QUIT  DD/MM/YY  CHEQUE  ----- PARTICULARS -----  $$$$.CC
 ITEM":J=0:I=0:SO=0
880 I=I+1:IFI>=51THENI=51:GOTO890ELSEIFCH$(I)=STRING$(50,35)THEN
890ELSEGOSUB140:PRINT"  "CH$(I)"  <"I">":GOTO900
890 GOSUB140:PRINT"NO MORE ITEMS":I=I-1
900 GOSUB160:IFK$<>"E"ANDK$<>"D"ANDK$<>"N"ANDK$<>"P"ANDK$<>"Q"TH
EN900
910 IFK$="N"THEN880ELSEIFI<=1ANDK$="P"THENGOSUB140:PRINT"TOP OF
RECORD":I=0:GOTO880ELSEIFK$="P"THENI=I-2:GOTO880
920 IFK$="D"THEN950ELSEIFK$="E"THENJ=9:GOSUB140:PRINT"  "CH$(I)"
  <"I">":GOTO980
930 IFSO=1PRINT:PRINT"FIRST....";:GOSUB1080
940 GOTO260
950 PRINT"DELETE THE ABOVE STRING (Y/N)?";:GOSUB110:IFK$="N"THEN
PRINT" NOT DELETED.  NEW COMMAND?":GOSUB140:GOTO900
960 V3=VAL(RIGHT$(CH$(I),7)):IFMID$(CH$(I),11,3)<>"DEP"THENV3=-V
3
970 PRINT" DELETED":GOSUB140:GOSUB1040:CH$(I)=STRING$(50,35):SF=
1:GOSUB140:GOSUB1080:I=I-1:GOTO900
980 E$=CH$(I):INPUTE$:IFMID$(E$,3,1)<>"/"ORMID$(E$,6,1)<>"/"ORMI
D$(E$,9,2)<>"  "ORMID$(E$,17,2)<>"  "ORMID$(E$,42,2)<>"  "ORLEN(
E$)<>50ORVAL(RIGHT$(E$,7))=0THENGOSUB170:GOTO980
990 IFE$<>CH$(I)THENJ=9:GOSUB140:GOSUB1020:SO=1:SF=1:GOSUB140:PR
INT"CHANGE NOTED.";:GOTO1010
1000 GOSUB140:PRINT"NO CHANGE.";
1010 PRINT"  COMMAND?":GOTO900
1020 V1=VAL(RIGHT$(E$,7)):V2=VAL(RIGHT$(CH$(I),7)):IFV1=V2THENCH
$(I)=E$:RETURNELSEV3=V1-V2
1030 IFMID$(E$,11,3)="DEP"THENV3=-V3
1040 BB=BB-V3:GOSUB410:GOSUB140:PRINT"BOOK BALANCE AMENDED TO"US
INGU$;BB:CH$(I)=E$:RETURN
1050 PRINT:PRINT"BANK STATEMENT BALANCE "USINGU$;LB:PRINT:PRINT"
EDIT (Y/N)?";:GOSUB110:IFK$="N"THEN1060ELSEINPUT"   NEW VALUE";V
1:IFV1=LBTHEN1050ELSESF=1:LB=V1:GOTO1050
1060 PRINT:PRINT:PRINT"BOOK BALANCE "USINGU$;BB:PRINT:PRINT"EDIT
 (Y/N)?";:GOSUB110:IFK$="N"THEN1070ELSEINPUT"   NEW VALUE";V2:IF
V2=BBTHEN1060ELSESF=1:BB=V2:GOSUB410:GOTO1060
1070 CLS:RETURN
1080 PRINT"SORTING RECORDS -";:E$=STRING$(50,35):FORK=49TO1STEP-
1:Z=0:FORJ=1TOK:IFCH$(J+1)<>E$ANDCH$(J)=E$THEN1150ELSEIFCH$(J)=E
$ANDCH$(J+1)=E$THEN1130ELSEIFCH$(J+1)=E$THEN1130
1090 Y1$=MID$(CH$(J),7,2):Y2$=MID$(CH$(J+1),7,2):IFY1$>Y2$THEN11
50
1100 M1$=MID$(CH$(J),4,2):M2$=MID$(CH$(J+1),4,2):IFM1$>M2$ANDY1$
=Y2$THEN1150
1110 IFLEFT$(CH$(J),2)>LEFT$(CH$(J+1),2)ANDM1$=M2$ANDY1$=Y2$THEN
1150
1120 IFLEFT$(CH$(J),8)=LEFT$(CH$(J+1),8)ANDMID$(CH$(J),11,6)>MID
$(CH$(J+1),11,6)THEN1150
