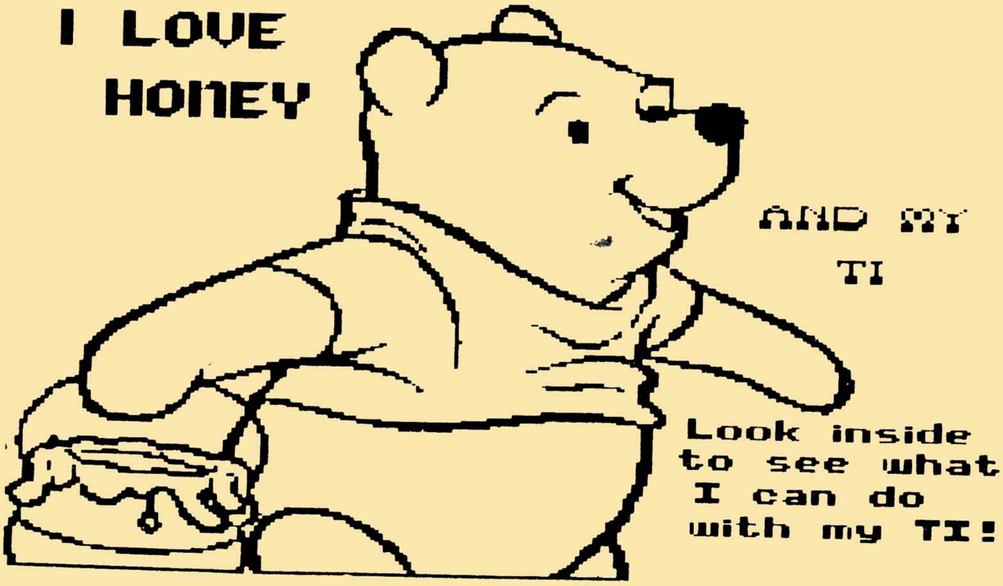
Issue 19

TIMES



Jan-Mar 1988

Issue 19
January 1988

A Happy New Year To All our Readers!

IN THIS ISSUE...

PAGE

1 Contact Addresses
2 Committee news
3 Your Letters
5 A Visit to the Netherlands Clive Scally
7 Rambles Stephen Shaw
36 Merge Format Boston Computer Society
43 Fitting a ZIF Socket and Reset Switch to a Romox Module Mike Goddard
46 Programming from Basic to Assembly Language Geoffrey Coan
50 Assembler Tutorial Sydney News Digest
52 Basic Beginnings GOTO Statement Christina
54 Speech Synthesizer as a Proof Reader Steven L. Richardson
56 File Dump Program in hex and ASC11 Chuck Reinhart
58 Flip Flop by the Tinman Ext. Basic Program Listing
60 For Sale and Wanted Ads

TI-99/4A USERS GROUP UK and TI*MES magazine is supported only by its subscribers. This TI user group is INDEPENDANT of Texas Instruments and is completely non-profit making. TI*MES is published quarterly. The annual subscription runs May to May each year. Details from the Group Secretary on request. Editorial, etc, is provided by group members, other WORLD WIDE TI User Groups, and other related sources. Views expressed are those of the writer and not necessarily those of TI-99/4A Users Group UK. While efforts are made to ensure accuracy, no responsibility can be accepted by TI-99/4A Users Group UK as a result of applying such information found within the pages of TI*MES . You are invited to contribute copy for publication in TI*MES. Please submit copy on A4 ONLY. This must be TYPED AND SUPPORTED WITH A DISK OR TAPE if a program is included. Last date of acceptance for next issue is 20th March, 1988 for publication in April. No material my be reproduced without credit to the author and TI-99/4A USERS GROUP UK.

Thanks for your support in 1987.

CONTACT ADDRESSES:

EDITORIAL:
Christina Mehew
Holly Croft, 4 Thornton Lane,
ULCEBY, South Humberside, DN39 6SG

MEMBERSHIP:
Peter Walker
24 Bacons Drive, CUFFLEY, Herts, EN6 4DU

DISK LIBRARY and JOURNAL EXCHANGE:
Stephen Shaw
10 Alstone Road, STOCKPORT, Ches, SK4 5AH

CASSETTE LIBRARY:
Maurice Rymill
231 Bournville Lane, Bournville,
BIRMINGHAM, B30 1RA

GENERAL SECRETARY:
Jim Ballinger
5 Offerton Avenue DERBY DE3 8DU

CHAIRMAN: Clive Scally

Other Committee Members:
Tim Anderson, Alan Bailey, Neville Bosworth
Bryan Cloud, Geoffrey Coan, Peter Cross,
Mike Goddard, Edward Shaw.

TIMES

Committee news

We are pleased to report that every committee member is working hard for you all, so that we can offer extra facilities, like a publications library, where you can borrow magazines and books for a small charge, to help you get to know your TI better. Proposals have been put forward for a module library, so that you can borrow different modules. I'm afraid I have to apologize to you all for confusing you yet again about the disk and cassette library. I referred to Tim Anderson as Disk Librarian – and gave him a shock! Sorry, Tim He'd only offered to help both the librarians with their massive job! This arose because Tim sent me a listing of a disk catalogue he had started as he was sorting out the useful programs held by the original Disk/Cassette Library. And in my last-minute panic I gave him his new title! Sheer incompetence on my part. So, our Secretary got busy to smooth the ruffled feathers, and clear up any gray areas still existing:

"It seems that confusion has arisen regarding the handling of Disk-based material Owing to some ambiguity in the recent issue of TI*MES.

To clear any doubt in member's minds:-
The Committee appointments are exactly as made at the Inaugural Meeting and published in the Commitee Members list circulated to each Committee member.

Stephen Shaw is Disk Librarian.

Maurice .Rymill is Cassette Librarian.

Tim Anderson offered (at the inaugural meeting) to offset some of the workload carried by Stephen and Maurice by vetting all the programmes submitted for inclusion in those libraries, and then passing those considered suitable to the appropriate librarian.(In a very recent letter Tim tells me that he is in no position to accept work until he has a permanent address.)'

Well, after preparing another grand issue of TI*MES for you, I can picture what all the committee will be doing on 25th December to 2nd or 3rd of January! Many thanks to Stephen Shaw for sending me some printed pictures and logos to use as covers for TI*MES, and also some contact addresses. Thanks also to Alan Bailey for doing some paste-ups for this issue. I am doing a computer course at home, and all of a sudden it has got hectic, so I'm very glad of any extra support anyone can give for the next issue. Exam is in May, so after that it's hunky dory.
Here's hoping this issue doesn't reach you too late. The printers are closing until 4th January, so I hope you are not suffering from withdrawal symptoms out there.

Best wishes,
Christina.

Your letters

Ian James writes: "I have bought a lot of my TI equipment through ads. in 'Micro Computer Mart'. It is a very useful fortnightly computer magazine. I often when reading the ads imagine what the seller could really mean. My observations:

Adverts:	Read between the lines!
Unused for twelve months	(Broken a year ago)
Head may need re-alignment	(Has been dropped)
Manual in loose-leaf binder	(Copy, original lost)
Pretty visual display	(Video processor wonky)
Excellent black and white	(No colour)
Supplied in original packing	(Required to hold it together)
Over £100 of software, yours for £10	(All pirated copies)

However, Ian thinks that computer owners are really very honest, and TI-99/4A owners are the nicest people around." Course, we all agree with that, Ian!

Barry Anderson wrote to let us know that he is pleased that we exist as a user group. He rescued his TI from the cupboard and telephoned Texas Instruments at Bedford. They sent him our telephone number. Well done, Texas! From having no cassettes and just the console, Barry now has six adventure cassettes, "and the wonderful TI*MES magazine. I would just like to thank Mr. Maurice Rymill, who sent me two back copies of the TI*MES mag and a listing for a game." Barry is after the Ext. Basic and chess modules. See the ads. at the back of this TI*MES.

NOTICE TO ALL TI*MES READERS

Following the latest feedback from membership questionnaires, I see that the most requested programming articles are on machine code, followed by c99, Extended Basic, and Forth. Stephen Shaw has sent me a large number of articles from international TI magazines, all suitable for publication in TI*MES. I shall send the articles in for publication in the above order; however, since about 1/2 of the articles are on Forth, they will never all get published.

If anyone has any preference as to what articles are published, or about the order of publication, or if you want photocopies of any of the Forth material, then write to me at this address: Geoffrey Coan, 76 Roundcroft, Romiley, Stockport, Cheshire SK6 4LS.

Remember, it's your magazine!

6, Glenhurst Grove
Keighley
BD21 4RN
23rd November 1987

Dear Christina,

It is not often that I put pen to paper but the feeling of elation after having successfully completed a project has prompted me so to do.

I have been a console only TI-99er for some time but very recently managed to obtain, second-hand, one of those elusive Extended Basic cartridges. Having got this I decided in my wisdom to have a go at building myself a 32K ram board. Let me say at this juncture that I have never, ever, attempted something like this.

Dave Hewitt's standalone unit with its own address decoding was chosen in preference to those requiring internal connections and the appropriate bits ordered from Maplin. Not having the facilities to produce a proper double sided PCB I had to use Veroboard with multitudinous wire links to produce the desired effect. Anyway, two weeks later I had produced ,a board that nobody in their wildest dreams could say was "neat". The moment of truth came when the board was plugged in. I switched on, did the usual, and then typed "SIZE". Total despair, only the usual 13K was present. Must be a bad connection somewhere I thought, but where in that mess of spaghetti?

*A continuity tester was subsequently built and I proceeded to check the connections. Oh God! Hardly any of the veroboard pins were making good electrical contact with the copper strips even though they were well and truly pushed in. The next few nights were spent laboriously testing every single connection and soldering where appropriate. Several blasphemous evenings later I plugged in. Euphoria! It worked. Then the computer locked up. Another duff connection missed. Eventually all was well and a few experiments with CALL LOADs extracted from back issues of TI*MES confirmed that everything was in order.*

The moral of the story is plain for all to see. Check all your connections as you go along!

The finished product, not yet boxed just in case, may be untidy but it works and I built it myself. The cost? Less than 30 pounds excluding swear box.

*I now have Ex Bas + 32K Ram but no info, apart from that published in TI*MES, regarding CALL LOADs, so if anyone out there can help I would be glad to hear from them. Also, how best to use the extra Ram without Disc Drives (maybe the next project? anyone got a very cheap 3.5" drive surplus to requirements?.)*

Finally, a word about the mag. Great, keep it coming.

Yours sincerely,

Dave Westmoreland

Dave Westmoreland

A visit to the Netherlands

Report on National Dutch TI User Group Show by Clive Scally

*In the 1965 Summer issue of TI*MES (No. 9) you may have seen an article reporting the activities of the Dutch TI-99/4A Users Groups. One thing missing from that report was how a national TI-99ers show is organized and also how a committee of 12 people living all over The Netherlands manage among other things to produce a quality A4 50.4+ page bi-monthly jam packed magazine (TI JDINGEN) and manage the group activities.*

On. Saturday. October 24, a bright warm autumn day Audrey, my two children and I visited Rotterdam in Holland. The big event was the National Dutch TI-99 User Group Meeting and Show.

I was greeted by B. Muller, the Group Chairman, and J. Prins, Treasurer, from the Dutch group known as TI GERBRUIKERSGROEP. They made Audrey and I very welcome. I was informed that the group have a membership of 600 strong, and with so many to cater for (probably the largest in the world) shows were held on National basis at least twice a year in the main cities of Holland.

The Rotterdam Show, held at a business center near Central Station, served two purposes. The First reason was for the working committee to .meet and discuss each of their spheres of activity and exchange ideas etc for the group magazine. This took place at 10am and lasted to midday. At 1pm a TI users fund-raising TI Show was open to all.

The show was well attended, nearly 200 people packed the Beatrix Suite in the business center. Those of you who attend our shows will have an idea how these are organized. Yes, something for everyone including a TI Auction.

Represented was a commercial dealer TI99 Legio of Rijswijk (shades of Parco). The Belgium Texsoft Club, West German TI99 Group, and the committee each manning a table within the spheres of activity they were responsible for.

It is worth reporting the details.

To help me on what was going on I met Vice Chairman Berry Harmsen who is a member of our UK group and a regular visitor to our TI Shows held in the past. He took me round.

As I said the show had all the elements of our last event held in Derby. However with such a large membership the Dutch show had more on offer. The stands manned by the committee were showing off various projects. There was DIY hardware in which the TI99/4a was transformed into a PC, then there was DIY expansion built on a budget, demos of new programs from the Software Library. The sub Group on assembler programs. Other tables were occupied with lots of used TI 99 items and books for sale. The commercial stands were packed with bargain hunters.

An interesting demo was the DIY expansion box complete with two DD disk drives and Myarc cards on the Belgium TEXSOFT Club table. They were showing sophisticated programs actually working. The Texsoft Club is run on a same basis as Peter Brooks TI-Lines Club, where it is of a commercial operation. Over 200 members receive a quarterly A4 40-page Texsoft Club newsletter. This is written mainly in Flemish with some articles in English. Subs are 650 Bel francs Address is Texsoft Club. Kerkeveldstraat 29, 2290 Grobbendonk, Belgium. British TI users are welcome to join.

On the West German table I saw a PC keyboard running TI software. I soon found that it was easy to spend too much time on one stand because there was so much to see. At 3.30 pm there was an auction. This was a popular attraction, also a prize draw for a lucky member who purchased a fl.2.50 (75p) admission ticket.

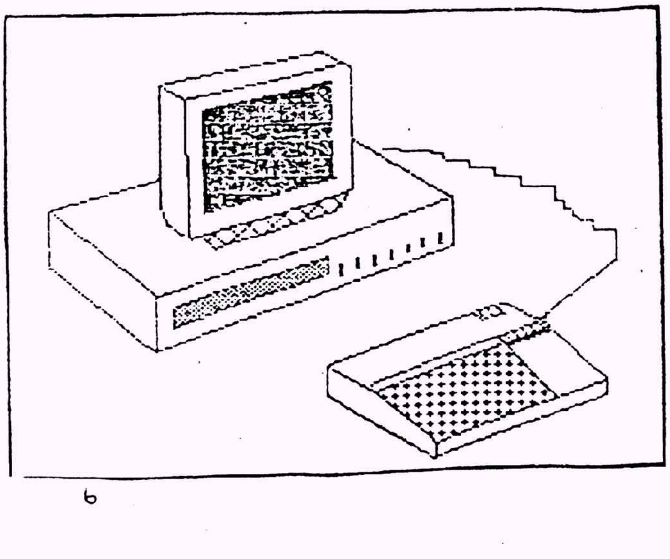
All in all a great day. If you would like to know more about the TI GERBRUIKERS of the Netherlands why not write to Berry Harmsen, vice-voorzitter, TI-99 Gerbruikersgroep, Amstelkade 16, 1078 AC, AMSTERDAM, Netherlands.

If you would like to join the membership, secretary /treasurer (penningmeester) J. Prins, address: Elzepas 52, 6662 XB, ELST. The magazine is well worth a subscription if you can read Dutch.

We were invited guests of Evert Smies, a previous Chairman of the Dutch group. Evert, his wife Annette and two children made Audrey my two children and me most welcome when we stayed for the weekend in Haarlem. As we share the same interests, the visit was even more enjoyable.

We think this is a wonderful idea you should all try and adopt within the TI community. Make friends with our friends across the seas as they wish to do with us. The benefits are mutual but the experience is always memorable and fun. So why not start a friendship today which will last forever.

Clive Scally.
Chairman.



RAMBLES

RAMBLES by Stephen Shaw
January 1988

Greetings and HAPPY NEW YEAR to you. Here we all are, more than four years after TI pulled the plug, with a more extensive range of goodies available than anytime TI were in the market! To help you sort it out, later on a selected list of desirable items for you, whether you want cassettes, disks, or peripherals!

I am delighted to hear from you- let me have your news, views, comments, requests, questions!
10 Alstone Road, STOCKPORT, Cheshire, SK4 5AH

And this is also the address for your User Group Disk Library- with WELL over three hundred disks of goodies. If you have a disk drive send for our current listing- supplied on two disks, so please send two disks and return postage.

And if you would like regular printed news of library additions and revisions, please send me a supply of SAEs. If I don't have your SAE when a list is posted out then you will need to obtain an up to date disk listing to find out what is new! Act quickly- there will probably be a mailing at the end of January!

Quite a few letters this last quarter, for which MANY thanks - even my optician and dentist wrote...

Overseas members please send five pounds sterling, which includes the cost of the disks and airmail postage.

Two of our members have supplied some nice disks- from David Vincent a remarkable graphic adventure CARFAX ABBEY in which you explore a 4x5x5 maze to find a crucifix, vanquish Dracula, and escape with treasures! ExBas with machine code utilities. Also a LOT of code, so regret not possible to transfer to tape. From John Seager, a surprise. There are plenty of (unused) character design utilities around. John has produced (almost) the ultimate- in XB with machine code links for speed. Superb- and also from John is a set of four utilities to help you to manipulate merge format program files.

From a Boston Computer Society publication, some tips from David Taub for ExBas programmers:

1. Is N odd or even?

   Normal:
```
   100 X=(N=2*INT(N/2))+1
```
   will set X=0 if N is even, and X=1 if N is odd.

   Faster:
```
   100 X=N AND 1
```
   does the same for values of N up to 32767.

2. Stop on space and continue on space:
```
   100 FOR P=1 TO 1000
   110 PRINT P
   120 CALL CHECK
   130 NEXT P
   140 GOTO 100
   .......
   900 SUB CHECK
   910 ST=Y :: CALL KEY(0,Y,Z)
   920 IF (Y<>32 OR Y=ST) XOR F=0 THEN F=1 :: GOTO 910
   930 F=0
   940 SUBEND
```
  TRY IT!

........more more more---->

DISK LIBRARY UPDATE

OLDIES BUT GOODIES. 1&2 on one disk. Biorhythm,Factor Foe, Hammurabi, Number Scramble, Word Scramble, Hidden Pairs, Peg Jump, TicTacToe and 3d Tic Tac Toe. Released by TI in 1980 on two cassettes. By 1982 they were 8.25 each. The price WE ask is closer to their real value perhaps! Authors John Plaster and Mary Anne Six (surname or age?). Of more historic value than usable value.

3. Is number N between X and Y?

Suppose we need to hit keys 1 to 5 only?

```
  Normal:
  100 CALL KEY(0,A,B)
  110 IF A<49 OR A>53 THEN 100
  120 PRINT "KEY ";A-48;"WAS PRESSED"
  130 STOP
  Slightly faster is:
  100 CALL KEY(0,A,B)
  110 IF ABS(A-51)>2 THEN 100
  120 PRINT "KEY ";A-48;"PRESSED"
  130 STOP
```

A little explanation!
Keys 1 to 5 have ASCII values of 49,50,51,52,and 53.
If we add 49 to 53 and divide by two we get 51.
This is the number we take away from A in line 110.
Now for the comparison figure: 53 less 49= 4. Divide by two=2.

For any range of numbers you can now find the figure to deduct in the ABS brackets, and also the number to compare to!

C99 is a nice language but not all of our members are interested in it- and it would take a lot of room to do a full tutorial! If there is an interest, I can take a standard C text book and list the amendments for C99, and also detail the method of using c99... but distribution would be on disk.

For Forth we do have two excellent tutorial disks. I also have quite a lot of printed information- all sent on to Geoffrey Coan- to publish it all in TI*MES would take up an excessive amount of room, and to photocopy in small quantities would be very expensive! The information is available- anyone have access to free photocopying?

We have two disks of machine code tutorial, two disks of ExBas tutorial, and one disk of TI Writer hints and tips. The Forth manual, and core source code are on disk- 5 disks and two disks respectively.

There is plenty of info available!

Next I suppose I should look at Turbo Pasc from Texaments...

Fortran I shall leave for someone who uses it- it looks a fiendish language!

:::::::::::::::::::::::::::::::::::

I have received two test versions and a final release version of FUNLWRITER Vn 4.0, which has been extensively rewritten, with more goodies added. The trouble is the configure routine is presently not working intuitively, and there are a couple of bugs as well, one of which can lock your console up with total loss of text!

For the time being I am supplving Version 3.4 (May 87) to anyone just asking for Funlwriter, but if you want Version 4 (on which I can offer NO help whatsoever!) then please ask for it. Added functions include a fast way of changing lower case to caps and vice versa, and a column ruler at the bottom of the text.

:::::::::::::::::::::::::::::::::::::

C99 BUG:

Clint Pulley has advised of a BUG in C.99 Vn 1.3 and 1.31:

Using #include, the file name MUST NOT exceed 6 characters else an error message is generated.

Dave Hewitt tells me that using Funlwriter 3.4 (May 87) he has to load C99 (the file CO that is) from a LOADER option rather than the prespecified menu option, as he otherwise cannot use #include, apparently due to a shortage of available files.

::

RAMBLES

If you are using the DRAGONSLAYER SPELL CHECKER with Funlwriter, you will notice that you MUST load the spell checker from DSK1, and it crashes on exit... well, here's a fix for the crash, from Joe Nuvolini:

Use a disk sector editor to search for hex 04600070 on the Dragonslayer disk and change it to read 0420 0000.

+++++++++++++++++++++++++++++++++

I have mentioned under magazine reviews elsewhere, one way to do a CALL FILES other than to type it in. Another way is to use TEXTLOADER by Curtis Provance- available from your disk library! - which is an XB program you load from disk.

First you create a DV80 file with TI Writer such as:

```
  CALL CLEAR
  DISPLAY AT(12,2):"ONE MOMENT"
  CALL FILES(X)
  NEW
  RUN "DSK.MYDISK.MYFILE"
```

THEN

you place on your disk the Textloader program, containing the name of your DV80 file, calling the amended Textload program LOAD.

XB will autoboot the LOAD, and TEXTLOAD will read and perform the commands in the text file, double quick. VERY NEAT. There are LOTS of other uses for such a clever program, including QUICKLY loading a program listed in a text file into a genuine memory resident runnable program! Its a gem. Try it.

****

Will we see MULTIPLAN on disk, like TI Writer? Nope. Open up your Multiplan module and check out those chips- then look at the size of the disk files they load into your 32k ram and VDP ram. That comes to about 66k. Now squeeze it into 32k? Nope. Stuck with the module until you get your 9640 (Geneve) anyway!

===================

OPERATING SYSTEM BUG

Seems a little late in the day to be discovering system bugs, but here we are with one I have not so far seen reported...

Type and run this:

```
100 CALL KEY(3,A,B) :: PRINT B;B;B :: CALL JOYST(1,X,Y) :: GOTO 100
```

Now hold down a key on the RIGHT side of the keyboard- hold it down continuously. What you get is a lot of +1's. Every key scan treats the continuously held down key as new.

Now hold down a key on the LEFT side of the keyboard... you may get an odd +1, the odds are you will not. You will get a lot of -1's.

Now change the CALL JOYST to scan JOYST(2... and try again to see if there is any difference...

These results are NOT what you might expect, and could well cause problems, especially if you use a CALL KEY to scan for the fire button AND check its status as well. This could explain why I have problems using the unreleased module game LASSO - and also explain why it is so unfinished?

This MAY be a hardware bug, the same as the alpha lock/joystick problem, so easily fixed with a suitable diode and a cut pcb track, or it could be a firmware bug. It is constant in every XB module I have, including Myarc XB.

======================================

I have been asked for a list of usable CALL LOADS. We have published several lists over the years, but they have contained some very odd entries... so, space permitting, watch out for MY list!

FORTH NOTE:

Would you like to input a floating point number from the keyboard? It is not too obvious how to do it and thanks go to George Sloane of the Aloha User Group in Hawaii for this one:

```
 : ENTFP (  --- f ) PAD  1+  DUP 20 EXPECT
     LEN SWAP 1- C! VAL FAC> ;
```

FORTH

Explanation:

The address of PAD+1 is placed on the stack- this is because VAL, perhaps contrary to what the Manual suggests, starts reading the entry at PAD+1. If you just used PAD, you would lose the first character, either the most significant digit, or perhaps a decimal point or a minus sign.

20 EXPECT allows you to enter up to 20 digits... more than you may need but so what!

VAL then converts the string to a floating point number.