1130 NEXTJ:IFZ=0THENK=1
1140 NEXTK:PRINT"  COMPLETED":RETURN
1150 K$=CH$(J):CH$(J)=CH$(J+1):CH$(J+1)=K$:Z=1:GOTO1130
1160 GOSUB1210:GOSUB1230:GOSUB1220:GOTO1200
1170 GOSUB1210:GOSUB1230:GOTO1200
1180 GOSUB1210:GOSUB1220:GOTO1200
1190 GOSUB1210
1200 I%=USR(0):RETURN
1210 POKE16553,255:RESTORE:FORI%=32700TO32759:READJ%:POKEI%,J%:N
EXT:FORI%=16526TO16527:READJ%:POKEI%,J%:NEXT:I%=VARPTR(AD%):POKE
32709,PEEK(I%):POKE32710,PEEK(I%+1):I%=VARPTR(BY%):POKE32712,PEE
K(I%):POKE32713,PEEK(I%+1):RETURN
1220 POKE32706,135:FORI%=32742TO32747:READJ%:POKEI%,J%:NEXT:RETU
RN
1230 POKE32719,24:POKE32720,9:POKE32750,24:POKE32751,2:RETURN
1240 DATA62,0,205,18,2,205,150,2,33,0,0,17,0,0,27,203,122,32,37,
70,35,126,35,229,102,111,175,184,40,22,58,64,56,254,4,40,19,197,
1,255,0,205,44,2,205,53,2,119,193,35,16,234,225,35,24,214,205,24
8,1,201,188,127,96,0,126,205,100,2
1250 CLS:PRINT"EMERGENCY BACK-UP LOAD.":PRINT:PRINT"WIND TAPE TO
 (050), PRESS PLAY THEN 'NEW LINE'":GOTO210
1260 PRINT:PRINT"ENTER THE BALANCE OF THE LAST BANK STATEMENT TH
AT HAS BEEN":INPUT"CHECKED ";LB:BB=LB:GOSUB120:PRINT:PRINT"ENTER
 ALL CHEQUES/DEPOSITS/FEES NOT ACCOUNTED FOR BY THAT STATE-MENT.
":RETURN
1270 CLEAR3500:DIMCH$(50):CH$(0)="##/##/"+STRING$(44,35):FORI%=1
TO50:CH$(I%)=STRING$(50,35):NEXT:J%=0:BY%=51
1280 X=30:GOSUB1290:X=50:GOSUB1290:PRINT"PROGRAM READY, RUN WHEN
 REQUIRED.":END
1290 PRINT"WIND TAPE TO ("X") PRESS PLAY & RECORD, THEN 'NEW LIN
E":GOSUB100:PRINT"SAVING DATA FILE...":AD%=VARPTR(CH$(0)):GOSUB1
180:RETURN
```

***** 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.

** PROPERTY INVESTMENT LII/4K (80) **

Use this progam to calculate how much you can afford to borrow - just tell it the current interest rate and how much you want to borrow. Once you find out the size of the loan you can get, you can add in your deposit and put in the retail price and see what your repayments will be exactly.

** POTHOLE LII16K (80) **

This crazy game has some very weird people, (PEOPLE ???), walking around in it. You must keep moving a pothole cover around to stop them falling into the potholes. It's dead easy to start with, but then there are more and more people turning up at the same time, and more and more and more...