FAC> brings the floating point number on to the stack.

If you want an integer number output, you could use FAC->S instead.

=================================

RAMBLES

CALL LOADs...

In response to requests, here is a list I have found useful...

Remember- you must do a CALL INIT before you can use CALL LOAD!

CALL LOAD(-31961,51)::END...return to title screen

CALL PEEK(-28672,A).........IF A=0 then NO speech synth attached.

CALL LOAD(-31888,63,255) then NEW...equivalent of CALL FILES(0).
(frees up another 250 odd bytes of memory from CALL FILES(1))

CALL LOAD(-31888,55,215) then NEW...reverses above and gives you back three drives!

NB:- Attempted disk usage after turning all drives off like this can be fatal!

CALL LOAD(-31931,0)...delete extended basic protection

CALL PEEK(-31863,A)...in a running program, if A=231 then expansion memory is fitted.

CALL LOAD(-31961,149)::END... reset console and load DSK1.LOAD

CALL PEEK(-31952,A,B)...Pointer to starting address of line number table.

CALL PEEK(-31950,A,B)...pointer to end address of line number table.

CALL PEEK(-31954,A,B)...current line being referenced in line number table.

The line number table is composed of four bytes for each line: two bytes hold the line number and two bytes hold the address.

CALL LOAD(-31806,16)...disable quit key.

CALL LOAD(-31878,N)...all sprites over N stop. Can be used to make several sprites start moving simultaneously- THIS LOAD is over-ridden by a subsequent CALL SPRITE or CALL MOTION.

CALL LOAD(-31806,64) will disable all sprite motion until it is reset with CALL LOAD(-31806,0) or (-31806,16).

If you wish to disable sprite motion AND quit key, use CALL LOAD(-31806,80).

CALL LOAD can also be used for SOUND and SPEECH. These entail long articles and have been covered in previous TI*MES.

=======================

SPRITE SAMPLER:

First a program using CALL LOAD(-31878):

```
  100 CALL CLEAR :: CALL INIT :: CALL SCREEN(2) :: R=PI*2/28
  110 FOR I=28 TO 1 STEP -1
  120 CALL SPRITE(#I,46,16,96,128,COS(I*R)*10,SIN(I*R)*10)) :: CALL
LOAD(-31878,0) :: NEXT I
  130 GOTO 130
```

or using -31806:

```
  100 CALL CLEAR :: CALL INIT :: CALL SCREEN(2) :: R=PI*2/28
  110 CALL LOAD(-31806,80)
  120 FOR I=1 TO 28
  130 CALL SPRITE(#I,46,16,96,128,COS(I*R)*10,SIN(I*R)*10)) :: NEXT I
  140 CALL LOAD(-31806,16)
  150 GOTO 150
```

These are both very short listings, so key them in and compare the results!

And nothing to to with CALL LOAD but related to the sprite sample:

```
  100 R=PI*2/28 :: CALL CLEAR :: CALL SCREEN(2)
  110 FOR I=1 TO 28 :: CALL SPRITE(#I,46,16,96,128,COS(I*R)*10,SIN(I*R)*10)
  120 NEXT I
  130 GOTO 110
```

==============================

If you would like to add PEEKV and POKEV to your Extended Basic programming, the following program will help you. It comes from the April 1984 issue of The Smart Programmer and was written by John Brown:

PEEKV-POKEV

```
100 CALL CLEAR :: CALL INIT
:: CALL LOAD(8196,63,232)
110 CALL LOAD(16360,80,79,75
,69,82,32,38,12,80,79,75,69,
86,32,37,164,80,69,69,75,86,
32,37,36)
120 CALL LOAD(9491,100)
130 CALL LOAD(9508,2,224,37,
20,3,0,0,0,2,0,0,100,200,0,3
7,18,4,192,2,1,0,1,4,32,32,1
2,4,32)
140 CALL LOAD(9536,32,24,18,
184,192,32,131,74,2,1,37,0,2
08,160,131,18,9,130,2,34,255
,255,4,32,32,44)
150 CALL LOAD(9562,4,197,209
,34,36,255,9,132,19,21,4,195
,60,224,37,18,200,5,131,76,2
0,5,131,78,200,5)
160 CALL LOAD(-9588,131,80,2,
5,64,0,161,68,2,131,0,1,17,6
,2,5,65,0,161,67,6,196,200,4
,131,76)
170 CALL LOAD(9614,200,5,131
,74,4,192,192,66,5,129,4,32,
32,8,6,2,22,221,4,96,37,254)
180 CALL LOAD(9636,2,224,37,
20,3,0,0,0,4,192,2,1,0,1,200
,1,37,18,4,32,32,12,4,32,32,
24,18,184)
190 CALL LOAD(9664,200,32,12
1,74,37,0,184,32,131,18,37,1
9,2,3,0,2)
200 CALL LOAD(9680,4,192,192
,67,4,32,32,12,4,32,32,24,18
,184,216,224,131,75,37,0,5,1
31,136,3)
210 CALL LOAD(9704,37,18,22,
242,192,32,37,0,2,1,37,2,192
,131,2,34,255,254,4,32,32,36
)
220 CALL LOAD(9726,4,192,216
,0,131,124,2,224,131,224,4,9
6,0,112)
230 CALL LOAD(9740,3,0,0,0,4
,192,2,1,0,1,4,32,32,12,200,
32,131,74,37,18,2,1,0,2,4,32
,32,12,4,32)
240 CALL LOAD(9770,32,24,18,
184,192,32,131,74,208,32,37,
19,4,32,32,48,4,91)
250 CALL LOAD(8194,39,104)
260 END
```

------->------->------> continued ---->---->---->

ARRAYS:

Jim Peterson has produced a table of memory usage by different types of arrays - but has made a fatal error!

He has assumed that an array DIM(2,2) has four elements. It does not- it has 9 elements, unless you use:

OPTION BASE 1

in your program! This error has relegated what I am sure was hard work into trash. Do not forget: arrays begin with element zero unless you specify otherwise!! And if element zero is unused, it still ties up memory!

DISK LIBRARY UPDATE

...

UTIL-16. Four character sets in Source, Object and XB Merge files, together with XB speech tutor programs and M-COPY from Mike Dodd (Version 1.0). M-COPY is one of those programs we should have had back in 1979! NOW available...

Have you noticed that as you get more and more files on a disk it takes longer and longer to find files? And can get pretty noisy? M-COPY can cut that down, as it moves ALL file descriptors to ONE block. Listing 72 files with DM1000 took 32 seconds before M-Copy was used, and only 12 seconds after. That fast file access is permanent! Also saves head wear. For optimum disk access time, first use DM1000 to FILE COPY, to repair segmented files, THEN use M-COPY to put all the descriptors together. WOW! A must for every disk owner who ever has more than 20 files on a disk! M-COPY is compatible with everything (including Geneve) EXCEPT Myarc HARD disk controller and Foundation ram card.

Now, type that in, RUN it, and you can now use in your programs:

```
 CALL LINK("PEEKV",address,variable,variable...)
 CALL LINK("POKEV",address,value,value,value...)
```

ADDRESS must be a VDP RAM address in DECIMAL (0 to 16383)
You may use up to 15 (fifteen) values / variables.

You also have:
CALL LINK("POKER",vdpregisterno,value) to write a value to one of the VDP registers.

Using Peekv and Pokev and VDP registers has been covered in some depth in previous issues of TI*MES. Let me know if you want reprints!

==============================

REPRINTS....

Here is your turn to VOTE again Judging from past performance that means four votes in total, all different, but I do try to find out what you want!
Please cast your vote for a reprint of past information on:

1. Using VDP Registers with PEEKV.
2. Using SOUND with CALL LOADS including continuous play.
3. Using SPEECH with CALL LOADS.
4. Enhanced Basic using the PRK and Stats modules or disk versions of same.
5. Your choice of any other topic!

These are major articles and we will only have room for one per issue. Vote TODAY or miss what you want. Write to

Stephen Shaw, 10 Alstone Road, STOCKPORT, Cheshire, SK4 5AH

If you do not wish to see any information reprinted, let's have a negative vote in as well! You may even outnumber the ayes!

==================

COMPETITION...

Not that many entries at all- what I thought might have been a last minute rush was just a large number of new entries for the disk library! However, one submission clearly deserves a prize!

The competition was launched to try to discover how many UK owners are presently programming - the answer is very few indeed! I had no idea what type of program would win - the result is a little surprising!

We have a number of utilities around to assist with the design of characters. We now have a "super utility" which stands out head and shoulders over the rest! Now into Version 5.1 (available on library disk UTIL 18), this excellent program from John Seager has very many capabilities, including the ability to print both definitions and character shapes to printer (Epson type) and to create merge files of CALL CHAR statements. Lots and lots of things you can do in this very well written program, making it an outright winner. Well done John, carry on programming!

====================

LIBRARY NEWS:

Lots of new items received in November/December, from many different countries (even SWITZERLAND!!!) so be sure to send for an updated disk library catalogue by sending two disks and return post, and enclose also another SAE for a printed list of items received up to the end of January (and another SAE for the end of February and so on).

During the last 5 months, the disk library has been used some 70 odd times by just 28 members. Perhaps we only have 28 members with disk systems? Or are the rest of you obtaining disks elsewhere? If you have a disk system, our disk library offers you a vast wealth of material at very low cost - and its continued growth depends on your usage! If you obtain material from our library elsewhere, a donation to library funds will ensure supplies keep a-coming!

Donations to Freeware authors seem to be a little thin from the UK- if you use a Freeware program, please make a donation to the author and keep the TI programers programming! Donations may be sent c/o me in sterling, but ensure you indicate the amount and authors name/program.

========================

PASCAL

TURBO PASC99.

At press time, I was still awaiting a positive price quotation from Texaments, prior to ordering. However, I have managed to obtain a German copy of the program and have done some benchmark tests as below.

A full review will appear as soon as I have the English version, with English documentation! I understand that Texaments have the source code and are making one or two amendments- these should not alter the benchmarks or general comments too much.

Error handling is excellent, with many syntax errors caught on compilation. Use of defaults when you ask for the impossible mops up many run-time errors, and the range of error messages for run-time is greater than that produced by the Editor-Assembler alone. An error caught on compilation will place you back in the editor with the cursor near the error. Most helpful!

This is NOT Borland's TURBO PASCAL(tm). It is close to "standard" ISO Pascal, amended to make it more friendly for programmers used to the TI99/4A.

ISO Pascal items NOT supported are:
  file, in, packed, record, set, type, with, char, ord, pred, round, sqr,
  succ, trunc, odd, reset, rewrite, dispose, new, pack, unpack.

Items included but not in ISO PASCAL (some replacing items above):
  Block, Module, Relative, Stream, String, Open, Seek, Append, Close, Asc,
  Cursor, Key, Screen, minint, pi, graphics, text, putln, cls, cir, cis, cri,
  crs, csi, csr, len, int, rnd, seg, tan, randomize.

Variations on ISO pascal are:
  Strings are within double quotes "string" instead of 'string'.
  REAL must be specified as 4, 6 or 8 bytes- if you select 8 you have normal ExBas precision.

Some of the portability has gone with these changes of course, but they seem to make the language easier to enter for a TI-user with NO Pascal experience!

In addition, disks containing additional procedures are available- one for Windows, and one for our familiar TI extensions with sprites, sound and so on. If the language is popular, expect more in due course.

NB: The German version I have REQUIRES the Editor Assember MODULE to function.

Books worth looking at:

A CRASH COURSE IN PASCAL (about 8 pounds) by D M Munro. Publisher:Arnold.
ISBN: 0 7131 3553 0

PASCAL USER MANUAL AND REPORT. Jensen and Wirth. SPRINGER-VERLAG.
ISBN: 0-387-96048-1 (specify third edition)

There are also books by Peter Lottrup (COMPUTE!) and Zaks (Sybex) which may be worth a look. Take care: Most Pascal books these days seem to be for Turbo-Pascal (NOT Turbo-Pasc 99 or ISO Pascal).

TURBO PASCAL BENCHMARKS.

These benchmarks were detailed in TI*MES issue 15, and have appeared several times in PERSONAL COMPUTER WORLD magazine.

For turbo pascal, the PASCAL code will first be given, followed by the timing and possibly some notes.

(Remember: You do not need a P Code card to run Turbo Pascal!)

continued on next page.....---> ---> --->>

PASCAL

```
PROGRAM intmath;
VAR t,
    i,x,y : INTEGER;
BEGIN
   writeln("........");
   t := 0;
   FOR t := 1 TO 100 DO
   BEGIN
    x := 0;
    y := 9;
      writeln("start");
      FOR i := 1 TO 1000 DO
      BEGIN
        x := x + (y * y - y) DIV y
      END;
   END;
writeln("---",x);
END.
```

The timing on this program equates to a benchmark of 0.337 seconds for 1000 loops, which compares well with a benchmark of 0.48 seconds for C99.

DISK LIBRARY UPDATE

...

DM99 by Mike Dodd- Version 2.3 now available. This version can be interrupt driven so you can call it up with FCTN 7, or you may use CALL LINKS. It can be loaded before you start your XB programming, or if there is an XB program resident, there is a fast loading CALL LOAD type file. Compatible with almost everything but NOT Myarc XB. A memory resident disk manager for XB use. Full disk. The earlier version, which can be used in TIB using MM or EdAs as well, but can only be used with CALL LINKS, remains available on UTIL-5.

```
PROGRAM realmath;
VAR t,
    i : INTEGER;
    x,y : REAL[4];
BEGIN
  writeln(".........");
  FOR t := 1 TO 5 DO
  BEGIN
   x := 0.0;
   y := 9.9;
   FOR i := 1 TO 1000 DO
   BEGIN
    x := x + (y * y - y) / y;
   END;
  END;
writeln("***END...",x);
END.
```

REALMATH using numbers of 4 bytes, took 8.20 seconds for 1000 loops, compared to 17.7 seconds for plain ordinary TI Basic!

DISK LIBRARY UPDATE

EDAS ONLY-- this disk REQUIRES the use of the EdAs module! WATOR, a type of Life with fishes and sharks (trans;ation from German welcome please!); WILLIWURM can be fast! Eat insects and avoid your tail! and from TI-UK, an unreleased module from 1980: Vat Accounting Module. NB: You use this entirely at your own risk!

```
PROGRAM triglog;
VAR i : INTEGER;
   x,y : REAL[4];
BEGIN
  writeln("........");
   x := 0.0;
   y := 9.9;
   FOR i := 1 TO 400 DO
   BEGIN
    x := x + sin( arctan( cos( ln(y))));
   END;
writeln("***",x);
END.
```

This was SLOOOW and the equivalent of 1000 loops took 625 seconds, which compares badly to 360 seconds in Extended Basic!

DISK LIBRARY UPDATE

HOPSCOTCH... from SWITZERLAND a very well protected disk. Use EdAs Option 3 or other Load and Run loader to load LOAD. Like Q*Bert but here we have a rude kangaroo. You play with your joystick rotated 45 degrees. And here you must land on the floor tiles. LOVELY music!

...

---> continued ----> ---->

PASCAL

```
PROGRAM textscrn;
VAR i : INTEGER;
BEGIN
text;
writeln("START");
    FOR i := 1 TO 1000 DO
     writeln("1234567890qwertyuiop",i);
writeln("***",i);
END.
```

This one took 69 seconds for a 1000 loop, again comparing badly with c99, which took just 38.7 seconds.

```
PROGRAM store;
VAR i : INTEGER;
    f : STREAM[80];
BEGIN
writeln("START");
open(f,"DSK2.TEXT",output);
FOR i := 1 TO 1000 DO
  putln(f,"1234567890qwertyuiop");
close(f);
writeln("******");
END.
```

This took 61.4 seconds, compared to 83 seconds in Myarc XB and 131 seconds in TI XB. Notice how easy disk access can be!

RESULTS:

Benchmarks are not the be all and end all of a language, although an advertiser can give the impression a language is incredibly fast by telling you what it is fast at, and not telling you what it is sloooow at! TURBO PASC 99 seems to be comparable with c99 overall, sometimes better sometimes not.

Personally I found it much easier to use TURBO PASC 99- I even made it write to disk! Look how short the STORE program is!

If you are a TI P Code user you may find these codes a little odd... I can assure you they work EXACTLY as printed here. If YOU have a TI P Code card, why not run comparative tests and let me know the times?

The answers printed out by the math results were:
intmath 8000, realmath 8.9E3 (both same as TI Basic) and
triglog -2.2021E2, compared to -2.20497E2 from TI Basic- this minor inaccuaracy is due to using only 4 bytes for the variable rather than 8, but we could have used 8 if we had wanted such accuracy.

=================

DISK LIBRARY NEWS:

Disk UTIL 16 contains the freeware utility MCOPY - this is especially for disk owners who sometimes have rather a lot of files on a disk! When there are a lot of files, the file descriptor sectors get scattered ALL over the disk, and loading a file can be slow AND wears your drive out faster! SO... why not put all the file descriptors together? THAT is what M COPY does!

For optimum disk use, first repair any segmented files by using FILE COPY, then gather the FDs together by copying the disk with MCOPY. Result? With one disk with 72 files, the time to catalogue using DM1000 was reduced from 32.3 seconds to just 11.4 seconds. More files = greater speed increase! Ask for UTIL16. Just one pound copying fee plus one pound handling fee if you send your own disk, or two pounds fifty plus one pound handling fee if we supply the disk. Write to Stephen today!

MAGAZINE REVIEWS...

Chicago TImes Sept issue:
News of a considerable upgrade to SPAD XIII, the machine code flight simulator by Not Polyoptics, with a RED BARON option, and lots more function keys, including one to let you see your own plane in flight! And an article which talked you through a simple flight on SPAD XIII (reprinted here if room allows).

Chicago TImes Oct issue (74 pages!):
A further exploration of SPAD XIII- which, unlike Microsoft's flight simulator, has trees!

PROBLEM: You have a utility program which you want to autoload on selection of XB, BUT the program insists on a CALL FILES other than the default of 3. You do not want to have to remember to do a CALL FILES(X)::NEW before you load the program. What to do?

Answer- this is XB but requires 32k ram and *may* not work on all set-ups!
First obtain values of M1 and M2 as follows:
Select XB, and with no program resident, type in:

```
  CALL FILES(X) - insert YOUR required value in place of X!
  NEW
  CALL PEEK(-31888,M1,M2)
  PRINT M1;M2
```

Note these down. Now write a LOAD program for your utility disk like this:

```
  100 DISPLAY AT(12,3)ERASE ALL:"ONE MOMENT..."
  110 CALL INIT
  120 CALL LOAD(-31888,M1,M2) ! INSERT THE NOTED VALUES HERE INSTEAD OF M1,M2
  130 ON ERROR 150
  140 OPEN #1:"NOSUCH.DEVICE" ! JUST like this! Moves internal pointers.
  150 ON ERROR 170
  160 CLOSE #1
  170 RUN "DSK1.MYPROGRAM" ! YOUR file name here!
```

Thanks to Rich Klein and Mike Maksimik.

A letter to The Editor from D Hathaway, suggesting that the life of the module port can be extended by switching the console OFF before inserting or removing a module - the theory being that TINY arcing between the contacts on insertion or removal takes place if the power is on, and gently removes micron by micron the protective plating on the contacts.

Reviews of new public domain, fairware and commercial items- DM1000 Vn 3.8 is for the Geneve! A review of Ryte Data's BASIC COMPILER- I shall go back and have another look at it (nope, it still won't do anything for me. Remains a no no!). And a demo of how to make music in assembly language with very helpful comments.

Part two of an article on the various graphics programs.

The Chicago User Group has around 800+ members and welcomes overseas subscriptions. They would like you to send US$24 for 12 months sub to:
Chicago TI Users Group, P O Box 578341 Chicago IL USA 60657
Allow 6 to 10 weeks for a response!

==============================

MICROpendium August 87 issue:
Article on Basic strings by Regena, the start of a new column on C99- starting at the beginning for beginners!, a long 'animated video' written in LOGO2, with music!, and news, views, reviews, programs and ads.

September 87 issue:
Articles on Basic and C99. News that the subscription has to rise due to declining ad sales- and non payment by some former advertisers.
Article on mini memory. Update on Geneve. XB and m/c listings for a new emphasised lower case font. Review of Triton XB.

Subscription:
US$26.50 by seamail, US$40.00 by airmail.
MICROpendium, P O Box 1343, Round Rock, TX USA 78680

SPIRIT OF 99 Oct 87 issue.
Reprints from other user groups- I failed to find any original material in this issue! No overseas subscription details.
November issue: Nothing to comment on.
Everett Wade, 179 Erie Road, COLUMBUS, OH, USA, 43214

========================

TopIcs - LA99er Computer Group. September issue:
Good selection of news and views. News of a disk copier (which I have sent for report elsewhere if it arrives) which will sort out the scattered file directory entries on a fullish disk, so that they all appear together- saving on loading time AND disk drive wear and tear. (= MCOPY)
No subscription details at all.

TopIcs. October issue.
Review of Texloader freeware program. News that MICROpendium advertiser Innovative Programming are leaving trails of tears (beware).

LA 99ers Computer Group,
P O Box 3547   Gardena   CA   USA   90247-7247

TI-LINES Vol 4 Issue 5 Oct 87:

TI WRITER FORMATTER BUG:
An enquiry from John Seager on how to make the TI Writer formatter CENTRE lines CORRECTLY — to confound those TI Writer users for whom .CE has always worked perfectly...
The answer is easy: If you wish to use .CE you MUST first use .LM and set it to a value of 1 or higher.
This is probably a bug in the Formatter program, but .CE does not like it at all if you rely on the default left margin setting, and even gets confused if you try to use .LM 0.

EAST ANGLIA REGION 99ERS.
Scott and JoAnn Copeland
13 Elm Walk, LAKENHEATH, Suffolk, IP27 9QR
October 1987:
Neat one-liner to right adjust money amounts, so that a numeric input of 4 or 4.5 or 4.67 neatly line up:
If the amount is X:

```
 X$=STR$(X+0.001) :: X$=SEG$(X$,1,LEN(X$)-1)
```

then to print out, with say the pence in column 12:

```
 C=12-LEN(X$)+1 :: DISPLAY AT(ROW,C):X$
```

No matter what the amount, the one liner will ensure that when printed as a string there will always be two decimal places. Neat.
Thanks to Joe Quigley.
Their October issue also carried a map of ZORK II and some hints.

TI JDINGEN Uitgave Juni/Aug 1987.
50 pages of Dutch.

BYTEMONGER (Bluegrass 99 Computer Society Inc)
October 1987. 7 A4 pages. Inc 2 pages of DBM reviews reprinted from Central Westchester 99ers, who reprinted it from Chicago TImes...

----

c99

Some short c routines to get you used to using c99 and maybe show how some things are done / some things are used.

These routines are by Donald L Mahler and come from the BOSTON COMPUTER SOCIETY. They have been printed from tested source code.

Remember:
  *s means "pointer to s" while
  &s means "the address of s"

File prf is as follows:

```
/* file dsk1.prf */
/* PRINTF REFS */
#asm
    REF   PRINTF
#endasm
```

Save it to disk!

```
/* 1;C */
#include dsk1.prf
int table[]={3,5,2,9,6};
/* sets up an array */

main()
{
  int i; i=0;
/* first term of array is "0th" */
  while (i<5)
  {
    printf("The address of the %dth \n",i);
    printf("element of table is %u.\n",&table[i]);
/*                */
/* "&table[i]" =  */
/* "address of ith term of array" */
/*                */
/* addresses are unsigned integers */
/* that is why we use 'u' */
/*                */
    printf(".. and the value stored there\n");
    printf("is %d\n",*(&table[i]));
/*                */
    ++i;
/*                */
/*  increment i   */
/*                */
    putchar('\n');
  }
}
```

Type this in using the Assembler editor, save it and then compile it, assemble the result, to say 1/OB. Do NOT select any assembly OPTIONS!
To run using LOAD AND RUN load:
  DSK1.1/OB
  DSK1.PRINTF
  DSK1.CSUP
then start with program name START.

CSUP and PRINTF are supplied with the c99 package.

DISK LIBRARY UPDATE

SPIELE 1. Ex Bas games from Germany: Artillerie, Berzerk, Buddybalon, Circus (actually from COMPUTE!), Desert Flight (from Computer Kontact), Fraggles, Two different froggers, and Indiana Jones. Rules in German but refer to line 1 for an English rem in SOME files, otherwise you should be able to work it out. Pretty good games.

SPIELE 2. Again from Germany, all machine code this time, loader is not supplied- you need EdAs or Funlwriter. OH MUMMY is my favorite as you try to make a path around the hidden trasures. When all the treasures have been discovered you may leave, not before, and DON'T bump into a mummy! There is also MOONFIGHT which comes with C source code, and KARATE- key F? to start fight, and keys QAZ WSX to hit/kick in attack.

DISK LIBRARY UPDATE

PERFECT PUSH... from yet another country! I THINK it is Sweden, but no address is given. Not seen it on commercial sale in the USA. Perfect Push is a DEVILISH hard game in which you push segments of rocket and make them line up under a launch hole. Avoiding some ugly critters which are out to get you. You can trap them- and position the rocket bits- by pushing rocks around. Wow is this hard. The only non-TI program I am aware of to have (a tiny bit) of digital speech (right at the start- SPEECH SYNTH IS NOT REQUIRED!!!). The credit list is as long as you are likely to see! ....

This second routine uses strings- and also requires the file prf defined above!

```
/* 2;c          */
#include dsk1.prf
main()
{
  char *ptr1, *ptr2 ;
/* two character pointers  */
  ptr1="Boston/Computer/Society";
/*              */
/* the address of a string is  */
/* the address of first letter */
/*              */
  ptr2=ptr1 ;
  while (*ptr2)
/*                 */
/*  "*ptr2 !=0" */
/*                 */
  {
    putchar(*ptr2++) ;
  }
/*                 */
/* spell out the string letter */
/* by letter      */
/*                 */
  puts("\n \n Now let's reverse it! \n\n");
/*                 */
/* ptr2 is now address  of last */
/* letter of string!!!          */
/*                 */
  while (--ptr2 >= ptr1)
/*                 */
/* decrease address until back  */
/* at original starting address */
/*                 */
  {
    putchar (*ptr2);
  }
  putchar('\n');
}
```

to transform your program to memory image format, to use with RUN PROGRAM FILE, then load these files, using LOAD AND RUN:

```
 DSK1.C99PFI
 DSK1,2;0
 DSK1.PRINTF
 DSK1.CSUP
 DSK1.C99PFF
 DSK1.FWSAVE
```

and now choose the program named SAVE.

Now you will have a single "PROGRAM" file which you can load in one piece, instead of having to load lots of other files.

And here is another short example of c99 in action. Try it out now!

```
/* 3;c          */
#include dsk1.prf
main()
{
  char x;
  puts("Enter any letter : \n\n");
  x=getchar();
  putchar('\n');
  printf("The upper case form of %c is",x);
  caps(&x);
  putchar(x);
  putchar('\n');
}
caps(ptr)
char *ptr;
{
  if (*ptr <= 'z' & *ptr >= 'a')
  *ptr = *ptr + 'A' - 'a' ;
}
/*          */
/* if letter is lower case then */
/* decrease ascii value by the  */
/* difference between 'A' (65)  */
/* and 'a' (97)       */
```

Now compile,
assemble (remember, NO options!)
and load and run:

```
  DSK1.3/O
  DSK1.PRINTF
  DSK1.CSUP
```

and program start name is START.

DISK LIBRARY UPDATE

GAMES-9:
Much revised APPLE SCRUMP, CHAINLINK Vn 5.0 (card solitaire game, now with ten deals on disk guaranteed solvable), Meltdown, Missile, Nora of Berner from Compute! and WHEEL OF FORTUNE (not unlike Bobs Full House, but Vanda is paid several times more than Bob....).

++++++++++++

GAMES-10: Mainly easy short XB games by Cliff Walters, but with some nice touches worth looking at. Would make good models if you want to try writing machine code games (or Pascal or C99!). Files: Astroattack, Blimp, Bomb, Break (most interesting), busjump, carrace, city attack, demolition, Hi-Lo, mars caves, nimbus, probe, shooting, simon, slither and timetrial. Suitable for younger children or for a more relaxed play.

+++++++++++++

TIPS FROM THE TIGERCUB

#44

Copyright 1987

TIGERCUB SOFTWARE
156 Collingwood Ave.
Columbus, OH 43213

Distributed by Tigercub Software to TI-99/4A Users Groups for promotional purposes and in exchange for their newsletters. May be reprinted by non-profit users groups, with credit to Tigercub Software.

Over 130 original programs in Basic and Extended Basic, available on cassette or disk, now reduced to just $1.00 each, plus $1.50 per order for cassette or disk and PP&M. Cassette programs will not be available after my present stock of blanks is exhausted.
Descriptive catalogs, while they last, $1.00 which is deductable from your first order.

Tigercub Full Disk Collections, reduced to $5 postpaid. Each of these contains either 5 or 6 of my regular catalog programs, and the remaining disk space has been filled with some of the best public domain programs of the same category. I am NOT selling public domain programs - they are a free bonus!
TIGERCUB'S BEST, PROGRAMMING TUTOR, PROGRAMMER'S UTILITIES, BRAIN GAMES, BRAIN TEASERS, BRAIN BUSTERS!, MANEUVERING GAMES, ACTION REFLEX AND CONCENTRATION, TWO-PLAYER GAMES, KID'S GAMES, MORE GAMES, WORD GAMES, ELEMENTARY MATH, MIDDLE/HIGH SCHOOL MATH, VOCABULARY AND READING, MUSICAL EDUCATION, KALEIDOSCOPES AND DISPLAYS

NUTS & BOLTS (No. 1), a full disk of 100 Extended Basic utility subprograms in merge format, ready to merge into your own programs. Plus the Tigercub Menuloader, a tutorial on using subprograms, and 5 pages of documentation with an example of the use of each subprogram. Reduced to $15.00 postpaid.
NUTS & BOLTS NO. 2, another full disk of 108 utility subprograms in merge format, all new and fully compatible with the last, and with 10 pages of documentation and examples. Also $15 postpaid.

```
****************************
* NUTS & BOLTS #3 is now   *
* ready, another full disk *
* of 140 new merge-format  *
* utility subprograms, all *
* compatible with the pre- *
* vious. With 11 pages of  *
* documentation, $15 ppd.  *
****************************
```

TIPS FROM THE TIGERCUB, a full disk containing the complete contents of this newsletter Nos. 1 through 14, 50 original programs and files, reduced to $10 ppd.
TIPS FROM THE TIGERCUB VOL. 2, another diskfull, complete contents of Nos. 15 through 24, over 60 files and programs, also just $10
TIPS FROM THE TIGERCUB VOL. 3, another 62 programs, tips and routines from Nos. 25 through 32, $10 postpaid.
TIPS FROM THE TIGERCUB VOL. 4, another 48 programs and files from issues 33 through 41, also $10 postpaid.

Thanks to Steve Chapman and Bill Wallbank of Stone & Webster Engineering Corp. TIUG for this one. If V=21 you are in Extended Basic, otherwise you are in Basic. I am not sure it will work with all consoles and modules. -

```
100 RANDOMIZE (0)
110 V=INT(RND*100)
```

How can you input a blank (CHR$ 32) with ACCEPT AT? As far as I know, you can't. With LINPUT, just hit the space bar, and with INPUT, type " ". But with ACCEPT AT the space bar gives a null string and " " gives " " ! However, you can code around it -

```
X$=CHR$(34)&CHR$(32)&CHR$(32
):: ACCEPT AT(1,1):T$ :: IF
T$=X$ THEN T$=CHR$(32)
```

And, to clear up the puzzling behavior of the "quote marks" -

```
100 CALL CHARPAT(34,CH$):: C
ALL CHAR(35,CH$)!written by
Jim Peterson
110 DISPLAY AT(1,7)ERASE ALL
:"THE # PUZZLE":" You can't
enter PRINT # or PRINT ### -
 the computer   demands an
even number of #."
120 DISPLAY AT(5,1):"1 PRINT
 ## !prints a null string (n
othing)":"2 PRINT ### !print
s *"
130 DISPLAY AT(8,1):"3 PRINT
 #### !prints #":"4 PRINT ##
*## !crashes as  STRING-NUM
BER MISMATCH"
140 DISPLAY AT(11,1):"5 PRIN
T ##*## !crashes as  SYNTAX
 ERROR"
150 DISPLAY AT(13,1):"6 PRIN
T ###### !prints ##":"7 PRIN
T ###*### !prints ###":"8 PR
INT ####*#### !print ####"
160 DISPLAY AT(16,1):"9 PRIN
T ######## !prints ###":"10
PRINT ####*#### !crashes as
STRING-NUMBER MISMATCH"
170 DISPLAY AT(19,1):"11 PRI
NT ####**#### !crashes as SY
NTAX ERROR":"12 PRINT ######
#### !####"
180 DISPLAY AT(22,1):"13 PRI
NT #####*##### !#####":"14 P
RINT #####**#####!###*##"
190 DISPLAY AT(24,1):"TRY IT
! LINE NO.(1-14)?" :: ACCEPT
 AT(24,25)VALIDATE(DIGIT)SIZ
E(2)BEEP:LN :: IF LN<1 OR LN
>14 THEN 190
200 CALL CLEAR :: ON LN GOSU
B 230,240,250,260,280,290,30
0,310,320,330,340,350,360,37
0
210 PRINT :;:;:"Press any ke
y"
220 CALL KEY(0,K,S):: IF S=0
 THEN 220 ELSE 110
230 PRINT "" :: RETURN
240 PRINT "*" :: RETURN
250 PRINT """" :: RETURN
260 PRINT ""*"" !crashes as
STRING-NUMBER MISMATCH - the
 * is misinterpreted as a mu
ltiplier!Same with +,-,/
270 !with anything else, inc
luding numerals, crashes as
SYNTAX ERROR - but inserts a
 space before the character!
280 PRINT ""**"" :: !crashes
290 PRINT """""" :: RETURN
300 PRINT """*""" :: RETURN
310 PRINT """**""" :: RETURN
320 PRINT """""""" :: RETURN
330 PRINT """"*"""" !crash
340 PRINT """"**"""" !crash
350 PRINT """""""""" :: RETU
RN
360 PRINT """""*""""" :: RET
URN
370 PRINT """""**""""" :: RE
TURN
```

The method of closing an "ajar" file, described in Tips #28, doesn't always work, but this one seems to be reliable -

```
100 ON ERROR 500 :: OPEN #1:
"DSK1.TEST" :: INPUT #1:A$ :
: PRINT A$ :: STOP
500 ON ERROR 510 :: CLOSE #1
510 INPUT "CHECK DISK AND DR
IVE, PRESS ANY KEY":DUMMY$ :
: RETURN 100
```

This one is just for the fun of it - it uses the contents of computer memory to create designs -

```
100 DISPLAY AT(3,10)ERASE AL
L:"COLORPEEK": :TAB(7);"by J
im Peterson": : :" Watch the
 computer's memory": :"displ
ayed in color."
110 DISPLAY AT(12,1):"Choose
": :"(1) plain colors": :"(2
) bars & checks": :"(3) patt
erns" :: ACCEPT AT(12,8)VALI
```

```
DATE("123")SIZE(1):Q :: CALL
 CLEAR :: IF Q=1 THEN 170
120 DISPLAY AT(12,5):"wait,
please" :: IF Q=3 THEN 140
130 FOR CH=32 TO 143 :: CALL
 CHAR(CH,RPT$("F0",8)):: NEX
T CH :: GOTO 160
140 RANDOMIZE :: FOR CH=32 T
O 88 :: FOR J=1 TO 4 :: X$=S
EG$("0018243C425A667E8199A5B
DC3DBE7FF",INT(16*RND+1)*2-1
,2):: B$=B$&X$ :: C$=X$&C$ :
: NEXT J :: CALL CHAR(CH,B$&
C$)
150 CALL CHAR(CH+55,B$&C$)::
 B$,C$="" :: NEXT CH
160 FOR SET=0 TO 14 :: CALL
COLOR(SET,SET+1,16-SET):: NE
XT SET :: CALL SCREEN(2):: G
OTO 180
170 FOR SET=0 TO 14 :: CALL
COLOR(SET,SET+2,SET+2):: NEX
T SET :: CALL SCREEN(16)
180 FOR J=-1 TO -2000 STEP -
1 :: CALL PEEK(J,A):: A=A-(A
<33)*(A+32):: A=A+(A>143)*(A
/2):: R=R+1+(R=24)*24 :: CAL
L HCHAR(R,1,A,32)
190 C=C+1+(C=32)*32 :: CALL
VCHAR(1,C,A,24):: NEXT J ::
GOTO 100
```

Unlike most of the number games played against the computer, you can win this one -

```
100 CALL CLEAR :: CALL SCREE
N(16):: DISPLAY AT(3,8):"THE
 '37' GAME" !by Jim Peterson
110 DISPLAY AT(5,1):" We wil
l take turns picking":"a num
ber from 1 to 5, but":"not t
he number that was just":"pi
cked."
120 DISPLAY AT(10,1):" The n
umbers we pick will be":"add
ed to the total count."
130 DISPLAY AT(13,1):" Whoev
er reaches 37 is the":"winne
r, but if you go over":"37 y
ou lose."
140 CALL SHOW(20,1,"Press an
y key to start")
150 CALL KEY(0,K,S):: IF S=0
 THEN 150
160 DATA 4,11,17,24,30,37
170 DATA 262,330,392,523,523
180 DATA 1047,784,659,523,52
3
190 C,P=0 :: CALL CLEAR :: C
ALL MAGNIFY(2):: R=10 :: FOR
 J=1 TO 5 :: CALL SPRITE(#J,
48+J,5,R,10):: R=R+30 :: NEX
T J
200 CALL SHOW(24,1,"(Y)ou or
 (C)omputer first?"):: ACCEP
T AT(24,28)VALIDATE("YC")SIZ
E(1):Q$ :: DISPLAY AT(24,1):
""
210 IF Q$="C" THEN CALL SHOW
(22,8,"I pick 4"):: CALL COL
OR(#4,1):: P=4 :: C=4 :: CAL
L SHOW(3,10,"COUNT=4")
220 CALL SHOW(20,8,"Pick you
r number"):: ACCEPT AT(20,26
)VALIDATE("12345"):N :: IF N
=P THEN 220
230 IF P>0 THEN CALL COLOR(#
P,5)
240 CALL COLOR(#N,1):: P=N :
: C=C+N :: CALL SHOW(3,10,"C
OUNT= "&STR$(C)):: IF C=37 T
HEN 320 ELSE IF C>37 THEN 34
0
250 RESTORE 160
260 READ X :: IF C<X THEN B=
X-C ELSE IF X<37 THEN 260
270 CALL SHOW(22,8,"I'm thin
king..."):: FOR Y=1 TO 700 :
: NEXT Y
280 IF B>5 AND B/2=INT(B/2)T
HEN B=B/2
290 IF B>5 OR B=P THEN B=1-(
P=1)
300 CALL SHOW(22,8,"I pick "
&STR$(B)):: CALL COLOR(#P,5)
:: CALL COLOR(#B,1):: P=B ::
 C=C+B :: CALL SHOW(3,10,"CO
UNT= "&STR$(C))
310 IF C=37 THEN 340 ELSE IF
 C>37 THEN 320 ELSE 220
320 RESTORE 170 :: FOR J=1 T
O 5 :: READ F :: CALL SOUND(
100,F,5,F*1.03,5):: NEXT J :
: CALL SHOW(12,8,"YOU WIN!")
330 CALL SHOW(15,8,"Play aga
in? (Y/N)"):: ACCEPT AT(15,2
6)VALIDATE("YN"):Q$ :: IF Q$
="N" THEN STOP ELSE 190
340 RESTORE 180 :: FOR J=1 T
O 5 :: READ F :: CALL SOUND(
300,30000,30,30000,30,F,30,-
4,5):: NEXT J :: CALL SHOW(1
2,8,"YOU LOSE!"):: GOTO 330
350 SUB SHOW(R,C,T$):: FOR J
=1 TO 10 :: DISPLAY AT(R,C):
" " :: DISPLAY AT(R,C):T$ ::
 NEXT J :: SUBEND
```

A couple more peculiarities of the computer -

```
100 DISPLAY AT(3,8)ERASE ALL
:"POS PUZZLE #1": :"       f
rom Tigercub"
110 DISPLAY AT(9,1):"Why doe
s the computer say":"that X=
1 if you answer the":"prompt
 with the Enter key":"(null-
string) ?"
120 DISPLAY AT(14,1):"110 IN
PUT M$"
130 DISPLAY AT(15,1):"120 X=
POS(""TESTING"",M$,1)::":"PR
INT X :: GOTO 100"
140 !POS PUZZLE #1 - why doe
s the computer say that X=1
if you answer the prompt wit
h Enter (null-string) ?
 - Jim Peterson
150 INPUT M$
160 X=POS("TESTING",M$,1)::
PRINT X :: GOTO 140
```

And -

```
100 DISPLAY AT(3,8)ERASE ALL
:"POS PUZZLE #2": :"       f
rom Tigercub"
110 DISPLAY AT(7,1):"Why doe
s the computer say":"that th
e first position of":"null-s
tring is at whatever":"posit
ion it is told to start":"se
arch at?"
120 DISPLAY AT(13,1):"100 M$
="""""
130 DISPLAY AT(14,1):"110 DI
SPLAY AT(20,1):""POS?"" :: A
CCEPT AT(20,6):P"
140 DISPLAY AT(16,1):"120 X=
POS(""TESTING"",M$,P):: DISP
LAY AT(22,1):""X="";X :: GOT
O 110"
150 M$=""
160 DISPLAY AT(21,1):"POS?"
:: ACCEPT AT(21,6):P
170 X=POS("TESTING",M$,P)::
DISPLAY AT(23,1):"X=";X :: G
OTO 160
```

Here is an improvement to the PRINTSPEAKER in Tips #40 - in lines 130 and 160, change the CHR$(1)&"1" to CHR$(3)&"255" . This will avoid problems if the program being converted opens FILE #1.

Irwin Hott informs me that assembly routines which have been imbedded into XBasic programs, using ALSAVE or SYSTEX, can be saved to cassette and reloaded. This could be very useful for those who have a stand-alone or "matchbox" 32k.

And, a mini-game for you to have fun with or improve on -

```
1 !      2-LINE GAME
       by Jim Peterson
- use S&D keys to paint the
white line on the highway
2 !if it is too easy, change
the 6 in A$=RPT$(CHR$(143),6
) to 5 and the 5 in C>T+5 to
 4
100 CALL CLEAR :: A$=RPT$(CH
R$(143),6):: CALL COLOR(14,2
,2,2,16,16):: CALL SCREEN(4)
:: T=11 :: C=14 :: CALL HCHA
R(22,C+2,42):: RANDOMIZE
110 T=T+INT(3*RND-1)+(T=21)-
(T=1):: PRINT TAB(T);A$ :: C
ALL KEY(3,K,S):: C=C+(K=83)-
(K=68):: CALL HCHAR(22,C+2,4
2):: IF C<T OR C>T+5 THEN ST
OP ELSE 110
```

And finally, one of the best examples of compact programming I have ever seen -

```
1 !JOHN WITTE'S 3-LINE VERSI
ON OF JOHN WILLFORTH'S WAVE
POWER - PUBLISHED IN GREATER
 OMAHA UG NEWSLETTER
100 CALL CLEAR :: A$(1)="ABC
DEFGFEDCBA" :: FOR I=1 TO 7
:: CALL CHAR(72-I,RPT$("0",2
*I-2)&"FFFF",47,"30303EFF7F3
E1E04"):: A$(I+1)=SEG$(A$(I)
,2,12)&SEG$(A$(I),2,1):: NEX
T I
110 CALL SPRITE(#5,47,2,180,
180,-23,0,#6,47,2,80,100,-23
,0):: CALL MAGNIFY(2)
120 FOR I=1 TO 12 :: PRINT A
$(I+(I>7)*2*(I-7))&A$(1+I+(I
>6)*2*(I-6)):: NEXT I :: GOT
O 120
```

Memory full

Jim Peterson

ARRAYS AND SORTS

by Jim Peterson

The concept of arrays, and especially of multi-dimensional arrays, is very difficult for many people to grasp. The following is the best explanation that I know of.

A variable name is a box in which you store something. When you write A$="X" you are telling the computer to "go to the box labeled A$ and put the character "X" in it". Or, more acurately, "go to the box labeled A$, throw away anything you find in it, and put "X" in it."

A simple array such as A$(3) is a row, labeled A$, of at least 3 boxes, labeled (1), (2), (3), and maybe more. When you tell the computer that A$(3)="X" you are again telling it to go to the row of boxes labeled A$, find the box labeled (3), and put "X" in it.

A 2-dimensional array such as A$(3,3) is a row, labeled A$, of at least 3 filing cabinets, labeled (1, and (2, and (3, and each having at least 3 drawers labeled 1) and 2) and 3). So, you can use A$(3,3)="X" to tell the computer to find the row of filing cabinets labeled A$, go to the one labeled (3, and open the drawer labeled 3) and put "X" in it.

And in a 3-dimensional array, A$(3,3,3)="X" tells the computer to find the A$ row of cabinets, find the one labeled (3 and find the drawer labeled ,3, and find the folder in that drawer labeled 3) and put.....

Finally, you can write A$(2,2,2,2,2,2,2)="X" to tell the computer to find row A$; cabinet (2 ; drawer ,2 ; folder ,2 ; paper 2, in the folder; line 2, on the paper; word 2, on the line; and letter 2) of the word!

Yes, TI Extended Basic can handle 7-dimensional arrays, but it is not very practical. Try running this - 100 DIM A(3,3,3,3,3,3,3) - and you will get MEMORY FULL IN LINE 100. Arrays with several dimensions are very wasteful of memory. I don't think I have ever seen a program that used more than a 4-dimensional array, and very rarely more than 3 dimensions.

Now then - A$(J)="X" means "go to the box labeled "J", find the number in it, then go to the row of boxes labeled A$ and find the box in that row which is labeled with that number....."

And even something as horrible-looking as A$(Y(J),Z(A,B))="X" just tells the computer to -
1. go to box J and find the number in it;
2. go to row of boxes Y and find the number in box number J of that row;
3. go to box A and find the number in it;
4.go to box B and find the number in it;
5. go to the row of filing cabinets labeled Z, find the one labeled with number A, open the drawer labeled with number B and find the number in it;
6. go to the row of filing cabinets labeled A$, find the one labeled with the number you found in Y(J), open the drawer labeled with the number you found in Z(A,B) and;
7. put the "X" in it!

Simple, isn't it?

Remember that, in a multi-dimensional array, only the last dimension holds the value; the others are just pointers to its location. A$(2,3)=A$(3,3) throws out whatever is in the 3rd drawer of the 2nd cabinet of the A$ row, and replaces it with whatever is in the 3rd drawer of the 3rd cabinet of that row, but the contents of the 3rd drawer of the 3rd cabinet are unchanged.

Also remember that box X or box X(1) or cabinet drawer X(1,1) or whatever, contain a 0 until you put something else in; box X$ or X$(1) or drawer X$(1,1) contain nothing at all until you put a string value into them. When you put something in the box, you throw away whatever was previously in the box. And to empty a box without putting anything in, you put a 0 in a numeric box or "" into a string box.

Enough, on that subject. Now, when you have all your data crammed into an array, the next thing you will probably need to do is to sort it into alphabetic or numeric sequence.

Sorting is one of the hardest jobs that you can give to a computer, and one of the things that a computer is the slowest at doing. Your TI can figure your bank balance in a split second, but might take half an hour to sort your mailing list.

Here's why. You can sort a bridge hand of 13 cards into sequence in 13 moves or less, by simply pulling out each card and slipping it back into its proper place. But, suppose those 13 cards were in 13 boxes, and you had to sort them without removing them from the boxes, except that you could hold one card in your hand? Even if you could figure out the best way, it would take you far more than 13 moves.

That is the problem that the computer has. You have just learned that the computer stores all those values in labeled boxes, or file drawers, and therefore must sort them by shuffling them from one box to another, emptying a box to shuffle into by holding one value in a temporary box while its value is compared with the others to find its proper place.