** HANGMAN (COLOUR) **

The popular word-guessing game for both young and old. Have fun and take some spelling practice at the same time. Graphics are included to make the game a little more interesting.

** CRICKET LII/16K (80) **

Play Cricket against your computer. Graphic display includes the stumps, the bowler, the fielders and you. The game has three different types of cricket - a one day match, test cricket and English cricket.

** FASTER LII/16K (80) **

All of you folks that still only have a cassette recorder are just going to love this program. It increases ALL of the cassette operations to 1000 baud, yes all of them - PRINT#-1, INPUT#-1, CLOAD, CLOAD? and CSAVE and it ADDS new abilities such as #LIST and #ON and best of all #CSAVEn"A" where n is a number from 1 to 255. It will CSAVE the program n times leaving a pause between each copy. The best news, though, is that while under test here at MICRO-80, the 1000 baud was found to be as reliable as the normal 500 baud.

** CALENDAR (COLOUR) **

With this program you can use your computer to produce a calendar on your printer, just like the one featured last issue.

APPLICATION FOR PUBLICATION OF A PROGRAM IN MICRO-80

Date ....

To **MICRO-80** SOFTWARE DEPT. PO BOX 145 MORPHETTVALE SA 5162
Please consider the enclosed program for . . .

Tick where appropriate

(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 if 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.

				APPROX.	START	POSITION
SIDE 1	TYPE	I.D.	DISK FILESPEC	CTR-41	CTR-80	SYSTEM 80
CHEQUE ACCOUNT MANAGER	LII/16K	C	CHEQUE/BAS	18	10	5
" " "	"	"	"	111	62	33
FAULT FINDER	LII/16K	F	FALTFIND/BAS	192	108	62
" "	"	"	"	218	123	72
JUMP THE RAPIDS	LII/16K	J	RAPIDS/BAS	245	138	80
" "	"	"	"	278	157	91
MORSE CODE TRANSMITTER	SYSTEM	MORSE	MORSE/CMD	311	176	105
" "	"	"	"	324	183	112
MORSE CODE TRANSMITTER	EDTASM	MORSE	MORSE/EDT	336	190	117
SIDE 2						
MORSE CODE TRANSMITTER	EDTASM	MORSE	MORSE/EDT	18	10	5
MOVIE UTILITY	SYSTEM	MOVIE	MOVIE/CMD	115	65	35
" "	"	"	"	125	71	38

TO:
MICRO-80, P.O. BOX 213, GOODWOOD,
SOUTH AUSTRALIA. 5034.

Please RUSH to me the items shown below:

$ .... enclosed Date ....

........................12 month subscription to MICRO-80

.... 12 mth. subs. to MICRO-80, plus the cassette edition

.... 12 mth. subs. to MICRO-80, plus the disc edition

.... The latest issue of MICRO-80 (see inside front cover for prices)

The MICRO-80 PRODUCTS listed below:

DESCRIPTION	PRICE
TOTAL ENCLOSED WITH ORDER	

□ Cheque □ Bankcard □ Money Order

Bankcard Account Number

Signature .... Exp. End ....

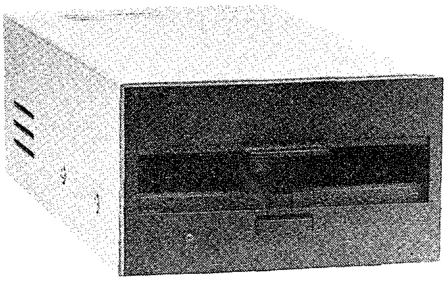
NAME ....

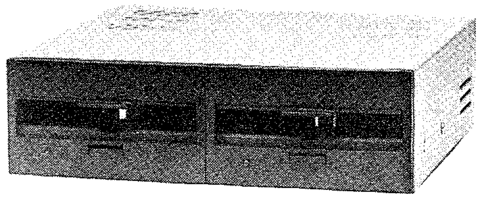
ADDRESS ....

.... Postcode ....

bankcard

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