Of course, you could just set up a new row of empty boxes, and then search through the old boxes for the lowest value and move that to the first box in your new row, etc. - but that would double the amount of memory that the job would require. This would be no problem for a small array, but the computer can sort small arrays fast enough by the one-row method - it is the largest arrays that are too slow by the one-row method and would need too much memory by the two-row method.

Many ingenious routines have been written to accomplish these one-row sorts. I have written a program called "Sort Watcher" which enables you to actually watch various sorts taking place on the screen. It will also tell you the number of swaps and comparisons that were made.

This program demonstrates that the time required for a sort increases greatly as the size of the array increases. Sorting an array of 20 does not take just twice as long as sorting an array of 10 - it may take 4 times as long. For this reason, some of the faster and more complex sorting routines divide an array into smaller segments to be individually sorted and then merged.

After an array has been sorted, my program will also let you change any value in any part of the array, and then let you watch the array being resorted. From this, you will learn that a sorting routine which is very fast for a completely random array may be very slow for an array which is already almost in sequence!

In fact, to add just one additional value to a sorted array, the fastest method is the simple "shoehorn" - just set up an empty box at the end of the row, and move each value down by one box until you come to the proper place for the new value.

A sorting routine can be either numeric or alphabetic depending on whether the variable names used are numeric or string. A numeric sort will be in strict numeric sequence and an alphabetic sort will be in ASCII sequence. That means that if all your strings are composed of upper case alphabetic characters, or all are lower case alphabetic characters, you will get an alphabetic sort - but if they are mixed, all of the upper case strings will come before any of the lower case strings, because the upper case ASCIIs are 65-90 and the lower case are 97-122. And if you have lower case words with capitalized initial letters...!

For the same reason, if you perform an alphabet sort of strings containing numeric digits, you will not get a numeric sequence - 10000 will come before 2 because 1 has a lower ASCII code than 2. It would be extremely difficult to devise a sorting routine which could sort numeric digits numerically within strings. However, if all the numbers are the same length, such as ZIP codes, the ASCII sort will be numeric.

Sorting a multi-dimension array becomes a very complex task. If you swap values around without also swapping all the related values, you will end up with complete garbage. Swapping all the related values takes time, and a dimensioned temporary variable name is also required.

Another way around this is to combine the data from an array into simple strings, or set it up originally as simple strings, and then perform a simple sort based on a specified segment of the string. For instance, you could use TI-Writer with tab settings to create a mailing list having first name at tab 1, second name at tab 15, address at tab 25, city at tab 45, state at tab 55 and zip code at tab 65. Then you could sort into last-name alphabetic sequence by sorting on SEG$(M$(J),10,255), or into zip code sequence by sorting on VAL(SEG$(M$(J),70,5)).

When using TI-Writer to set up such a file, be very sure to save it by PF with the C option, not by SF, and don't leave any blank lines at the end or elsewhere.

Alternatively, elements of data can be crammed into a string separated by control codes, and sorted by position of the code -
FOR J=1 TO 5 :: READ A$ :: M$=M$&CHR$(J)&A$ :: NEXT J
and then sort on element X by -
SEG$(M$(J),POS(M$(J),CHR$(X),1),255)

DISK LIBRARY UPDATE

+++++++++++++++++
R A G M A C - Macro Assembler. Now into Version 6.
Changes: Bugs fixed:
Macros with names beginning with R, read from the macro file were lost when the R option entered the register symbols into the dictionary.

NEW FEATURES:
Characters # and $ are accepted as the 2nd and following characters in symbols

DORG is partially implemented: the OPERAND of the DORG directive must be an absolute expression.

New option: C for compressed object code

New Prompt: ID/DATE allows both the listing and object file to be identified and matched up.

New directive OBJREC allows writing arbitrary records into the object file

Two new system macros:
$S4 contains the ID/DATE entered
$S5 contains the source file name

The documentation has been updated to reflact these changes.

+++++++++++++++++

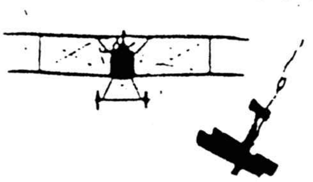
+++++++++++++++++
R A G LINKER - Now into Version 2.
Creates image format files from DF80 files.
And a little bit more.
Associated program new in, LINKER LIBRARIAN by Tom Bentley.
Should be considered essential if you use the linker. Two disks for both!

+++++++++++++++++

RLE/23: 18 files: Doug Quin, Female Face #15, Mad Hatter, Marlene, Pyramid, Rogue and Trident (Adventure characters!), Sad Duck, Snow White, Star Trek 15, Sunset, TerraHawk, TI99Console, Tiger #2, Weekend, and two Winnie the Pooh!

+++++++++++

Great Software From



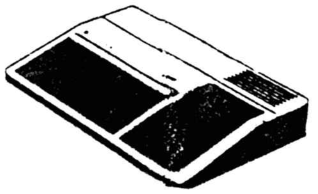
not POLYOPTICS

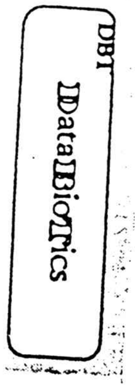
SPAD XIII

Aerial Combat Game!

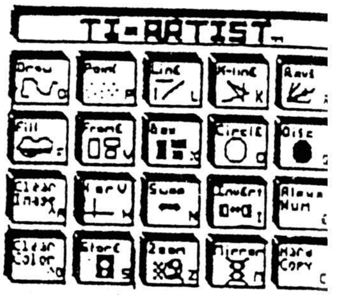
100% Assembly Language!

"I think this is the best flight simulation available for the TI." — Micropend





FONT WRITER II



micro pal®

BRAND COMPUTER PRODUCTS

SPOIL YOURSELF

NAVARONE

FEATURE
AVAILABLE PRODUCTS!

The following pages detail some of the items you may purchase for your TI99/4A, together with prices and addresses.

We have received no payment for this publicity which is brought to you as a service to our members.

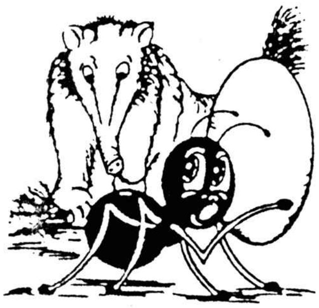
Please advise Stephen Shaw of any poor (or good) service you receive.

DATA BASE MANAGER

- Design Your Own Screens
- Sort On Up To Six Keys
- Great For Business Or Hon

DATABASE MANAGEMENT SYSTEM. rone's latest improved data manager offers plu tridge ease of use. It allows you to devel organize your files, design your own input customize data entry and index information b ple keys. A powerful sort utility will organiz disk files in any sequence by up to six keys. I versatile reporting function. Requires disk syste

31858 Cartridge



ANTEATER. Some people say it's like Dig-Dug . . . with lots of action and just as much fun! The object is to retrieve a pile of sugar cubes from the surface and carry them safely to the nest while avoiding the deadly Anteaters. You can also lay deadly exploding eggs to slow them down.

42104 Cartridge

Wordwriter Plus

- Includes a Printer Hookup

WORDWRITER PLUS. Same great features as described above, but with an exciting PLUS . . . this cartridge includes a cable hookup for any standard parallel printer, including Star, Epson, Axiom, and many more. That's right — you can attach your printer directly to the WORDWRITER PLUS cartridge using the included cable and start printing your word processing documents immediately — no other face required! Includes printer cable. From BioTics.

65564 Cartridge & Cable

CONSOLE WRITER

Plug-in Letter Writer

- Super Easy To Use
- No Disk Drive Needed
- No Memory Expansion Needed

CONSOLE WRITER. This word processor is simple enough for a child to use, and is great for unexpanded systems that have only a printer hooked up to the peripheral port. Features include a full screen text editor that lets you insert and delete both characters and complete lines. This program is perfect for many home word processing needs such as writing letters, notes, homework assignments, etc. A breeze to learn and use! Requires only TI 99/4A console and a properly connected printer. Cassette or disk storage is optional.

32329 Cartridge

BARRAGE. How long can you defend your position and protect your arsenal from enemy onslaught? Tv players position their laser-cannon sights, lock onto the target, and then fire to blast the enemy. It requires all your instincts, reflexes and agility to defend your base!

68645 Cartridge

MYARC

TENEX Computer Expres

CorComp Inc.

PRODUCT REVIEW

SPOIL YOURSELF

In TI-LINES Sept issue mention was made of a firm supplying binders suitable for both TI-LINES and TI*MES. I have previously reported that the binders supplied by the Post Office Philatelic Bureau were suitable, but the price quoted in TI-LINES was much cheaper. And wrong. Here are correct details.

TI-LINES and TI*MES are nominally A5 trimmed size.They require a binder which can take a page size of 8.75" by 6".

Supplier: BINDERS, 78 Whalley Road, Wilpshire, BLACKBURN, Lancs, BB1 9LF

Type: CORDEX.
Navy blue binding. Magazines held by 13 spring loaded cords.

Page size:	Spine width:	Price inc vat:
8 3\8" x 6"	1.75"	2.95
11 x 8.5"	2.75"	2.95
11.75 x 8.25"	2 1\8"	2.95

Type: WIREX.
Black binding.
Magazines held by steel wires- 12 supplied, extra wires 2p each.
Each wire is 1.4mm wide- allow for this in calculating spine width required!

Page Size:	Spine Widths:	Prince inc vat:
8.75 x 6"	2" OR 3.5"	2.95
10 x 8"	2" OR 3.5"	2.95
11 x 8.5"	2" OR 3.5"	2.95
11.75 x 8.5"	2" OR 3.5"	2.95
12 x 9"	2" OR 3.5"	2.95
12 x 9"	2" only	2.95

POSTAGE EXTRA:
1 binder:1.50 2 or 3 binders: 2.00 4,5 or 6 binders: 3.00
More than 6 binders: 3.50

Gold blocked adhesive labels:
Year labels: 10p each.
Title: 1 to 4 lines. 1 label 50p.
2-5 labels 35p each. 6 or more 25p each (same title).

Binders in other sizes made to order (and quotation) for miniRum 50 copies.

The service is prompt and the binders very acceptable. A 2" spine width will take 6 issues of TI*MES or 12 issues of TI-LINES.

The Wirex binder is MANUFACTURED by Modern Bookbinders Ltd, Chadwick St, Blackburn, but send your orders to the supplier address above!

GOODIES

On the following pages you will find details of some goodies STILL available for your computer (and yourself). Suppliers are in the USA or Canada so first a note on personal importing.

Since I bought my TI way way back I have been personally importing, with very few problems, and have saved myself a GREAT deal of money by doing so. Here is how it goes:

If possible quote a credit card - Barclaycard etc should be referred to only as VISA while you should refer to Access only as MasterCard. You are billed in $$$ which is converted into sterling by your credit card company when they receive the voucher - so if the exchange rate moves between your ordering and their debitting you, you may pay more or less than you anticipated.

----> more

GOODIES. Continued.--->

The advantage of using a credit card is that if goods are not available, it costs nothing. If the price differs from what you think it is, your credit card account can be charged with a different amount.

You MUST advise the expiry date of your credit card! and the delivery address should be the same as the address on your credit card statement (PLUS Country!).

If credit card facilities are not available, you will need to purchase a US DOLLAR INTERNATIONAL MONEY ORDER, available on demand from Barclays and Lloyds, at a charge of THREE POUNDS. Available to customers AND non-customers. For Canada, life is more expensive as you must buy a BANK DRAFT- costs vary so check with your bank- they do cost more though.

If you need a fast response ask for INSURED AIRmail. Remember that packing can weigh quite a bit! When Texaments sent me my Myarc ram card the ACTUAL POSTAGE alone was US$20. They charged me US$35 for the post, packing and insurance.

Although ships can cross the Atlantic in a week or thereabouts, sea mail actually takes anything up to 8 weeks! (Turtle post?).

Upon delivery, you MAY be required to pay CASH to the postman before he hands over the goodies. This covers import duty and VAT and of course a Post Office handling charge. Allow about 23% of invoice value.

Small orders (say under £25) often go through uncharged. The larger the amount the more likely you will be required to pay the duty!,On VERY large amounts the Customs MAY write to you and ask for a cheque before they clear the parcel. They may also request more documentation, so keep a copy of your order.

Now a note on the following data. Products listed have been SELECTED as being representative, and of interest. Many other products are available.

Suppliers listed are thought to be reliable but no responsibilty is accepted by the writer or the club should any problems occur. In the unlikely event of difficulties however, the writer will give whatever assistance is possible.

Products are often listed in abbreviated format, so we can list more!
The following initials in SQUARE brackets [] mean:
Cartridge/ Joystick required / Speech synth required /
Disk system required / Ram expansion required / Myarc ram expansion required
Printer required / Tape program

Prices are quoted first exclusive of post and packing, and then in round brackets () an inclusive price is SUGGESTED. If the amount in round brackets is underlined then the inclusive price is advertised and therefore fairly firm.

All prices are in DOLLARS. I wrote to every major TI supplier in September, and these details are based on the replies of those who bothered to reply. A request for updated information to December was unmet at pressdate. Prices and product information are as at October 1987.

In general, supply by airmail is usually within 23 days, but customs clearance may cause delay. The supplier SHOULD advise you if there are to be delivery delays - always chase up any airmail order not received within a 36 days.

Product Review Sources in these ( ) as follows:
M = MICROpendium. Then MONTH and YEAR eg 10/87
T = TI*MES. Then ISSUE NO and PAGE (eg 4/12)

NOTE: Mains powered equipment will be for 110V and you MUST purchase a step down transformer (240v to 110v). Do not buy Modems from the USA- they will not be Telecom approved!

On to the goodies!

Tigercub Software
156 Collingwood Ave.
Columbus OH 43213
U.S.A.

The 130 programs in the Tigercub catalog have been reduced to $1.00 each, plus $1.50 per order (not per program) for the cost of the cassette or disk and packaging and mailing. Minumum order $10, please. Cassette orders can only be filled until my stock of blank cassettes is exhausted. The Handy Dandy programs and the Color Programming Tutor are no longer available on cassette, as they are too long to load from disk.

The 18 Full Disk collections, consisting of 5 or 6 of my catalog programs in one category, and the rest of the disk filled with a bonus of public domain programs of the same type, have been reduced to $5.00 each, postpaid, minimum order $10.

These prices are comparable to what many user groups are charging their own members for public domain programs, or what it would cost to download an equal number of programs from Genie or CompuServe.

The four Tips from the Tigercub disks are still available at $10 each, postpaid, and the three Nuts & Bolts disks at $15 each, postpaid.

I still have a few catalogs available at $1, deductable from first order.

SPOIL YOURSELF

MICROPENDIUM: The leading monthly devoted to the TI99/4A

----MICROPENDIUM----

P O BOX 1343, Round Rock, TX, USA, 78680
Annual Subscription (12 issues):
Air Mail US$40.00
Sea Mail US$26.50
Payable by US$ International Money Order Only.

Please send me the next 12 issues of MICROpendium.
I enclose US$_____ in U.S. funds.
NAME_______________________________
ADDRESS____________________________
____
TOWN_______________________________
______ENGLAND_____POST CODE_________

(TI*MES UK JAN88)

Back Issues are US$2.00 each by seamail and US$3.50 by airmail. Issues 1 and 2 of Volume 1 are not available. Volumes 2, 3 and 4 each have 12 issues. Volume 5 commences with the February 1988 issue. BACK ISSSUES ARE ONLY AVAILABLE TO SUBSCRIBERS.

HARDWARE MANUAL FOR THE TI 99/4A

$23

DESCRIBES:

- Console Design
- Custom Chip Operation
- TMS 9900 H/W Organization
- TMS 9900 Instruction Set
- Interfacing Pitfalls
- Console Schematics
- PEB Card Descriptions
- GROM Simulator Design
- Extended Basic Module description & schematics

CANADA & FOREIGN SEND $23.00 U.S. FUNDS

THE BUNYARD GROUP
PO Box 53171, Lubbock, TX 79453
U.S.A.

THE TIGERCUB NUTS & BOLTS DISKS

What are they? The Nuts & Bolts disks are collections of 100 or more subprograms in merge format, ready to merge into your own programs.

And what does that mean? Well, TI-99/4A Extended Basic allows the use of user-written subprograms. And what are subprograms? You know them well. CALL COLOR, CALL SOUND, CALL HCHAR - those are all subprograms which are built into the Basic language. You can write your own subprograms, to do anything that Extended Basic is capable of, and tack them onto the end of your program to be CALLed whenever you need them.
To put it in another way, using a subprogram is almost like running one program from another - except that you can access it much faster, you can pass along any values that you want to, and you can return to where you left the first program.

Also, with a disk drive you can save programs in MERGE format and then MERGE them into a program in memory. Providing that the line numbers are different, the program which you MERGE in will be added to the existing program.

The variables used in a subprogram are entirely separate from those used in the main program, therefore libraries of utility subprograms can be developed in MERGE format, and MERGEd into any program without conflict.

The Nuts & Bolts disks are libraries of such subprograms. The first disk contains 100 such subprograms, plus a tutorial on using them. Disk No.2 contains 108, and No.3 contains 140 of them in 114 files. Nothing like them has ever been offered by anyone else for the TI-99/4A computer.

These 348 subprograms have been consecutively line-numbered with high line numbers so that they will not overwrite your program line numbers, and so that any number of them may be MERGEd into a program without overwriting each other.
Advanced programming techniques have been used to make these routines as compact as possible, averaging hardly more than 3 sectors each, so that a hundred or more could be crammed onto a disk and so that they would add very little to the length of a program. If you are learning to program, you might learn a great deal by studying these subprograms.

Each disk is accompanied by several pages of printed documenation, explaining the use of each subprogram and giving a short demo routine which you can key in, run, and experiment with.

Many of these subprograms can be used by persons with almost no programming knowledge, to modify existing programs. For instance, a program written in Basic, which crashes with BAD VALUE when run in XBasic, will run with a simple CALL BXB, and CALL KILLQUIT will disable the infernal QUIT key. Many different screen character styles are available, as well as colorful wipes to replace CALL CLEAR.

However, it is the programmer who will find these disks truly invaluable. Even if you have the skill and ingenuity to develop these routines for yourself, wouldn't you rather just pay fifteen cents apiece for them?

The three Nuts & Bolts disks are available for $15 each, postpaid, from Tigercub Software, 156 Collingwood Ave., Columbus OH 43213. USA

L.L. CONNER ENTERPRISE
1521 Ferry Street LAFAYETTE INDIANA USA 47904

This company has been retailing TI99/4 products for four years. I have not dealt with them, but have heard of no complaints. VISA AND MASTERCARD ARE ACCEPTED so use them if you can. Quote expiry date and ensure delivery address is the same as the address the credit card company think you are at!

If you can guess the weight, postage is actually US$10 for 1lb, US$27 for 5lb and US$55 for 10lb. with insurance an extra US$5 up to US$500 value.

Huge list of modules. Here are some samples:
For US$10 (US$15)-Jawbreaker II (lovely!), Parsec (classic); Rabbit Trail (T.2.2)(love it); Othello; Early Learning Fun; TI Invaders, Car Wars, Munch Man, Munchmobile (like it); Ant Eater (fun);

Modules For US$15 (US$20)- Q'Bert(M.5/84); Super Fly; Donkey Kong(M.4/84); Defender(M.10/84); Pole Position (M.7/84)

Educational modules for US$15 ($20): Story Machine, Face Maker, Early Reading[speech rqd]; Reading... On,Fun,Roundup and more; Dragon Mix etc.

Tunnels of Doom module $19 ($24); Tunnels of Doom Editor [D](M.3/86)$20(25);

Personal Record Keeping$16(21); Personal Report Generator[D]$16(21)
Multiplan $39 (55); Video Chess $24(29); TI Writer $39 (55);
Mini Memory $49 (60);

Cassette programs by Moonbeam Software for $10 each (T.6.40)[all XB]:
Death Drones, Strike Force 99; Robot Runner; Zero Zone; Cavern Quest; Astromania; Garbage Belly; MoonVasion; Moonbeam Express.

MYARC HARDWARE:
256k ram card $230 (265); 512k ram card $279 (315);
Disk Controller Card $169 (205); RS232 card $120 (155)

T.I. Code Card $149 (185) [P Code disks and PEB required];
NAVARONE Cartridge Expander (specify for TI) US$24 (30)
BOOKS:
THE C PROGRAMMERS LANGUAGE $24 (34); The C Programmers Handbook $16 (23)
Small c handbook $19 (26); Starting Forth $19 (26);
Smart Programming Guide for Sprites (T.4.11)$7 (10);
The Orphan Chronicles $10 (14);
TI Writer MANUAL $10 (20)
TI Joysticks $10 (16);
WICO TRACKBALL specify for TI. $20 (30).Requires power supply bought in UK.
Joystick adaptor - Atari to TI - $8 (12)
Speech Synth $49 (60); TI 32k ram card $99 (130);

TI Artist Version 2.01 (T.12.54)[DR]$20 (25); Artist Extras $7 (11);
Artist Companion 1 $18 (24); Companion 2 and 3 $10 each (14)
The Printers Apprentice Vn 2 [DR]$23 (28); TPA Font Disk 1 $12 (16);
TPA Toolbox $23 (28)

MG: Explorer[DR](T.10.55)$24 (30); Advanced Diagnostics (T.10.54)[DR] $20 (26) DiskAssembler [DR] (M.1/87)$20 (26)

Font Writer II[DR] $25 (30); High Gravity[DR] $15 (20)

TENEX COMPUTER EXPRESS
P O Box 6578 South Bend IN USA 46660-6578

Other members and myself have purchased from this company with complete satisfaction. They have been around a LONG time- at least since December 1982.

VISA, MASTERCARD and AMEX credit cards ACCEPTED AND PREFERRED.
NB: They REQUIRE catalogue numbers with orders. Drop me an SAE or ask them for FREE TI catalogue. HOW do I summarise a fifty page TI catalogue!!!!

POSTAGE: Actual cost charged on credit card order. Ask first for heavy orders paid for by IMO. On "smaller" orders they suggest you add on the following postage:

Order Value:	<$20	$20>$39	$40>$74	$75>$149	$150>299
Postage Extra:	$12	$13	$14	$15	$16

Insurance $5 extra if required.
If postage is actually more than you send by IMO they will write to ask for the extra.

CORCOMP PERIPHERALS:
Disk controller card $150 ; RS232/PIO Card $90 ;
32k ram card US$120 ; Triple Tech Card $138 (M.12/85)
Stand alone RS232/PIO peripheral- NO PEB Required! $127
MYARC PERIPHERALS: Care for a HARD DISK! Ask Tenex for a quote, and specify INTERNAL or EXTERNAL. You need a DISK CONTROLLER CARD and a HARD DISK of course, but if you want it outside the PEB you also need a HARD DISK BOX and POWER SUPPLY and EXTERNAL CABLES. The disk supplied is 20Meg.
Disk controller card $150 (M.6/86) ; RS232 card $80 ;
256k ram card $179 ; 512k ram card $239 ;
XB Level 4 [MD] $60 or if bought TOGETHER with ram card:(M.1/87)
256k + XB [D] $229 ; 512k + XB [D] $289
Power supply to keep ram cards going with PEB switched off - caution, not for archival storage - $18 [NB:110V!!!]

SPYS DEMISE [C][XB or MiniMem-same tape] $10
TE2 MODULE $20 ; TI LOGO 2 [M]$30 (-huge manual!)
MINI MEMORY MODULE [T]$40

Games Modules: $3.50 each : TI Invaders, Munch Man, Car Wars
Modules for US$5 : Personal Record Keeping; Parsec
TI WRITER MODULE [RD]:$30 ; MULTIPLAN [RD] US$30 ;
TI Speech Synth US$50 ; TI Joysticks US$9

NEW GAMES MODULES!!!
Junkman Junior $20 (M.7/87); Anteater $14
Star Runner $20 (This is TI Runner on a module,with 25 screens)(M.1/85) Spot Shot $20 ; Barrage (SUPER!) $20
Computer War + Submarine Commander + River Rescue on disk $30

Not Polyoptics SPAD XIII flight simulator ON DISK [DR] $25 (M.3/87)
HIGH GRAVITY game [DR] $13

UNEXPANDED WORD PROCESSING:
Wordwriter MODULE- requires printer interface and printer.$40
Wordwriter Plus module: INCLUDES printer cable and ready to work with any printer with a standard parallel interface. The only extra needed is the printer. [C or D - R optional] $60 . NB: ONLY EXTRA is a PRINTER.
Navarone Console Writer Module, requires printer and interface.
Cassette or disk files. $15 only! NB: These W-P are NOT advertised as being able to read files produced with any other word processor.

BETTER MINI MEMORY ASSEMBLING with DOW EDITOR ASSEMBLER. Allows you to enter a 4k m/c program into mini mmem. Supplied on cassette. US$22 .

Its no good. Too many goodies for one page!

FORTRAN IV (said by users NOT to be Fortan IV but rather 77- I have no idea!) [DM] $50 (Programs written with this package may be loaded by non-owners using a Public Domain loader released by the authors-available from me).

FONT WRITER II [DM] $25 ;
TI ARTIST [DM] $20 ; ARTISTS COMPANIONS: 1 $18 , 2 and 3 $10 each ;
(M.8/85)(M.5/86)(T.12/54) Display Master $15 ;
ARTCONVERT (TI Artist to TI Writer) $10
Navarone Disk Fixer Module [DM] $25 .
Rapid Copy [DM] $15
SUPER SPACE- [D] a bigger mini memory! MODULE containing E/A grom plus battery backed RAM.
SUPERSPACE I (8k) $40 ; SUPERSPACE II (32k) $70

SMASH program compactor -very intelligent- [D]-$22
Display Enhancement Package (do not confuse with EDP!) [DM] $26 (M.12/85) MG Advanced Diagnostics [DM] $18 (M.7/85)(T.10/54)

DUAL CASSETTE CABLE $10 ; SINGLE CASSETTE CABLE $6 ; CASSETTE REMOTE ADAPTOR (if the remote doesnt work) $6

SPOIL YOURSELF

TRACTOR FEED FOR EPSON FX80- allows you to use narrower tractor widths $45

STANDALONE PARALLEL PRINTER INTERFACE WITH CABLE $45
PRINTER CABLES:
Parallel: PEB to Epson,Star-Micronics. 5 feet $25
RS232- TI PEB to printer ETC: five feet. $25

JOYSTICKS: Epyx 500XJ (adaptor rqd) $16
Slik Stik (Adapt rqd) $7 ; TAC 5 (adapt rqd) $ 15 ;
Wico Command Control (Adapt req) $25 ; THE BOSS (Adapt req) $17
JOYSTICK ADAPTOR (Atari to TI) $5

Console Cover $5 ; Epson RX80,FX80,MX80 cover $9
PEB COVER $12

BOOKS:
Sams Technical Service data: TI99/4A; FX80; RX80; MX80 FT- each $18
HARDWARE MANUAL FOR TI99/4A $20
PROGRAMS FOR THE TI HOME COMPUTER. A steal at $5 . Excellent.(M.5/84)
XB MANUAL. $9 ; TI Intern $18 (M.8/86)
INTRODUCTION TO ASSEMBLY LANGUAGE FOR THE TI HOME COMPUTER (Molesworth) at only $9 Bargain!
(M.7/84)(T.4/12)
Editor Assembler Manual- loose leaf- $15
COMPUTE!'S BEGINNERS GUIDE TO ASSEMBLY LANGUAGE ON THE TI99/4A by Lottrup. Excellent book. US$13.95

RAVE KEYBOARD 99/105 $200 - 105 keys, so no more two key operations. Just one key instead of FCTN-4 etc. Remove TI keyboard, fit blanking plate, plug in new keyboard to console PCB and off you go. Curly cable!

MICROPAL EXTENDED BASIC (Identical to TI XB) $50
INCLUDES a word processor and database program on both cassette and disk, they dont need extra ram, all you need is a printer and interface.

Bought separately the XB word pro, TYPWRITER, is $28 on tape and the XB data base NAME-IT is $28 on cassette. Add $2 each for disk.

MECHATRONIC XB II PLUS module- TI XB plus a bit. Runs on the hot side. If you want to use the bit map graphics (pseudo bit map only!) you need 32k ram. $80 (M.10/85)(T.11/84)

IBM DISK READER/WRITER MODULE - plug into the TI and it reads and writes IBM PC disk files which are in ASCII ONLY. REQUIRES CORCOMP DISK CONTROLLER CARD AND TWO DOUBLE SIDED DOUBLE DENSITY DRIVES! $60

Like I keep saying - there is still PLENTY out there for you to buy! And even unexpanded owners are not left out! Phew. You should see the FULL Tenex catalogue.

TEXAMENTS 53 Center Street PATCHOGUE New York USA
11772

One of the newer suppliers, they provide an exemplary service and are very
highly recommended to you. They do NOT take credit cards.

However, the postage inclusive rates quoted below are all FIRM RATES. Some of the information may differ from adverts- this is the latest, printed out for me BEFORE any ads were typeset!
Postage rates are:
SOFTWARE: 1st item $5, subsequent items $1 each (Possibly more for TurboPasc, confirmation is awaited- it is a heavy package!).
PEB CARDS: $15 each. (Myarc computer $55 each)
INSURANCE IS EXTRA $5 per order (value up to $500).

TI ARTIST: [DR](M.8/85)$19.95
ARTISTS COMPANION:(M.5/86)#1:$17.95 #2 AND #3: $9.95 EACH.
DISPLAY MASTER: $14.95 ARTIST EXTRAS:$6.95

xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)
TURBO PASC 99: Requires 32k and disk only- NO PEB CARD REQUIRED.
Single pass, UP TO 80 lines per second compilation.
Up to 50% faster than C99. Code is relocatable.
Reference manual included.
US$59.95
WINDOWS 99- supplementary disk requires Turbo Pasc 99 - object code routines that allow windows to be used with TB99. $19.95
NB: Early ads stated WINDOWS was part of basic TB package, at a higher price. Windows has now been separated. TURBO PASC 99 TOOLBOX- requires TB99- library disk of object code routines to allow graphic sound and speech commands with TB99. $19.95
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx:
IMPORTANT: Texaments WERE only permitted to supply TURBO PASC to addresses within the USA and Canada. We have now arranged for them to supply the U.K. also. Readers elsewhere- please make your own arrangements, sorry!
This is not their fault. You CAN get a US friend to obtain this product for you!
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
RAPID COPY. Used to copy library disks here!Fast. $14.95

XB DETECTIVE. Useful utility.[DR] $9.95 only.
MYARC 512K CARD US$239.95
MYARC STANDARD DISK CONTROLLER CARD:US$149.95
MYARC RS232 CARD: US$109.95 MYARC HARD DISK CONTROLLER CARD US$299.95
MYARC XB LEVEL 4: [DM] US$65
MYARC XB LEVEL 4 PLUS 512K RAM CARD: US$279.95 (POST $25 INC INSURANCE!).

Limited supply: Graphing program, screen or disk, VON GRAPH [DR] only $5.95

In looking at goodies, there are some which are available direct from the programmer:

JIM PETERSON has suppli::ed his :wn text- reprinted elsewhere in this issue
==
Another author going out on his own is EDGAR DOHMANN who has SUPER BUG 2 available- this is version 2 - on disk, with printes documentation -a 52 page manual.
Superbug 2 starts off with DEBUG supplied with Editor Assembler, and takes it further, with 32 possible commands. Load, disassemble, test,patch,save. Edgar asks for only $10 for the disk and the excellent manual- but send him some postage as well. Suggest US$15 in total.
Edgar Dohmann, Rt 5, Box 34, ALVIN, Texas, USA, 77511
==
Your User Group disk library has lots of goodies for disk owners! If you obtain any FREEWARE/FAIREWARE programs, and you use them, dont forget to send a donation to the author! To avoid undue exchange costs, you may send donations in sterling to the user group library- a $10 suggested donation is about £6.50 in sterling, or thereabouts. NOT a huge sum for a good program. A suggested $2 donation can be taken as £1.50... if you think the donation requested is too high, but you still use the program, send a smaller one: something is better than nothing. Donationes ensure continued support, and maybe the odd revision or improvement!
:::::::::::::::::
TRIO PLUS SOFTWARE have produced some good programs available as FREEWARE from your disk library AND commercially from Tenex. YOU choose! TI SINGS is a disk program for TE2 owners which sings songs for you quite tunefully, and lets you write your own singing programs. There is also an extra DATA DISK with more songs on it. They have also produced ARTCONVERT, a disk for TI Artist owners, which converts your TI Artist instances into TI Writer format.
==
Computer Download, mentioned in the last issue, have failed to provide the necessary postage quotation both for this article, and also for a specific order I wished to place with them. Therefore they do not get my order and I am forced to invite you NOT to buy from them. I am puzzled why they sent me a mail shot, only to refuse to quote for a specific order - but I think we can still do without poor service, when there remain firms who can and do offer a fast efficient service.

IMPORTANT UPDATE- TURN FOUR PAGES--->

BARRY TRAVER/Genial Traveller... had some problems with his diskazine, and took two years to get through the first years sub. He promises to do better with Volume two... on your own head!

RYTE DATA... an ad appeared from this firm in TI*MES 16. It has now been paid for, by two free copies of their newsletter. The add clearly shows Mastercard acceptable. THREE members of our group placing orders in this way have NOT had their credit cards charged. Only one member had any reply at all (he concurs with my judgement of the so called compiler but likes the GPL software).

The Ryte Data June 87 newsletter indicates that all is not well. "We have reached a crossroad with the TI market....what we'll have to do here is reserve final judgement until the end of September".

They reprint an article by Terrie Masters supporting Bytemaster...the last Bytemaster magazine I have is dated December 1986 and was received June 1987. They comment very adversely on MECHATRONIC... apparently Ryte Data had 736 orders for the 80 column card, and were unable to get any from Mechatronic. They indicated disatisfaction with the Eprom, the documentation, and the lack of promised 80 column software. I can confirm that Mechatronic documentation is truly awful. IF Ryte Data are still trading when you receive this, you order at your own risk.

BYTEMASTER (Smart Programmer). As mentioned above, no magazines have been received here. The proprietor seems to be travelling round user group meetings, but again, you send money at your own risk.

TEX-COMP

One of the oldest suppliers for the TI market.

I have no personal knowledge of their service. A few grumbles have been heard, but one of our members reports good service.

A reliable source close to the owner reports that foreign customers complaining of non-delivery are ignored as a matter of policy. TEX-COMP were offered space in this newsletter on the same terms as everyone else (free) but have declined to provide the requested information.

HAGERA RAUSCH, Bonn, GERMANY have sent us their catalogue. They have a standard postal charge of DM20. Although you are stuck with paying VAT (if assessed) because they are in the EEC there should be no additional duty- a saving of 7 to 9% on imports from the USA. However, the good folk in Germany have a standard of living WAY above ours, and expect to pay much higher prices...

I have here used exchange rates of DM3 to the pound and US$1.7 to the pound, for cross calculations...

TURBO PASCAL...US$140 plus p&p. Texaments sell for US$60 plus p&p

TI Writer, Multiplan and Logo, at US$89 each- even Parco is cheaper when you allow for postage! Tenex charge US$30 for these packages plus p&p.

Joy Paint Pal, US$30, compared to Tenex US$10.

Font Writer, US$35 compared to Tenex US$25...

Need I go on! Many thanks to this German supplier for bothering to send an English insert to us with their catalogue, but the prices are just not comparable, even AFTER you allow for saving duty. Sorry guys, we're just a lot poorer than you are.

PARCO ELECTRICS have been supporting the TI for many many years. They are excluded from this section, as many of you will already have their catalogue and be aware of their prices. In my professional capacity I am also aware of a frightening number of County Court Judgements which remain on the Registry as unsettled. Even if they have been settled but not deregistered, the number of judgements alone indicates real problems. NO MEMBER has written to tell me of money sent but products not arriving so I have no reason to warn you off, except a deep unease.

ASGARD SOFTWARE: Have proved incapable of supplying goods to overseas destinations. However they do have good products, which are forunately available from TENEX from whom you should order them.

INNOVATIVE PROGRAMMING. We have some very serious reports on this company. Suggest you do not send any money. You should be able to obtain any products of worth from other suppliers such as Temex.

QUALITY 99 SOFTWARE seem to be charging rather a lot for rather a little! You can find their programs described in the Tenex catalogue. DATABIOTICS have had some bad press regarding copyright but you can find some interesting offerings from them in the Tenex catalogue.

We have previously mentioned the severe criticism of the National Assistance Group but one of our members advises me he has dealt with them satisfactorily. We have not written to them as they have not advertised for a considerable period.

LEGENDS requires XB plus 32k plus Disk.

44 Game Screens, 50 monsters, 6 Dungeons.

Graphic Adventure program. Available from TENEX as item #68111 for only US$24.95 plus post and packing.

Legends

Recommended:-

$27.95

TRIO+ SOFTWARE Box 115-A, Liscomb, IA, USA, 50148

The TRIO+ product "TI SINGS", available commercially for US$6 is also available from the group disk library as Freeware. The following products are only available commercially from TRIO+ or Tenex:

Enveloper2- US$10 - reads a disk directory and prints out a disk jacket to cut and paste.

SINGS DATA 1.US$10....requires TE2..16 "rock" songs for your TI to sing, together with a control program to select tracks to play and order of play. Does not require TI-Sings but is compatible with it.

ARTCONVERT...US$10...converts a TI Artist pic into a form directly printable with the TI Writer Formatter, and hence suitable for letter headings etc.

PRINT WIZARD V 2.1...$25...using TI Artist fonts/pics, or those supplied, print birthday cards, posters, banners, letterheads etc upright,inverted,sideways up or down, mirrored. Fonts borders and graphics supplied. For Epson or Gemini10X printers.

Postage: None suggested, but $5 should be adequate for all except Print Wizard which has a 20page manual and comes on 3 disks- suggest $8 for that one!

TRITON. They refuse to export so I won't give their address. When TI finally pulled out and divested their stocks in Oct 1984, they passed all their stock to TRITON, who had no prior TI connection. Although they wont export, you may be able to obtain items via friends in the USA. For US residents there is a fixed shipping charge of $3.90 and a minimum order of US$19.95

TRITON XB is unique to Triton, and is an XB module with TI XB plus 33 new and 6 modified commands,plus Draw N Plot. Listed at US$59.95

Triton sell the CorComp range of peripherals and DataBiotics ram card.

TE2 module...US$14.95   Mini Memory Module...YS$19.95

SuperSpace 2 module(32k ram)..US$59.95   Ed/As module...US$32.95

Wordwriter+ (word processor AND printer interface) US$59.95

TI Writer module, disk and manual US$22.95

Printer parallel interface US$49.95   Multiplan US$22.95

The following modules $5.95 or 6 for 19.98:

Defender, Picnic Paranoia, Shamus, TI Invaders, Munchman, Parsec, Chisholm Trail, Jawbreaker II, The Attack, Blasto, Car Wars, Personal Record Keeping etc

TI Joysticks $9.95. Epyx 500XJ j/s $19.95, adaptor $4.95

The following modules are 19.95 each (special offers on some):

Spot Shot, Barrage, Munchman II, Jumpy, Spy's Demise, Blackhole, Junkman Jnr, Star Runner, Burger Builder, Centipede, Micro Pinball 2, 4A Flyer

Also: Ms PacMan $11.95, Pole Position 9.95, Jungle Hunt 9.95,
Moon Patrol 9.95,   Computer War $24.95

Connect Four 9.95, Bigfoot 8.95, Sewermania 8.95, Video Chess 19.95,
A Maze Ing $14.95, Super Demon Attack $8.95, Munch Mobile $7.95, Alpiner 7.95
Hopper 7.95, Fathom $14.95, Sneggit 7.95, Hustle 8.95,
Tunnels of Doom (cassette OR disk, specify) $9.95
Adventure (cassette or disk-specify) $9.95
All ten original Adventures together for $29.95, or 7.95 each.
Spad XIII $22.95, Zork 1,2 and 3, 14.95 each or all 3 for $34.95
HitchHikers Guide $14.95 The Witness and Planetfall, 14.95 each
LOGO 2 $19.95, Speech Synth $49.95, Graphx $19.95,.
BEST OF 99er $9.95   "READING..."modules 8.95 and 9.95
Plenty of educational modules.

DELIVERY:

Tenex. Order sent 19Oct,Received 11Nov.
Trio+ Order sent 5 Nov, Arrive 9 Dec.
Phil Trotter reports orders to Jim Peterson and Paragon Computing(EDP) delivered in ten days.

EXTRA EXTRA EXTRA

LATE SUPPLIER NEWS- Computer Download update-

I heard from Bob Boone! He replied by SEA MAIL, and although posted November 1st, his response was only received December 21st- PAST the press date for this issue of TI*MES!

Bob was kind enough to send (=sell!) a copy of BEST OF 99ER and TE2 PROTOCOL MANUAL, both added to our publications library - I also have a hint book for the first ten Infocom Adventures, which I'm keeping to my self for a little while! Bob also supplied me with a VERY rare module I've been looking for for years- READING RAINBOWS.

To business. Bob (Boone- of Computer Download Unlimited) says that many products he has are "one offs", but he has good stocks of the items now listed. He prefers to supply by surface mail, and can take C$ or an equal value of US$, but add $1.25 if paying by US$ to cover his exchange costs.

COMPUTER DOWNLOAD UNLIMITED.
25 Ottawa Street. Arnprior. ONTARIO. CANADA. K7S 1W7

PRICES QUOTED BELOW ARE CANADIAN DOLLARS!!!

C$10 each: Laws of Arithmatic, Integers, Division 1, The Attack - MODULES.
C$15 each: Household Budget, Centipede, Munchman, Defender, Car Wars, Alpiner, Pacman -MODULES.
DISK BASED:
Advanced Diagnostics, Diskassembler, Night Mission- C$25 each.
High Gravity, XBasher, XBug, Miller Graphics Game collection C$20 each
Spad XIII, LEGENDS, C$30 each.
Horizon Ramdisk- 1 MEGABYTE size, C$700 ready built!
" " - 512k size, ready built, C$410

In short supply: LOGO 2 and MULTIPLAN C$35 each. Speech Synth C$50
Atari Joystick adaptor C$8
Books-maybe 1 offs, many 2nd hand:
Beginners Guide to Assembly Language (Compute!) C$5 (SUPERB BOOK)
Orphan Survivors Handbook C$30 Orphans Chronicles C$15
Introduction to Assembly Language for the TI Home Computer C$12 (GOOD BOOK)
Programs for the TI Home Computer (Steve Davis) C$12
Smart Programming Guide for Sprites C$10
SAMS Entertainments Games in TI Basic and XB C$10

POSTAGE:
POST INCLUSIVE PRICES FOR:
Spad XIII C$36, High Gravity C$21, Orphan Survivors Handbook C$34
- all good things to have.
Logo 2 and Multiplan $40 post inclusive- these are heavy packages. LOGO 2 AT A GOOD PRICE! Fontwriter 2 at C$37 postage included. EZ Keys C$25 post inc.
In general add C$1 per book or simple module, or C$5 for heavier things.

Postage quoted is for SEA MAIL which may take a while!!!

Stephen. Stephen

From:
Boston Computer Society

DAVID J. TAUB
BCS-TIUG
JANUARY 13.1985

MERGE FORMAT

1. INTRODUCTION

The TI EXTENDED BASIC, when used in conjunction with a disk drive allows the use of the MERGE command. A program saved by the command SAVE DSK1.PROGNAME.MERGE cam be merged into a program resident in RAM by the command MERGE DSK1.PROGNAME. While programs saved with the SAVE command alone are saved in either a memory image file(PROGRAM) or INT/VARIABLE 254 file, a program saved with the MERGE option is stored in a DIS/VARIABLE 163 file. Furthermore, unlike the PROGRAM file, where statements are listed in the order entered or edited, the MERGE file is stored with each record representing statements in consecutive order. Thus one can readily access each BASIC statement as it is stored by the computer; moreover, one can make any changes in the statements directly in the individual records. One can even write statements in BASIC by writing directly to the MERGE file. How to accomplish all of the above is the goal of this article.

2. TOKENS

Computers do not read English; they prefer numbers. Although a BASIC statement may contain a key word such as "PRINT", the computer can very efficiently store this in omly one byte. In the TI, the regular ASCII characters are stored from CHR$(30) through CHR$(127). The characters CHR$(128) through CHR$(255) are used for abbreviations as mentioned above. In addition, certain instructions to help the computer differentiate among line numbers, numeric constants, and/strings are included in this group. The abbreviations for the keywords are commonly called TOKENS.

3. READING MERGE FORMAT

In each record of the MERGE file, the computer decodes the instructions as follows:

A. The first 2 bytes represent the line number. Since the maximum ASCII code is 255, the line number is formed by dividing by 256. The quotient is byte 1 and the remainder is byte 2(this is a modulus 256 system). So BYTE1=CHR$(INT(LN/256)) and BYTE2=CHR$(LN-256*BYTE1). For example, if the line number were 1200, BYTE1=CHR$(4) and BYTE2=CHR$(176). Another way to form BYTE2 is BYTE2=LN AND 255.

B. The last byte of the record must be CHR$(0) to end the record.

C. The last record to end the file must be CHR$(255)&CHR$(255). This would normally represent a line number of 65535, whereas the highest line number is really 32767, or CHR$(127)&CHR$(255).

4. THE TOKENS REVEALED

Each record is composed of two bytes at the beginning, representing the line number, CHR$(0) at the end to close the record, and up to 160 bytes in between. These are all concatenated to produce a string.

It is for this reason that the maximum length of a line of BASIC is 160 characters(when represented in tokenized form).

One can actually write a short program in BASIC to produce all of the TOKENS and other formatting code:

```
 5 !PROGRAM NAME:TOKENS
10 OPEN #1:"DSK1.FORMAT",VARIABLE 163
20 Z$=CHR$(0)
30 FOR I=128 TO 255
40 PRINT #1:Z$&CHR$(I)&CHR$(I)&Z$
50 NEXT I
60 PRINT #1:CHR$(255)&CHR$(255)
70 CLOSE #1
```

Line 10 opens the MERGE file, then lines 30 to 50 print each record. Line 40 converts the first 2 bytes to line #'s 128-255 [Z$&CHR$(I)]. It then asks the computer to store the format code as defined by CHR$(I), and it then closes the record. The file is then closed.

When this program is run and the MERGE file is brought into memory by typing NEW then MERGE DSK1.FORMAT, the program statements can then be listed as shown below. Notice that spaces are automatically included between the line number and the statement itself.

128	??	160	CLOSE	192	>	224	MIN
129	ELSE	161	SUB	193	+	225	RPT$
130	::	162	DISPLAY	194	-	226	?? (VALHEX)
131	!	163	IMAGE	195	*	227	??
132	IF	164	ACCEPT	196	/	228	?? (TERMCHAR)
133	GO	165	ERROR	197	^	229	??
134	GOTO	166	WARNING	198	??	230	?? (ALPHA)
135	GOSUB	167	SUBEXIT	199	*quoted str	231	?? (LALPHA)
136	RETURN	168	SUBEND	200	*unquoted str	232	NUMERIC
137	DEF	169	RUN	201	*line number	233	DIGIT
138	DIM	170	LINPUT	202	EOF	234	UALPHA
139	END	171	??	203	ABS	235	SIZE
140	FOR	172	??	204	ATN	236	ALL
141	LET	173	??	205	COS	237	USING
142	BREAK	174	?? (DEFINT)	206	EXP	238	BEEP
143	UNBREAK	175	?? (DEFREAL)	207	INT	239	ERASE
144	TRACE	176	THEN	208	LOG	240	AT
145	UNTRACE	177	TO	209	SGN	241	BASE
146	INPUT	178	STEP	210	SIN	242	?? (CONTINUE)
147	DATA	179	,	211	SQR	243	VARIABLE
148	RESTORE	180	;	212	TAN	244	RELATIVE
149	RANDOMIZE	181	:	213	LEN	245	INTERNAL
150	NEXT	182	)	214	CHR$	246	SEQUENTIAL
151	READ	183	(	215	RND	247	OUTPUT
152	STOP	184	&	216	SEG$	248	UPDATE
153	DELETE	185	??	217	POS	249	APPEND
154	REM	186	OR	218	VAL	250	FIXED
155	ON	187	AND	219	STR$	251	PERMANENT
156	PRINT	188	XOR	220	ASC	252	TAB
157	CALL	189	NOT	221	PI	253	#(files)
158	OPTION	190	=	222	REC	254	VALIDATE
159	OPEN	191	<	223	MAX	255	*end of file

* ENTRIES (   ) = MYARC XB ONLY *
[CONTINUE IS USED WITH RUN]

Notice that some of the codes have a question mark, since their meaning is unspecified. To see how these codes are used, one could take a line of BASIC and note how the computer translates it. For example, take the code:

```
        10 FOR I=1 TO A::PRINT "#=";I::GOSUB 100::NEXT I::CALL CLEAR
```

To convert this statement to its component format codes, SAVE the line in MERGE format as SAVE DSK1.TEST,MERGE. Then run the following program:

```
        10 OPEN #1:"DSK1.TEST",VARIABLE 163
        20 LINPUT #1:A$
        30 FOR I=1 TO LEN(A$)
        40 PRINT ASC(SEG$(A$,I,1))
        50 CALL KEY(0,X,Y)::IF Y=0 THEN 50
        60 NEXT I
        70 CLOSE #1
```

One will then get the following code, reading vertically:
The code in quotations mean that it is represented by its normal ASCII code from 32-127.

10	0 10	PRINT	156	NEXT	150
FOR	140	"#="	199 2 "35 61"	I	"73"
I	"73"	;	180	::	130
=	190	I	"73"	CALL	157
1	200 1 "49"	::	130	CLEAR	200 5 "67 76 69 65 82"
TO	177	GOSUB	135	*end	0
A	"65"	100	201 0 100		
::	130	::	130		

In reviewing the above, one sees that variable names are simply converted to the appropriate ASCII codes. Unquoted strings, including both numbers and keywords such as "CLEAR," are preceded by CHR$(200),then a byte showing the number of characters, then the ASCII representation. Quoted strings, such as "#=" above, are treated in a similar fashion except they are preceded by ChR$(199). Finally, line numbers within a program are preceded by CHR$(201) and followed by the line number in modular 256 representation.

The program in MEMORY IMAGE(PROGRAM FORMAT) is stored using a similar format except the statements are not in order. The line numbers are in a separate line number list in two bytes, with the following two bytes showing the statement's location in memory. To show some of the TOKENS as used in the PROGRAM format, enter a line number(eg. 10) followed by "!"(to avoid a syntax error), followed by control characters A to Z then ". ; = 8 9 " in that order. This gives CHR$(129) through CHR$(159). Enter the statement. When it is listed, the TOKENS appropriate to the above codes will be displayed in the statement. By pressing control 1-7 and some of the other keys, other TOKENS can be displayed as well.

5. PRACTICAL APPLICATIONS

A. MEMORY SAVER

Since numeric constants take a minimum of 3 bytes(CHR$(200) plus the length byte plus the string) and variable names take 1 byte, one can save memory by using variables.

B. CHANGE PROGRAMS

1. One could easily write a program to remove all REM statements by removing all statements whose third byte is CHR$(154) or CHR$(131). With only slightly more effort, one could also look

for tail remarks and remove statement beyond that.

2. One could index all variables, subprograms, and key words by line number to make debugging a long program easier.
3. One could write a program to allow resquencing or moving of parts of code only. This is a somewhat more difficult program to write as one must keep track of line numbers within statements by putting them into an array so that these can be changed accordingly within the individual statement.
4. One could write a program that would search for all variable names[any ASCII code <127 that is not a keyword or does not follow CHR$(199-201)] in order to reduce them to the smallest length(eg. A,B,C...)

C. WRITE A PROGRAM IN BASIC TO WRITE A PROGRAM

This could be particularly handy in writing graphics programs or musical programs. Below is an example of a simple program in BASIC that will allow you to key in the starting and ending coordinates on a line for filling in with black squares. These coordinates will then be put into appropriate HCHAR statements in a separate program entitled PICTURE, which is in MERGE format.

```
  1 !DEMO PROGRAM TO SHOW THE USE OF MERGE FORMAT TO WRITE A BASIC
    PROGRAM THAT WILL DRAW A PICTURE USING HCHAR
  2 !DAVID J. TAUB, JANUARY 13,1985
 10 GOSUB 100 !SETUP
 20 GOSUB 200 !SHOW PROMPTS
 30 GOSUB 250 !CLEAR INPUTS
 40 GOSUB 300 !ACCEPT INPUTS
 50 GOSUB 400 !# LINE NUMBER
 60 IF FL THEN GOSUB 500 ::GOTO 30 !ADD TO MERGE FILE
 70 GOSUB 600 !END FILE
 80 STOP
 90 !*******************
 91 !*SUBROUTINES BEGIN*
 92 !*******************
100 !*  SUBR FOR SETUP *
101 !*******************
105 ON WARNING NEXT::ON BREAK NEXT::CALL INIT::CALL LOAD(-31806,16)
110 OPEN #1:"DSK1.PICTURE",VARIABLE 163
120 FL=1::I=10::A$,B$,L$="  "
130 Z$=CHR$(0)::LP$=CHR$(183)::RP$=CHR$(182)::COM$=CHR$(179)::CAL$=
    CHR$(157)
140 DEF Q$(X$)=CHR$(199)&CHR$(LEN(X$))&X$
150 DEF UQ$(X$)=CHR$(200)&CHR$(LEN(X$))&X$
160 PRINT #1:Z$&CHR$(1)&CAL$&UQ$("CLEAR")&CHR$(130)&CAL$&UQ$("CHAR")
    &LP$&UQ$("96")&COM$&CHR$(225)&LP$&Q$("F")&COM$&UQ$("16")&RP$&Z$
170 CALL CLEAR
200 !******************
201 !*SUBR FOR PROMPTS*
202 !******************
210 DISPLAY AT(10,1):"LINE # DEFINED(1-24):":"   PRESS ""25"" TO END:"
220 DISPLAY AT(16.1):"STARTING POINT(1-32):": : :"STOPPING POINT(1-32):"
230 DISPLAY AT(22,11):"LAST ENTRY:"
240 RETURN
250 !********************
251 !*SUBR TO CLEAR INPUT*
252 !********************
260 DISPLAY AT(16,25):""::DISPLAY AT(19.25):""
270 DISPLAY AT(23,4):"X=";L$;"  FROM Y=";A$;" TO Y=";B$
280 RETURN
```

```
300 !****************
301 !*SUBR FOR INPUT*
302 !****************
310 ACCEPT AT(10,25)VALIDATE (DIGIT)SIZE(2):L
315 IF ABS(L-13)>12 THEN 310  ELSE IF L=25 THEN FL=0 :: GOTO 340
320 ACCEPT AT(16,25)VALIDATE (DIGIT)SIZE(2):A
325 IF A>32 OR A<1 THEN 320
330 ACCEPT AT(19,25)VALIDATE (DIGIT)SIZE(2):B
335 IF B>32 OR B<A THEN 330
340 RETURN
400 !*********************
401 !*SUBR TO CHANGE LINE*
402 !*********************
410 L1=INT(I/256)
420 L2=I AND 255
430 LN$=CHR$(L1)&CHR$(L2)
440 I=I+10
450 RETURN
500 !*****************************
501 !*SUBR TO WRITE LINE OF BASIC*
502 !*****************************
510 N$=STR$(B-A+1)
520 A$=STR$(A):: B$=STR$(B)
530 L$=STR$(L)
540 PRINT #1:LN$&CAL$&UQ$("HCHAR")&LP$&UQ$(L$)&COM$
    &UQ$(A$)&COM$&UQ$("96")&COM$&UQ$(N$)&RP$&Z$
550 RETURN
600 !*************************
601 !*SUBR TO KEEP PICTURE UP*
602 !*    AND CLOSE FILE     *
603 !*************************
610 PRINT #1:LN$&CHR$(134)&CHR$(201)&LN$&Z$
620 PRINT #1:CHR$(255)&CHR$(255)
630 CLOSE #1
640 RETURN
```

Line 105 defeats WARNING, CLEAR, and QUIT. 110 opens up MERGE file. Line 120 sets and initializes line #. Lines 130-150 define format codes to make code easier to read. 160 prints first statement: 1 CALL CLEAR::CALL CHAR(96,RPTS("F",16)) 200-240 display prompts. 250-280 clear the previous input and gives summary of it on line 23 of the screen. 300-340 input L=X co-ord, A starting Y co-ord, and B=ending Y co-ord with error checking. 400-450 calculates line # in MOD 256 and prepares next line number. 500-550 calculate N=B-A+1, which tells how many characters to print and then prints out the statement CALL HCHAR (L,A,96,N) for each input. 600-640 prints LN GOTO LN where LN is the last line number, and then closes file with CHR$(255)&CHR$(255).

This, then, completes an introduction to MERGE format. Since the same tokens are used in memory image (PROGRAM) files, one can also operate directly on statements within memory. One can, for example, look up the location of a statement in question by PEEKing into the line list (the pointer to the end of the line number list is at >8330), then look up the statement and change it on the fly. This is the technique used to catalog the disk and run a file from a menu selection. When one selects the program MYPROGRAM, for example, a line of Basic which was written as RUN DSK1.XXXXXXXXXX can be modified to say RUN DSK1.MYPROGRAM. One might also use this same technique to change a GOTO statement in a program. It could be entered as GOTO 32767 and then changed within the program to whatever line was desired. This technique would duplicate an ON GOTO statement.

HOW A BASIC PROGRAM MODIFIES ITSELF

More specifically, one can use a Basic program to modify itself as follows: It is first necessary to understand how a Basic program is organized. Each statement is entered into memory in tokenized form. Programs can fit into the 24k bytes in "high memory" from approximately >A000 to >FFFF (40960 to 65535). Note: > stands for hexadecimal. The corresponding decimal location will follow in parentheses. Each statement is entered in the order typed in (not in sequence) beginning near >FFFF and working back toward >A000. When a statement is edited, it is moved to the lower end of the memory where the last statement resides. Each statement is entered as in MERGE format, except the first two bytes for line number are missing. Instead, the first byte signifies the length of the statement (in bytes). Immediately below the statements closer to >A000 is the Line Number Table, which is constantly being moved down in memory to make room for additional statements.

Here the line numbers are listed in Hex format(identical to the modular 256 system used before) in two bytes. They are followed by two bytes to give the location in memory of that statement. In the table the line numbers are in ascending order, going from high memory, immediately below the end of the program, to lower memory. The beginning of the table is pointed to by the two bytes in >8332 and >8333(33586 and 33587), and the end of the table(in lower memory) is pointed to by >8330 and >8331(33584 and 33585).

```
                    A000<--------Begin high memory
                      :         _End Line Number Table at location
            highest st #.<-----/   pointed to by >8330. The table
                     loc            begins here with the highest
                      :             statement #(2 bytes) followed
                      :             by 2 bytes for its location.
                      :
                      :         ____Begin line Number table at location
              lowest st # <-/       pointed to by >8332.  The last 2
                     loc            bytes of the table are the location
                    XXXX  <-\       of the lowest number statement
                      :       \
                      :        \__Begin Program with last entered or
                      :             edited statement
                      :
                    XXXX  <------End program(>FFE0)
                      :
                    FFFF  <------End high memory
```

With this in mind, below is a simple program that allows one to run a program and change the DIMENSION statement while running it(as though it were dimensioned with a variable).

```
    5 !PROG NAME:DIMENSION,1/85
   10 !DAVID J. TAUB
   20 DIM A$(0000)
   30 GOSUB 30000
   40 RUN 50
   50 A$(4000)="IT WORK'S"
   60 PRINT A$(4000)
   70 STOP
30000 DISPLAY AT(12,1)ERASE ALL:"ENTER NUMBER OF DIMENSIONS":
      "DESIRED IN ARRAY A$( )": :"      DIM A$( )=:"
30010 ACCEPT AT(15,18)VALIDATE(DIGIT)SIZE(4):B$
30020 IF VAL(B$)>5000 THEN 30010
30030 CALL PEEK(-31950,A,B) !>8332
30040 LOC=256*A+B
30050 CALL INIT
30060 M=LEN(B$)
30070 CALL LOAD(LOC+7,M)
30080 FOR I=1 TO M
```

```
30090 L(I)=ASC(SEG$(B$,I,1))::CALL LOAD(LOC+I+7,L(I))
30100 NEXT I
30110 CALL LOAD(LOC+I+7,182,0)
30120 RETURN
```

After entering the program, edit line 20 to make it the first statement in the program. Then run the program and enter the dimension desired. Lines 50-60 show that it works(try running it first without line 30). The subprogram first finds the beginning of the line # Table. Since the DIM statement comes first in memory, adding 7 bytes takes one to the number of digits byte after CHR$(200). Line 30070 puts in the number of digits, M, the four digits of the dimension are then replaced with the desired value in 30080-30100. Line 30110 adds the parenthesis and CHR$(0). Finally, line 40 reruns the program at line 50, but the prescan first redimensions A$ as changed. Check line 20 after running the program to see the change!

DAVID J. TAUB

Rambles

These two from LA TopIcs
August 1987 ↓

IF N/2=INT(N/2) THEN PRINT...

This code has been around a long time. It let s you know if N is even or not. But try this one. It accomplishes same thing, but runs faster.

IF NOT N AND 1 THEN PRINT...

The code <u>IF X THEN</u> really means:

IF X<>0 THEN

With this logic we can come up with a neat flag toggle.

IF X THEN X=0 ELSE X=1

D.I.Y.

FITTING A Z.I.F. SOCKET AND RESET SWITCH TO A ROMOX MODULE

by MIKE GODDARD

December, 1987

This project was prompted for two reasons: the first was "Ah! Romox modules are very nice, now let's see what I can do to it". The second reason was the suddenm increase in price from 50 pence to £5, not quite as bad as the cost of living, but serious enough.

The object is to fit a Z.I.F. of zero insertion force socket so that the EPROM (the chip that contains the program) in the module can be changed at will and as EPROMS can be bought for a couple of pounds it does present quite a saving over the cost of a complete module. The Z.I.F. socket has a mechanism which allows the EPROM to be removed or replaced without any force on the EPROM pins at all. Although this type of socket is more expensive than the usual type of chip socket it does have obvious advantages. All of the programs which are available for the module can be put onto EPROM as all the module really is, is a vehicle to adapt EPROMS to the TI99/4a. The other thing which became quite desirable was the fitting of a reset switch to the module to enable a "warm start" after changing the EPROM, thus avoiding the neccesity of switching off. On the console below is a list of instructions for dismantling the module and fitting the modifications.

1/ DISMANTLING

Release the locking tabs of the module by putting a small screwdriver into the two slots on the underside of the module at the front and gently lever outwards. The module should now split into two parts. Remove the sticky label and discard. You should now have one section containing the circuit board and the other the lid. Place the lid to one side and with the same small screwdriver placed in the holding tab at the rear of the section, gently lever out the circuit board.

2/ REMOVE EPROM

If done with care the EPROM should be useable (but as I am informed, the EPROMS appear to have been used before fitting by ROMOX, in fact, this proved to be the case with the one I removed). Probably the best way to do this job is with the correct type of desoldering tool, which heats all of the pins at once, but I managed quite well with a solder sucker and some desoldering wick.

3/ FIT SOCKET

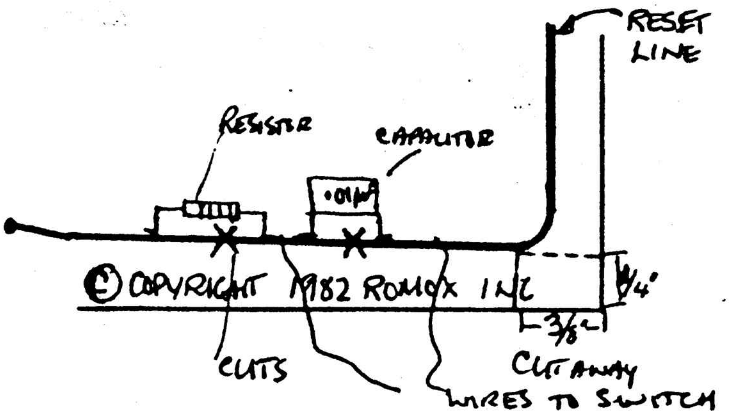
Remove all surplus solder from the board and make sure all the holes are clear. I did have trouble soldering the pins of the socket that are on the same side of the board, as the socket covers the area to be soldered. It would be a good idea to fit some short lengths of fine wire through the holes and solder them to the tracks before fitting the socket. I fitted the lever end where the notch in the EPROM should be so there should be no mistake in position. I don't know what the consequence of reversal would be and have no intention of finding out. Once the socket is fitted in, all connections should be checked for continuity to their respective connections on the edge connector.

4/ FIT RESET SWITCH

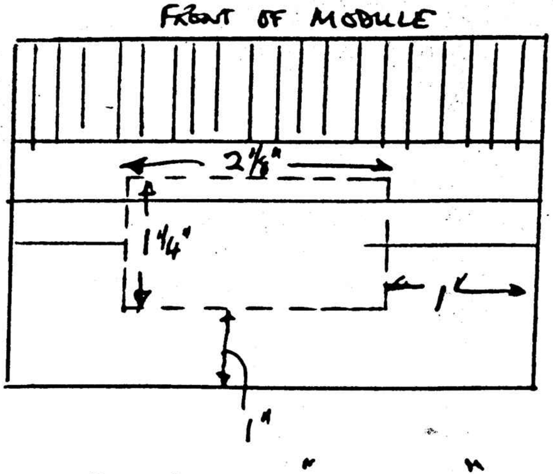
This is done by cutting away the rear left hand corner of the circuit board (see drawing). I fitted a miniature push button switch into the rear of the module case by filing a half circle cut of each section of the case, so that it lines up with the notch in the circuit board and the connections come one on top of the board and the other below. Now for the butchery bit. Make two cuts in the reset line where it follows alongside the legend "COPYRIGHT 1982 ROMOX INC", approximately 1/2" apart and 1/8" wide. Clean the laquer from the tracks and solder a resistor between 100 and 500 ohms across the gap nearest the edge connector, and a .01 microfarad capacitor across the other. These components debounce the switch contacts. Then solder a short piece of insulated wire to the track between the resistor and capacitor, and another to the track where it passes the switch side of the capacitor. These wires are then soldered to the switch.

5/ MODIFY CASE

The case needs to have a hole cut into the top to provide access to the EPROM, and also to clear the top of the EPROM as it will now sit higher in the case because of the socket. I cut a hole in mine with a fine holesaw, but a coping saw or similar, will probably do the job (see drawing for details). All there is to do now is to re-assemble the module and test it.



DETAILS OF RESET SWITCH



DETAIL OF "WINDOW"

Parts required: miniature push to make switch, 100-500 ohm resistor, .01 mf capacitor, 28 pin Z.I.F. socket.

...Cont'd over

I can have EPROMS programmed, price on application. The programs available are:-

St. Nick, Crossfire, Hopper, Ambulance, Face Chase, Centipede Log Jump, Beyond Parsec, Anteater, Astroblitz, Driving Demon Defender, Rabbit trail, Shamus, Midnite Mason and Editor Assembler.

I am now working on two more projects for ROMOX modules: one is a multi-EPROM one with switching facility and the other is a modification to use 27128 EPROMS, allowing 16K programs to be used. These will be published in due course, but if anyone wants details in advance, please get in touch.

Happy Soldering,

Mike

Mike

2021 note: This article seems to be referring to the module sold as "TRITON SUPER EXTENDED BASIC" referred to in an earlier article in this issue as "Exclusive to TRITON". Possibly the programmers sold the rights to TRITON- they are credited as authors in the Triton manual.

Super Extended Basic - A Review

L.A. 99'ers

by Ron Albright

When Miller's Graphics became simply "MG" and invented the MG/Triton Bridge Box for an XT-clone (a box that allows the TI console to control the hardware of an IBM-clone), many in the TI community lamented that one of our real ingenious people had gone the way of the "Big Blue" mentality. With the appearance of the latest offering from MG called "Super Extended Basic", those worries have been, at least partially, denied. The name of the product is no misnomer. This is, truly, what Extended Basic should have been from TI, but I am sure they didn't have the creativity to put it together. MG and its stable of elite programmers did.

Based on the work done by Mike Dodd of Knoxville, TN and Danny Michael of Muscle Shoals, AL with the incredible (and now extinct, thanks to the trade fights and rising chip prices) "Gram Kracker", the new software is a hybrid - all the compatibility of all the XB programs PLUS an incredible new array of enhancements. From "cutesy" stuff (like CALL CHIMES, and CALL HONK) to the indispensable (CALL CLOCK, MOVE and COPY), this new plug-in-and-run cartridge is an important upgrade to the XB programmer. Even with these enhancements, it would be a major upgrade in what is still the most popular programming language for the TI, but there is more. How about the "Draw 'n Plot" routines from Quality 99 Software? When you have the cartridge plugged and do a CALL FILES(2), NEW, CALL INIT, and CALL DRAWNPLOT, 6K of tight code is sucked out of the module and into Low Memory Expansion. The 15 subroutines thus added to the XB environment can be used for some remarkable hi-resolution plotting, and graphing, and saving to disk as well as dumping to your printer. You can even control the graphics with joystick or trackball. Circles, squares and lines are drawn easily straight from the XB environment - that is, from YOUR program. Further, with the CALL LINK("EDIT") command, you can directly edit, pixel by pixel, the drawing area using your joystick. Amazing stuff for sure. The art work you produce can, in turn, be called up and displayed from your XB programs (with a few restrictions) as well.

This is a tremendous addition to the XB programmer's toolbox. Combine this with Tigercub Software's Nuts and Bolts I, II, and III (156 Collingwood Avenue, Columbus, OH 43213), and anybody, and I mean ANYBODY (HEY! Yeah, I am talking to you!), can churn out some terrific code. This product is sold mail order through Triton (1-800-227-6900) and, I have been told, by Tex-Comp (818-993-5606) as well. Retail price is $59.95 plus shipping. It is a real programmer's delight - enough to make even me (a.k.a. "technoklutz" as well as "programoklutz") burp out a several line ditty with some neat graphics. Amazing, how things keep getting better and better for us despite the rigors of the Orphanage. The bowls we use may be a bit cracked and the spoons a bit tarnished from age, but this ain't gruel we're eating, folks!

Thanks to Tom and Terrie and the LA 99ers for letting me drop by again. What a group! Love to write for and be among the core staff of writer's this publication has pulled together. The best from across the country (Howie Rosenberg, Mike Dodd, Barry Traver) have joined the genius of Tom and George Steffen and the other LA locals to make this a truly national publication. Great job guys! Thanks for having me.

PROGRAMMING – FROM BASIC TO ASSEMBLY LANGUAGE

by GEOFFREY COAN
(With many thanks to Christina for typing this!)

Part II. Extended Basic

This article is a continuation of the Basic program presented in TI*MES Issue 18.

First of all, a few bug fixes and wrinkles from last month. Most of the article survived unscathed but there were a couple of typos on my part:

on page 46 the end of the explanation of Line 570 should read "ELSE 570".
on page 47 there is a "H" character missing in line 330 of the listing before the 'I=. The line should read:

```
330 T$="sc=0 <14 spaces> HI= "&STR$(HI)
```

And as was pointed out by a TI*MES reader, Line 570 was not written as clearly as it could be. The original line was:

```
570 IF (Y=23)*((X=H)+(X=H+1)) THEN 580 ELSE 590
```

which could be re-written as:

```
570 IF (Y<>23)+((X<>H)*(X<>H+l)) THEN 590
```

which is better since it removes a redundant THEN part.

On to this month's program which is the Breakout program from last month re–written in Extended Basic. The basic program from last month runs directly in Extended Basic , but I have changed it to use Extended Basic's Sprites and DISPLAY AT facility to improve it somewhat.

I have defined the ball to be sprite number 1 and the bat to be sprite numbers 2 and 3. The ball is set in motion in Line 220 and the main loop of the program then just has to check if the ball has hit anything, and if it has, to bounce it off the object by changing the ball's velocity using CALL MOTION (Line 350).

Since we cannot easily check if a sprite has hit an object in Extended Basic, I have used CALL POSITION in Line 240 to determine where the ball is (X,Y co-ordinates in SPY and SPX), and then to check the ball's location against that of the fixed objects. The problem then is the wall of bricks, which disappear as the ball hits them. To check if a brick exists where the ball is, I have defined a two-dimensional array of numbers, W, the elements of which are all set to 1 initially to show that a brick exists in every position in the wall. Line 390 of the program then sets the appropriate element of W to zero when the ball hits a brick. Thus we can simply determine if there is a brick in the wall at a given ball position by testing if the appropriate element of W is 1 (this is done in Line 300 of the program).

The control of the bat has been changed also to give a feel of acceleration to it so that if you continue to press the 'S' key to move the bat left it will slowly accelerate to the left across the screen (and similarly if you press the 'D' key to move right). If you release the key then the bat will slowly decelerate until it comes to rest. This additional feature is fairly easy to implement by keeping a note of the bat's velocity in DH and changing the bat's motion using CALL MOTION in Line 3S0 (note that the bat is two sprites, #2 and #3 placed side-by-side so both sprites' velocity must be changed at once). Now when we want to move the bat left, we simply decrement DH by 4 (Line 330), and to move right we increment DH by 4 (Line 340). To stop the bat reaching massive velocities we use the MAX and MIN functions to limit the ball's velocity to +/- 16. The bat slowly decelerates if no key is pressed by making DH move towards zero by 2 each time around the main loop,

i.e., if the bat's. motion is negative (to the left) then we add 2 to DH each time around the main loop
or if the bat's motion is positive (to the right) then we subtract 2 from DH each time around the main loop.
If the bat is at rest (DH=0) then we don't need to change DH at all.

The easiest way of doing this is to use the SGN (or sign) function of Extended Basic. This function returns -1 if its argument is negative, 0 if its argument is zero, and +1 if its argument is positive. In Line 250 I have used SGN to subtract 2 times the sign of DH from DH – thus DH will be moved 2 places "towards" zero each time around the main loop.

And that's just about it for all the changes to the program; the only other changes are to use DISPLAY AT to write strings directly to the screen instead of a FOR-NEXT loop in the Basic version (Lines 1140-1160), to use multi-statement lines where-ever possible, and to use the extension of CALL CHAR to define multiple characters in Lines 120 and 130.

How well does the program work?

Well, so much for the theory, what about the program's performance in practice?

The major problem with Extended Basic programs of this type is that the sprites (the ball and bat) are moving independently of the program and the program is always trying to "catch up" with the sprites in terms of working out where they are at any given moment,

i.e., By the time the program has CALL POSITIONed the ball and determined if the ball has hit anything, the ball has moved on to possibly hit something else!

These problems are partially alleviated by the line 240 of the program which adds the ball motion to the current ball's position to "predict" where the ball will be by the end of the main loop. This works fairly well in practice but some of the collisions with the walls and bat are still a bit bad, as sometimes the ball will bounce perfectly off the wall, and other times it will half move through the wall before it "bounces".

Unfortunately there is not a lot that can be done to cure these problems with Extended Basic. The program must be written in 'C' or machine code so that it can react fast enough to the ball's movement. However, if any TI*MES readers can come up with any improvements to the program to cure these problems, then why not share them with the rest of us?

Variables used in the program

The variables A,B,H,HI,I,J,K,S,SC,X,Y, and Z are all used the same as in the Basic program (see TI*MES Issue 18, page 45). and the following variables are new to the program:

SPX	Pixel X – co-ordinate of sprite number 1, the ball, on the screen.
SPY	Pixel Y – co-ordinate of sprite number 1, the ball on the screen. SPX and SPY differ from X and Y in that they refer to the sprite pixel location rather than the balls' row and column location.
DH	Sprite velocity of the bat along the bottom row of the screen; negative velocity ³ the bat is moving lef t, positive velocity º the bat is moving right.

Next issue.

Next issue I would like to present this program re-written in C. However, unless I am able to get a replacement disk controller card, then I will be unable to produce the program. In June, my disk drive died completely and refused to read or write disks; after consultation with the ever-helpful Francis Parrish of Parco Electrics I shipped the drive and controller card off to him. Unfortunately, the Post Office dropped the parcel en route, the drive was smashed and the controller card never made it.

I now have two replacement half-height drives but I am at present awaiting a disk controller card from America. I have been waiting three months now, and I am getting really fed up with cassettes (I lost this program once after a tape failed to load), so if anyone out there has a spare disk controller card that they don't want, I want to hear from them!

Until next time,
Geoffrey Coan,
76, Roundcroft, Romiley, Stockport, Cheshire SK6 4LS.

Program listing:

```
1 REM BREAKOUT
2 REM (EXT. BASIC VERSION)
3 REM BY GEOFFREY COAN
100 CALL CLEAR :: CALL SCREE
N(2):: RANDOMIZE :: HI=0 ::
DIM W(7,16)
110 DEF EVEN(Q)=(INT(Q/2)=Q/
2)
120 CALL CHAR(91,"0303030303
030303000000000000FFFFC0C0C0
C0C0C0C000304E8F8FF1F1723C")
130 FOR Z=96 TO 136 STEP 8 :
: CALL CHAR(Z,"FF80808080808
0FFFF010101010101FF"):: NEXT
 Z
140 FOR Z=3 TO 14 :: READ A
:: CALL COLOR(Z,16,A):: NEXT
 Z
150 CALL VCHAR(3,3,91,22)::
CALL HCHAR(2,4,92,28):: CALL
 VCHAR(3,32,93,22)
160 ! NEW GAME
170 SC=0 :: DH=0 :: DISPLAY
AT(1,2):"SC= 0        HI= "&
STR$(HI)
180 H=INT(RND*100)+80 :: CAL
L SPRITE(#2,92,16,184,H,#3,9
2,16,184,H+7)
190 ! NEW SCREEN
200 X=RND*200+40 :: Y=RND*72
+112 :: I=INT(RND*2)*16-8 ::
 J=-8
210 FOR A=6 TO 11 :: FOR B=4
 TO 30 STEP 2
220 CALL HCHAR(A,B,A*8+48)::
 CALL HCHAR(A,B+1,A*8+49)::
W(A-5,INT(B/2))=1 :: NEXT B
:: NEXT A :: CALL SPRITE(#1,
94,16,Y,X,J,I)
230 ! MAIN LOOP
240 CALL POSITION(#1,SPY,SPX
):: SPY=SPY+J :: SPX=SPX+1 :
: Y=INT(SPY/8)+1 :: X=INT(SP
X/8)+1
250 CALL POSITION(#2,A,H)::
DH=DH-2*SGN(DH):: IF H<34 OR
 H>232 THEN DH=0
260 IF SPY<17 THEN J=8
270 IF SPX<26 THEN I=8
280 IF SPX>240 THEN I=-8
290 CALL COINC(#1,#2,6,A)::
CALL COINC(#1,#3,6,B):: IF A
 OR B THEN J=-8 :: I=I+INT(D
H/2)
300 IF SPY>44 AND SPY<90 THE
N IF W(Y-5,INT(X/2))THEN GOS
UB 370 :: IF INT(SC/84)=SC/8
4 THEN 200
310 IF SPY>190 THEN 410
320 CALL KEY(3,K,S):: IF S=0
 THEN 350
330 IF K=83 AND H>34 THEN DH
=MAX(DH-4,-16)
340 IF K=68 AND H<232 THEN D
H=MIN(DH+4,16)
350 CALL MOTION(#1,J,I,#2,0,
DH,#3,0,DH):: GOTO 240
360 ! HIT WALL ROUTINE
370 J=-J :: CALL MOTION(#1,J
,I)
380 IF EVEN(X)THEN CALL HCHA
R(Y,X,32,2)ELSE CALL HCHAR(Y
,X-1,32,2)
390 SC=SC+1 :: DISPLAY AT(1,
5):SC;:: W(Y-5,INT(X/2))=0 :
: CALL SOUND(100,500,5):: RE
TURN
400 ! GAME OVER
410 CALL MOTION(#1,0,0,#2,0,
0,#3,0,0):: IF SC>HI THEN HI
=SC :: DISPLAY AT(1,24):HI
420 DISPLAY AT(15,11):"GAME
OVER";:: DISPLAY AT(18,4):"P
RESS ANY KEY TO REPLAY";
430 CALL KEY(0,K,S):: IF S=0
 THEN 430
440 CALL HCHAR(15,13,32,9)::
 CALL HCHAR(18,6,32,23):: GO
TO 170
450 ! DATA FOR COLORS
460 DATA 2,2,2,2,2,2,3,14,8,
11,6,10
```

... so get TI*YPING!

TIMES

ASSEMBLER TUTORIAL

AN INTRODUCTION TO 9900
ASSEMBLY CODE
BY
DAVID L. RAMSEY

This is an article on how to use assembly language routines in you Extended Basic Programs.

In it I will cover the major utilities at the disposal of the programmer and how to interface these with TI's Extended Basic.

I will show those interested in assembly language programming how to construct routines to read joysticks and move sprites, all at speeds far faster than those available in Extended Basic.

First, let's discuss the unusual features of the TMS 9900 microprocessor. The 9900, unlike most microprocessors used in home computers today, is a 16 bit processor. It has a 64 kilobyte direct memory address range unlike its 8 bit cousins who use various paging techniques to achieve the same memory address range.

Another important feature of the 9900 microprocessor is its lack of built in hardware registers. Only the workspace pointer, the program counter and the status register are built into the hardware. The 16 working registers are defined by the user. In addition, each subroutine can have its own set of registers. The constant "pushing" and "popping" of values onto and off of the stack is not necessary with the 9900 chip.

Another feature of the 9900 instruction set is its memory to memory architecture. This allows the programmer to perform many operations on data in memory without every moving it into a register first. For instance, I can use the instruction CB (Compare Bytes) and reference the two bytes of data being compared in the symbolic addressing mode. If I have previously defined the location LABEL and CHECK, I could use the following line of code and never move the data into the workspace registers:

```
CB @LABEL,@CHECK .
```

Yet another feature of this versatile chip is its ability to extend its own instruction set by using the XOP instruction. This gives the programmer the capability to define up to 16 of his own operations and use them in his assembly language programs. Related to this is the ability of the 9900 chip to build and use "macros". If I needed to use a type of operation where a stack became necessary, I could define the two instructions PUSH and POP and then designate a specific area of memory as the stack with the BSS (Block Starting With Symbol) directive.

The final important feature of the TI 99/4A's assembly language instruction set is its large number of built in machine language subroutines. These make it much easier for the beginning progammer to develop application programs. As an example, the TI sees the screen as a memory mapped device and cannot access it directly. Instead, values must be read into certain registers and the VDP must be given an instruction on what it is to do with these values.

The necessary instructions to access the VDP RAM can be written by the programmer but he need not bother since, with a simple BLWP (Branch and Load Workspace Pointer) instruction he can access any of a number of utilities to do precisely that. To write a single byte of data to the VDP RAM you could simply write BLWP @VSBW (Video Single Byte Write) or you could write the necessary routine yourself which would be made up of from 5 to 10 separate instructions depending upon how you decided to do it.

A SIMPLE TUTORIAL

To get a feel for using the 9900 instruction set, let's put together a simple routine to read the keyboard for input and then to output that data on the screen.

```
        DEF    READS
        REF    VSBW,KSCAN
*
HEXFF   BYTE   >FF
*
STATUS  EQU    >837C
*
BUFF1   EQU    >8375
*
READS   NOP
*
RESET   LI     R5,>0000
        LI     R6,>0300
*
KEYBD   LI     R0,>0000
        MOVB   R0,@>8374
        BLWP   @KSCAN
        CB     @HEXFF,@BUFF1
        JNE    WRITE
        JMP    KEYBD
*
WRITE   MOVB   @BUFF1,R1
        MOV    R5,R0
        BLWP   @VSBW
        INC    R5
*
LOOP    LI     R1,0
        LI     R2,6300
LOOP1   DEC    R2
        C      R1,R2
        JNE    LOOP1
        C      R5,R6
        JNE    KEYBD
        JMP    RESET
*
        END
```

This listing shows a simple program to scan the keyboard and display the input on the screen. It was put together on the Editor/Assembler package and uses some of the unique features of that package. Those of you who have the Mini Memory Module and Line-By-Line Assembler should note the following differences. First, the Editor/Assembler package supports 6 character label names, the Mini Memory Module supports only 2 character label names. Next, the Editor/Assembler package supports the DEF directive, the Mini Memory Module does not. What this means is that to use the above routine, Mini Memory Module users will need to shortern label names to 2 characters and they will need to delete the DEF directive.

Also, after they have completed entering the program they will need to make an entry in the REF/DEF table. This is what the DEF directive accomplishes for the Editor/Assembler user. Finally, the Mini Memory Module user cannot use the names of the utilities such as VSBW. Instead, they must use the address when the utility entry point is located. The BLWP @VSBW example given before becomes BLWP @>6024 with the Mini Memory Module. (Please note that the > symbol indicates a hexadecimal value).

At this point we can begin to examine the routine that is listed above.

```
        RESET LI R5,>0000
LI   R6,>0300
```

This portion of the program is given the name RESET. In it we simply load the values for the first position on the screen and the last position on he screen. In this way, we can compare our present screen location with register 6 and determine when it is time to return to the top of the screen. Register 5 is used by the program to indicate the first screen position to which we must write.

continued -

Next is the principal routine of the program, the keyboard scanning routine. It is listed below.

```
    KEYBD CLR R0
MOVB R0,@>8374
BLWP @KSCAN
CB   @HEXFF,@BUFF1
JNE  WRITE
JMP  KEYBD
```

The label that I gave this portion of the program is KEYBD. First, I placed all zeros in register 0 with the clear (CLR) instruction. Before we can access the KSCAN utility, it must know what device to scan. A value of >00 placed in memory location >8374 tells the utility to scan the entire keyboard. A value of >01 tells it to scan the left side of the keyboard and joystick #1. A value of >02 sets the utility to scan the right side of the keyboard and joystick #2. Since we want to scan the entire keyboard for this tutorial we need to place a value of >00 in memory address >8374. Now that we have a value of >0000 in register 0, all we need to do is move the most significant byte in that register to >8374. We do this with the move byte (MOVB) instruction. We place the byte from register 0 (R0) at (@) memory address >8374. Now that we have set this single necessary parameter we can access the utility with the single instruction BLWP @KSCAN.

At this point we need to check if the utility actually found the key depressed during the keyboard scan. There are two ways to accomplish this. The first method is to check the status byte and see if there has been a change since before we accessed the KSCAN utility. The other, which is simpler, is to simply check the value at >8375 with a value of >FF. This value (>FF) is placed at the location by the KSCAN utility when it finds tha no key was depressed during the scan.

Since we have used the Compare Byte (CB) instruction to make this check, we can now use any of the conditional jump statements to transfer program control. I have chosen to use he jump if not equal command because the value I am testing against is an indicator of no input. What I want to do is jump if the two values were not equal to the WRITES routine. For this I use the JNE instruction and I tell the computer to jump if not equal to WRITES. The actual instructon looks like this. If the value is equal to >FF then the jump will not occur so I need to cover those cases when no input is received. To do this I follow the conditional jump instruction with an unconditional jump instruction of JMP KEYBD.

This means that if input is found the computer goes to the WRITES routine, if no input is found, then the computer goes back to the KEYBD routine and scans the keyboard again.

This segment of our program is the heart of it. All of the other routines are built around it and with it in mind. It drives our program and we will always come back to it for more input.

The second most important routine is the WRITES routine that I mentioned above. It is listed below.

```
    WRITE   MOVB  @BUFF1,R1
MOV  R5,R0
BLWP @VSBW
INC  R5
```

The WRITES routine will write the character of the ASCII value that the computer detected during the keyboard scanning routine. To do this we use the VSBW utility. There are two things that the VSBW utility must know before i can accomplish what we want. First, in register 0 we need to have the location in the video RAM that we wish to write to and secondly, we must place in register 1 the value we wish to write to the chosen location. The value we wish to write is obviously the value at BUFF1 (>8375) which is the value we detected during the keyboard scan. We can move it into the register with the move byte instruction. Next, we need a screen location. If you think back, when the program was just beginning we loaded register 5 with just such a value. So we can use the move byte instruction to move our location into register 0. Now we can access the VSBW utility. We do this with the branch and load workspace pointer instruction and the character will appear on the screen. At this point we increment the location counter by one with the INC instruction and we fall into our delay loop.

Now, the delay loop is not a mandatory part of this program but to not use it can mean that a single press of a key could fill one half to two thirds of the screen with the same character. Those of you who are unfamilar with the speed of machine language may like to delete the lines from the label LOOP to the line with the instruction JNE LOOP1. If you assemble the program without the delay loop you will get a vivid indication of the real speed of machine language.

The delay loop is nice however because it illustrates something that every assembly and machine language programmer knows all too well. When working with assembly language, you must frequently think of ways to slow down the program rather than speed it up.

After you have it assembled, go ahead and run it with the LOAD AND RUN option on the Editor/Assembler Module.

From L.A.TOPICS

JUNE 87

SLASHED_ZERO

From "on the back burner" by Jean Wilcox (seen in the SUNCOAST BEEPER), if your printer doesn't have a slached zero in TI-Writer, use this translit- ering command:

TL. 48:48,8,47

It prints your normal zero (CHR$(48)), then backspaces (CHR$(8)) and then prints a slash (CHR$(47)) over the existing zero. Slashing zeros in dates or zip codes for your letters serves no purpose whatsoever.

One of the more important features of computers is their ability to make simple decisions. In this series – BASIC statements – I will try to show how to make use of this decision-making power.

Normally, the computer executes each line of a program in the sequence specified by the line numbers, i.e., starting at the lowest line number, going to the next highest, and so on to the end of the program. However, you can instruct it to change the normal sequence by using a GOTO statement. When the computer reads:

```
20 GOTO 40
```

it decides to ignore any lines in between Lines 20 and 40 and to go straight to Line 40, then continue as normal.

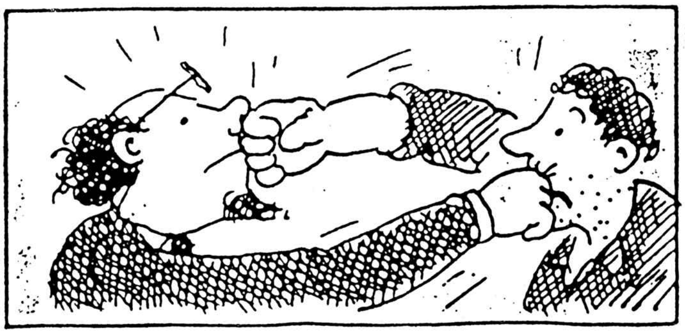
Try:

```
10 PRINT "4+3="
20 GOTO 40
30 PRINT "9"
40 PRINT "7"
50 END
```

RUN

You will see this result on the screen:

```
  4+3=
  7
** DONE **
```



Here is a simple program to demonstrate the use of GOTO:

```
10  PRINT "'TWAS ON THE COAST AT MIDNIGHT"

20  GOTO 50

30  PRINT "AND IN THEIR HOUR OF JOYMENT"

40  GOTO 70

50  PRINT "DID TOM AND PEPPER WADE"

60  GOTO 30

70  PRINT "THE MOTHS DID THEM INVADE"

80  END
```

RUN this and see my natty poem (well, what do you expect for 5 minutes' work?)! So, have a go at something more elaborate yourselves. Next time we'll look at relational expressions, and then we'll be able to put GOTO to more useful work.

– Christina

Put your speech synthesizer to work as a proofreader

By STEVEN L. RICHARDSON

If you have had a speech synthesizer attached to your TI99/4A computer for very long, it is likely that you might think of it as a novelty, a delightful addition to such games as Parsec, but one that has rarely been able to do anything useful for you. Sure, it's wonderful for education, but after the kid learns to read he is expected to fall into line and read the messages from the screen just like everybody else, and leave the speech synthesizer alone.

This article will explore a use for the little parasite that you might not have thought possible: as an assistant to help you proofread computer programs that you have typed into the computer from MICROpendium program listings, but just can't get to work right.

I don't know about you, but for me the biggest frustration I have is in typing those seemingly endless strings of 0s and Fs that will be used in creating graphics. I make other mistakes, too, of course, such as semicolons when there should be colons and insufficient parentheses enclosing math operations. Some of these surface immediately when the program is attempting to RUN, while others nestle among the program lines like kids playing hide and seek, double daring you to find them.

If you have a Terminal Emulator II module plugged into your computer and a program loaded into the memory you can immediately put the speech unit to work by typing LIST "SPEECH". Immediately the computer will begin reading the program to you as quickly as it can, line numbers and all. As it does this, you can attempt to follow along in the program listing. If the program is 200 lines long it won't stop until it's done unless you turn it off or press FCTN QUIT. Of course, you can chop the program up into smaller and more easily handled chunks by specifying which portion of the program you want to hear, as LIST "SPEECH":100-300, but there is a more serous problem to this approach. As the computer reads through the program you can never be sure whether the various words really are spelled correctly, and most repeating characters (like "FFFFFFFFFFFFFFFF") are pronounced exactly the same if there are 60 of them, or only two. To make matters worse, there are a lot of characters that are vital to the correct operation of the computer that aren't even mentioned by the speech synthesizer. For example, when is the last time you heard your computer pronouncing quotation marks, brackets or exclamation points? Probably not as often as you'd like!

Well, to make a long story short, accompanying this article is a program that will force your computer to acknowledge the existence of every symbol that can be displayed on the screen from the keyboard, and not only that, but also call it by name. Since the TE II module only operates in TI-BASIC, the program will freak out when confronted by Extended BASIC-exclusive commands, such as CALL SPRITE or LINPUT, but it handles those difficult DATA statements and most other program lines in both versions of BASIC very nicely.

You will note that the last line number for the program is 99. This is to allow it to fit before the very beginning lines of the program (starting with 100) and not interfere with the sequence as you type it in. If you don't want to use the speech portion, type RUN 100.

If you do want to use the speech program you need to follow a certain procedure: To begin with, you should list the lines of the program you will want to hear on the screen, locating them within two lines or so from the top. When you have done that, simply type RUN and the speech synthesizer will spell out everything displayed on the screen letter by letter, number by number and character by character. Mercifully, it has been instructed not to bore you with the details of how many empty spaces there are following each program line. Now, by following along in your book, you can mark the places where your typed program disagrees with the HCM version and go back to fix them when convenient. When the computer has said everything on the screen you can list the next sequence on the screen and type RUN again to hear more. When you no longer need the speech portion of the program, you can delete the first 99 lines of program, or, if you don't mind tying up 1608 bytes of memory, you can add the following line:

1 GOTO 100

After that, the main program can be RUN in the usual way.

```
-----------------
 Speak TI-BASIC
-----------------

2 REM *SPEAK TI-BASIC PRGM*
      Requires TEII Module
by S. Richardson, 11/9/86
3 OPEN #1:"SPEECH",OUTPUT
4 FOR Y=1 TO 22
5 FOR I=30 TO 1 STEP -1
6 CALL GCHAR(Y,I,A)
7 IF A<>32 THEN 9
8 NEXT I
9 FOR X=3 TO I
10 CALL GCHAR(Y,X,A)
11 GOSUB 16
12 PRINT #1:B$
13 NEXT X
14 NEXT Y
15 STOP
16 IF A>57 THEN 53
17 IF A<48 THEN 20
18 B$=CHR$(A)
19 RETURN
20 ON A-31 GOTO 21,23,25,27,
29,31,33,35,37,39,41,43,45,4
7,49,51
21 B$="SPACE"
22 RETURN
23 B$="^EXCLUH MATION POINT"
24 RETURN
25 B$="^QUOTE"
26 RETURN
27 B$="^NUMBER"
28 RETURN
29 B$="^$"
30 RETURN
31 B$="^%"
32 RETURN
33 B$="^&"
34 RETURN
35 B$="^>APOSTRUH _FEE?"
36 RETURN
37 B$="^(?"
```

Speak TI-BASIC

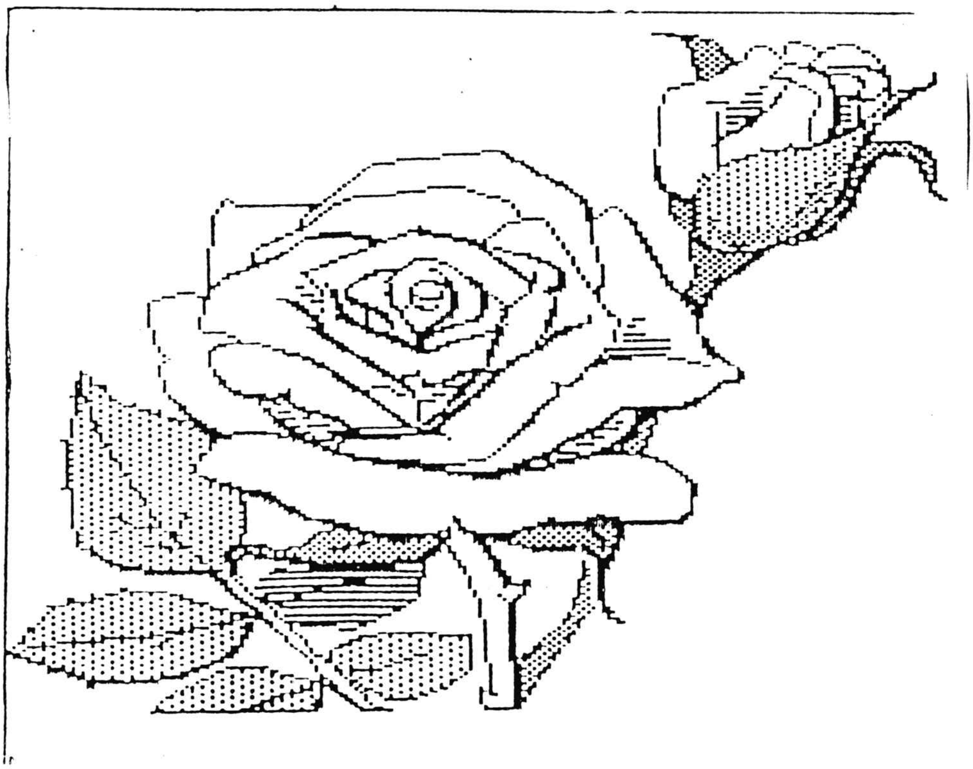
```
38 RETURN
39 B$="^)."
40 RETURN
41 B$="^*"
42 RETURN
43 B$="^PLUS"
44 RETURN
45 B$="^COMMA"
46 RETURN
47 B$="^MINE US"
48 RETURN
49 B$="^PERIOD."
50 RETURN
51 B$="^/"
52 RETURN
53 IF A>64 THEN 69
54 ON A-57 GOTO 55,57,59,61,
63,65,67
55 B$="^COLEN"
56 RETURN
57 B$="^SEMI COLEN"
58 RETURN
59 B$="^LESS THAN?"
60 RETURN
61 B$="^="
62 RETURN
63 B$="^GREATER THAN."
64 RETURN
65 B$="^QUESTION MARK?"
66 RETURN
67 B$="^@"
68 RETURN
69 IF A>96 THEN 86
70 IF A>90 THEN 73
71 B$=CHR$(A)
72 RETURN
73 ON A-90 GOTO 74,76,78,80,
82,84
```

```
FROM TIGER CUB #44

10 !TURNS ALL NUMERALS AND P
UNCTUATION WHITE! BY HARRY W
ILHELM IN TWIN TIers UG NEWS
LETTER

20 !TURN IT OFF BY CALL LOAD
(-31804,0)::TURN IT ON BY CA
LL LOAD(-31804,63)

100 CALL INIT

110 CALL LOAD(16128,2,224,38
,0,2,0,8,17,2,1,63,36,2,2,0,
3,4,32,32,36,2,224,131,192,3
,128)

120 CALL LOAD(16164,240,240,
240)

130 CALL LOAD(-31804,63)
```

MICROpendium/June 1987

```
74 B$="^( BRACKUT?"
75 RETURN
76 B$="^REVERSE SLANT"
77 RETURN
78 B$="^) BRACKUT."
79 RETURN
80 B$="^>CIRCUMFLEX"
81 RETURN
82 B$="^UNDER _LINE"
83 RETURN
84 B$="^GRAVE"
85 RETURN
86 IF A>122 THEN 89
87 B$=CHR$(A)
88 RETURN
89 ON A-122 GOTO 90,92,94,
,98
90 B$="^LEFT BRACE"
91 RETURN
92 B$="^>VERTICAL _LINE"
93 RETURN
94 B$="^RIGHT BRACE."
95 RETURN
96 B$="^>TILDA"
97 RETURN
98 B$="&DELETE >SYMBOL"
99 RETURN
```

[Above program amended per Micropendium July 1987- lines 63 and 74]



File dump program in hex and ASCII

FROM MICROpendium/June 1987 Page 37

Chuck Reinhart of Bellaire, New York, describes the following utility as a "user friendly" file dump program. "It is the ultimate in file dump programs because information about the file (such as file type and length) is not required," he writes. Readers will have noticed a number of useful utilities by Reinhart that have appeared in recent editions of MICROpendium, and this utility adds to his contributions.

The program, which requires an expansion memory, disk system and Extended Basic — a printer is optional — will dump the contents of a file to a printer or display the file contents on the screen. It will display any file type, except program.

The program is fully prompted. To run it, select the drive number of the file to be read, the output device (screen or printer), and whether you want the file to appear as hex or ASCII. Actually, as the accompanying printout shows, the hex dump includes an ASCII equivalent as well.

After making these decisions, the program will catalog the disk, listing the first 34 files it encounters to the screen. You select the file by entering the corresponding number, from 1 to 34, and the program takes care of the rest. The file reading may be stopped or restarted by depressing the spacebar.

Reinhart says the program supports the Foundation 128K card, "but due to the file structure of the card you will have to enter the file name, file type and record length for that file."

When the file has been read and outputted, another file may be read by pressing the enter key.

*See note at end

```
                    SEQUENTIAL DUMP OF

                        TIMP.DATES

                   FILE TYPE IS INTERNAL

               RECORD TYPE IS VARIABLE 100

08 40 02 00 00 00 00 00 00 00 08 40 0E 00 00 00 00 (.@........@.....)
00 00 08 40 56 00 00 00 00 00 00 00 07 32 2F 31 34 (...@V.......2/14)
2F 38 36 08 41 16 23 00 00 00 00 00 00 03 50 49 4F (/86.A.#......PIO)

                    *** END OF FILE ***
```

```
100 REM SAVE DSK2.FILE-DUMPL
110 C$="1":: GOTO 160:: DIM
B$(34)
120 CALL ERR(W,X):: GOTO 140
130 CALL ERR(A,D):: ON ERROR
  120:: CLOSE #D
140 IF D<>5 THEN PRINT::"***
  I/O ERROR -RETRY ***":::ELS
E PRINT::"      *** END OF FILE
  ***":: IF L>0 THEN PRINT #L
:::,"*** END OF FILE ***"
150 INPUT F$:: GOSUB 620:: R
ETURN 170
160 CALL CLEAR:: CALL SCREEN
(5):: FOR A=1 TO 12:: CALL C
OLOR(A,2,8):: NEXT A:: CALL
VCHAR(1,31,1,96)
170 L=0:: DISPLAY AT(2,7):"F
ILE DUMP PROGRAM":::" DSK (
1-6,X)      ";C$::::::" PRINTE
R          S"::"       OR SCREEN":
:::" HEX OUTPUT          N"
180 ACCEPT AT(5,17) SIZE(-1)
  VALIDATE("123456X") BEEP:C$
:: ACCEPT AT(11,17) SIZE(-1)
  VALIDATE("PS") BEEP:D$:: AC
CEPT AT(17,17) SIZE(-1) VALI
DATE("NY") BEEP:E$:: ON ERRO
R 130
190 IF C$<>"X" THEN DISPLAY
AT(21,2):"PLACE FILE DISK IN
  DRIVE ";C$::,"<ENTER>":: AC
CEPT AT(23,24) BEEP:F$:: GOS
UB 620:: GOTO 210 ELSE DISPL
AY AT(5,2):"FILE NAME"::::"
1-DIS/FIX":" 2-DIS/VAR":" 3-
INT/FIX"
200 DISPLAY AT(12,2):" 4-INT
/VAR"::::" RECORD LENGTH 80"
:::: ACCEPT AT(5,16) SIZE(10
) BEEP:G$:: ACCEPT AT(12,16)
 VALIDATE("1234") SIZE(1) BE
EP:F:: ACCEPT AT(16,16) VALI
DATE(DIGIT) SIZE(-3) BEEP:H:
: GOSUB 620:: GOTO 260
210 OPEN #1:"DSK"&(C$)&".",I
NPUT,RELATIVE,INTERNAL:: INP
UT #1:F$,C,C,D:: DISPLAY AT(
1,3):"AVAILABLE=";D;"USED=";
C-D::"      FILENAME","     FIL
ENAME":"      ----------","   -
---------":: FOR C=1 TO 34
220 INPUT #1:B$(C),H,A,D:: I
F LEN(B$(C))=0 THEN 240:: IF
 ABS(H)>4 THEN 220:: IF C<18
 THEN DISPLAY AT(C+4,3-LEN(S
TR$(C))):STR$(C);"-";B$(C) E
LSE DISPLAY AT(C-13,15):STR$
(C);"-";B$(C)
230 NEXT C
240 DISPLAY AT(23,3):"SELECT
 A FILE":: ACCEPT AT(23,17)
VALIDATE(DIGIT) SIZE(2) BEEP
:D:: IF D<1 OR D>C THEN 240:
: G$=B$(D):: CLOSE #1:: OPEN
 #5:"DSK"&(C$)&".",RELATIVE,
INTERNAL,INPUT:: INPUT #5:F$
,A,A,A:: I$=F$&"."&G$
250 INPUT #5:J$,F,H,H:: IF J
$<>G$OR F=5 THEN 250:: CLOSE
  #5
260 IF D$="S" THEN I=14:: J=
4:: K=22 ELSE I=41:: K=59::
L=4:: J=16:: OPEN #L:"PIO"
270 ON F GOTO 280,290,300,31
0
280 OPEN #5:"DSK"&(C$)&"."&(
G$)&"",FIXED H:: GOTO 320
290 OPEN #5:"DSK"&(C$)&"."&(
G$)&"",VARIABLE H:: GOTO 330
300 OPEN #5:"DSK"&(C$)&"."&(
G$)&"",FIXED H,INTERNAL:: GO
TO 320
310 OPEN #5:"DSK"&(C$)&"."&(
G$)&"",VARIABLE H,INTERNAL::
  GOTO 330
320 PRINT:: INPUT " ENTER ST
ARTING RECORD:":C:: FOR A=0
TO C-1:: INPUT #5:K$:: NEXT
A
330 GOSUB 620:: DISPLAY AT(1
4,12):"WORKING":: J$="SEQUEN
TIAL DUMP OF":: GOSUB 430::
J$=I$:: GOSUB 430:: J$="FILE
  TYPE IS ":: IF F<3 THEN J$=
J$&"DISPLAY" ELSE J$=J$&"INT
ERNAL"
340 GOSUB 430:: J$="RECORD T
YPE IS ":: IF F=1 OR F=3 THE
N J$=J$&"FIXED" ELSE J$=J$&"
VARIABLE"
350 J$=J$&" "&STR$(H):: GOSU
B 430:: IF E$="Y" THEN 440
```

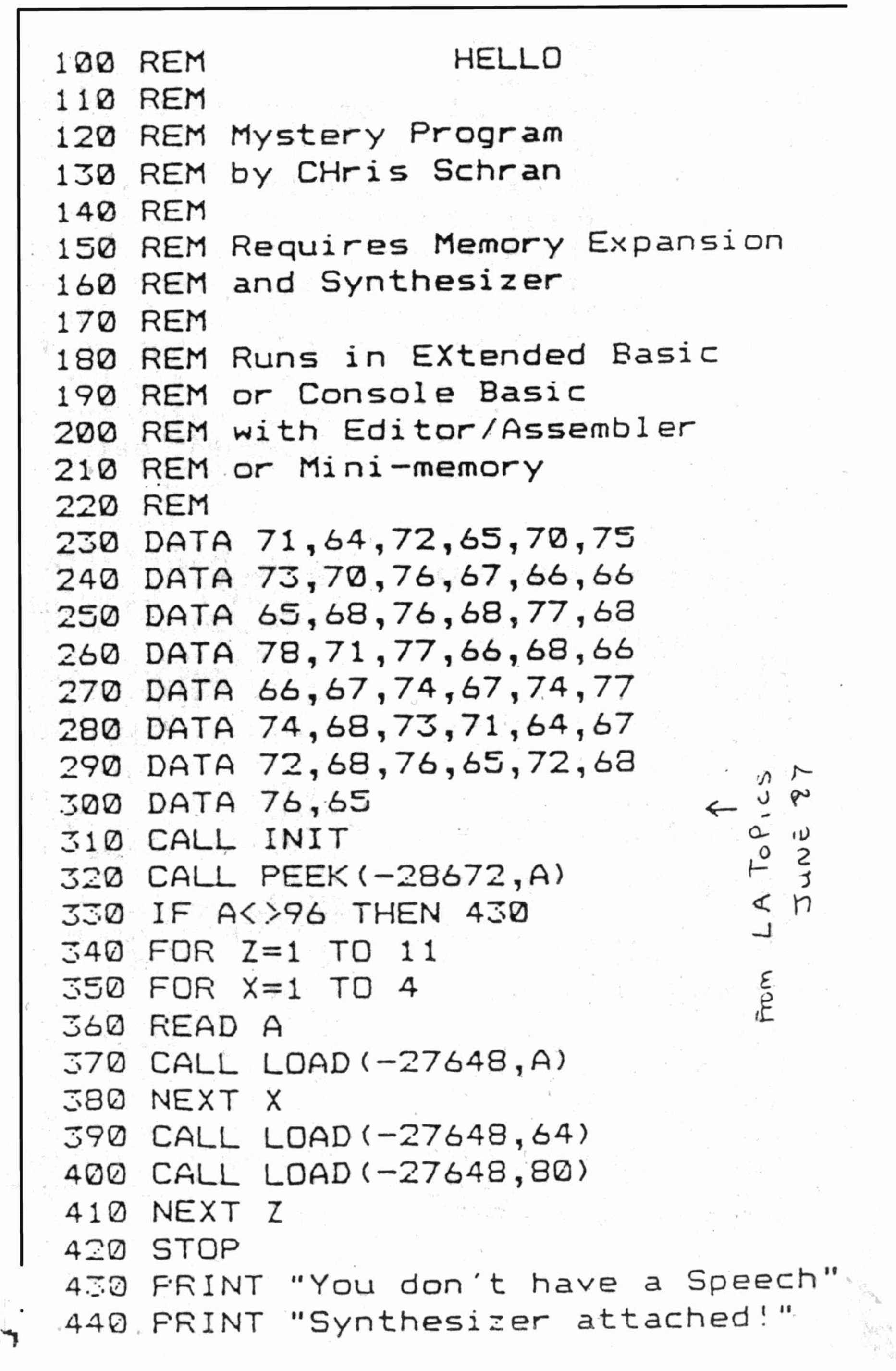
*See note at end of listing for instructions on entering the longer lines.

User Notes

```
360 INPUT #5:K$,:: N=LEN(K$)
:: IF N=0 THEN 360:: O=O+N::
  GOSUB 580:: P=1:: IF N<>8 T
HEN 410:: IF ASC(SEG$(K$,1,1
))>128 THEN 370:: IF ASC(SEG
$(K$,8,1))>0 THEN 410
370 FOR A=1 TO 8:: Q(A)=ASC(
SEG$(K$,A,1)):: B$(A)=STR$(Q
(A)):: IF (Q(A)>99)+(A>2)<>-
2 THEN 380:: P=0
380 NEXT A:: IF P=0 THEN 410
:: M$="":: IF Q(1)<128 THEN
400
390 FOR A=1 TO 2:: C=Q(A)-12
7:: B$(A)=STR$(Q(A)-C*2+A)::
  NEXT A:: M$="-"
400 M$=M$&B$(2)&".":: FOR A=
3 TO 8:: M$=M$&B$(A):: NEXT
A:: M$=M$&"E"&STR$((VAL(B$(1
))-64)*2):: PRINT #L:VAL(M$)
,:: GOTO 420
410 PRINT #L:K$;
420 PRINT #L:" ":: IF H<>255
  OR O<H THEN 360:: INPUT #5:
K$:: GOTO 360
430 PRINT #L:TAB(I-LEN(J$)/2
);J$:::: J$="":: RETURN
440 E$="0123456789ABCDEF0"::
  FOR A=1 TO 17:: B$(A)=SEG$(
E$,A,1):: NEXT A:: GOSUB 550
450 INPUT #5:K$,:: N=LEN(K$)
:: D=1:: GOSUB 460:: FOR D=D
  TO N:: N=ASC(SEG$(K$,D,1)):
: GOSUB 460:: NEXT D:: GOTO
450
460 U=INT(N/16):: V=N-16&U::
  N$=N$&B$(U+1)&B$(V+1):: IF
(N<32)+(N>126)=-1 THEN O$=O$
&"." ELSE O$=O$&CHR$(N)
470 T=T+1:: O=O+1:: GOSUB 58
0:: IF T/8<>INT(T/8) THEN 48
0:: N$=N$&" "
480 N$=N$&" ":: IF T<J THEN
490:: GOSUB 540
490 IF O<H THEN 530:: ON F G
OTO 510,530,500,530
500 IF H<>255 THEN 510:: INP
UT #5:K$
510 D=1024:: O=0:: IF T=0 TH
EN 520:: GOSUB 540
520 IF F<>3 THEN 530:: PRINT
  #L:" ":: N$="00 = "
530 RETURN
540 PRINT #L:N$;TAB(K);O$&")
"
550 IF F=3 THEN 560:: N$="
  ":: GOTO 570
560 U=INT(O/16):: V=O-16*U::
  N$=B$(U+1)&B$(V+1)&" = "
570 O$="(":: T=0:: RETURN
580 CALL KEY(0,W,X):: IF X=0
  THEN RETURN ELSE CALL SCREE
N(11):: DISPLAY AT(24,7):"**
  E = END **"
590 CALL KEY(0,W,X):: IF W=6
9 THEN 610:: IF X=1 THEN 600
  ELSE CALL SOUND(-1,1000,12)
:: GOTO 590
600 CALL SCREEN(5):: RETURN
610 CLOSE #5:: CALL SCREEN(5
):: GOSUB 620:: GOTO 170
620 DISPLAY AT(1,1):::::::::
:::::::::::::::::: RETURN
```

User Notes is a column of tips and ideas designed to help readers put their home computers to better use. The information provided here comes from many sources, including TI home computer user group newsletters. MICROpendium will pay $10 for any item sent in by readers that appears in this column. Mail tips to: MICROpendium, PO Box 1343, Round Rock, TX 78680.

* Note from the July issue regarding this listing: Extra long program lines are entered by using the FCTN REDO key. For example, you would begin by entering as much of the program line as possible. The cursor will not move after the fifth line but will stay at the lower right-hand corner of the line. Press ENTER. Now press FCTN REDO and the line will reappear. Position the cursor as the end of the program line and continue typing.



```
100 REM              HELLO
110 REM
120 REM Mystery Program
130 REM by CHris Schran
140 REM
150 REM Requires Memory Expansion
160 REM and Synthesizer
170 REM
180 REM Runs in EXtended Basic
190 REM or Console Basic
200 REM with Editor/Assembler
210 REM or Mini-memory
220 REM
230 DATA 71,64,72,65,70,75
240 DATA 73,70,76,67,66,66
250 DATA 65,68,76,68,77,68
260 DATA 78,71,77,66,68,66
270 DATA 66,67,74,67,74,77
280 DATA 74,68,73,71,64,67
290 DATA 72,68,76,65,72,68
300 DATA 76,65
310 CALL INIT
320 CALL PEEK(-28672,A)
330 IF A<>96 THEN 430
340 FOR Z=1 TO 11
350 FOR X=1 TO 4
360 READ A
370 CALL LOAD(-27648,A)
380 NEXT X
390 CALL LOAD(-27648,64)
400 CALL LOAD(-27648,80)
410 NEXT Z
420 STOP
430 PRINT "You don't have a Speech"
440 PRINT "Synthesizer attached!"
```

← From LA ToPics June 87

from 'Word Play   April 1987

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

The Game Room

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

by The TInman

FLIP FLOP

This little game of "Flip-Flop" will test your patience. When you run the program you will be presented with a square divided into nine small connected squares, one or more of which will be white and the rest red. Your job is to change the colors so that finally you have the center square red and all the rest white. You do this by pressing the number of any white square and the ajacent colors will flip from white to red or vice-versa. There is a scoring area that keeps track of the number of times you hit the keys and another area that records the shortest time for a player. I did it once in 17 tries but have not come anywhere near that on subsequent games.

```
100 REM FLIP
110 REM TI-99/4A EXTENDED BA
SIC
120 REM WESLEY R RICHARDSON
130 REM BLUEGRASS 99 COMPUTE
R SOCIETY
140 REM VARIABLES B,C(),C$,D
$,I,K,M,N,P,S
150 DIM C(9)
160 CALL CLEAR
170 DISPLAY AT(10,11):"F L I
 P"
180 DISPLAY AT(12,2):"BY WES
LEY R. RICHARDSON"
190 CALL CHAR(92,"FFFFFEFCF8
FEFEFEFEFEFEFEF8F8FFFFFFFF3F
3F3F3F3F3F3F3F3F0F0FFFFF")
!ONE
200 CALL CHAR(96,"FFFFFCF8F1
F3FFFFFFFFFCF8F0F0FFFFFFFF3F
1F8FCFCFCF8F1F3FFF0F0FFFFF")
! TWO
210 CALL CHAR(100,"FFFFFCF0F
3FFFFFFFFFFFFF3F0F8FFFFFFFF3
F1F8FCFCF1F1FCFCF8F1F3FFFFF"
)! THREE
220 CALL CHAR(104,"FFFFF3F3F
3F3F3F3F0F0FFFFFFFFFFFFFFFFF
FFFFFFF3F3F0F0F3F3F3F3FFFFF"
)! FOUR
230 CALL CHAR(108,"FFFFF0F0F
3F3F3F0F0FFFFFFFF0F0FFFFFFFF0
F0FFFFFFF3F1F8FCF8F1F3FFFFF"
)! FIVE
240 CALL CHAR(112,"FFFFFCF8F
3F3F3F0F0F3F3F3F8FCFFFFFFFF3
F1FCFFFFF3F1FCFCFCF1F3FFFFF"
)! SIX
250 CALL CHAR(116,"FFFFF8F8F
FFFFFFFFFFFFFEFEFCFCFFFFFFFF0
F0FCFCF9F9F3F3F7F7FFFFFFFFF"
)!SEVEN
260 CALL CHAR(120,"FFFFFCF8F
3F3F3F8F8F3F3F3F8FCFFFFFFFF3
F1FCFCFCF1F1FCFCFCF1F3FFFFF"
)! EIGHT
270 CALL CHAR(124,"FFFFFCF8F
3F3F3F8FCFFFFF3F8FCFFFFFFFF3
F1FCFCFCF0F0FCFCFCF1F3FFFFF"
)! NINE
280 CALL CHAR(128,"3F1F1C1C1
C1C1F1F1C1C1C1C1C1C1C3EF8F80
8000020E0E020000000000000000"
)! F
290 CALL CHAR(132,"3E1C1C1C1
C1C1C1C1C1C1C1C1C1F3F00000
00000000000000000000808F8F8"
)! L
300 CALL CHAR(136,"070101010
1010101010101010101107F0C0C
0C0C0C0C0C0C0C0C0C0C0C0C0F0"
)! I
310 CALL CHAR(140,"3F1F1C1C1
C1C1C1F1F1C1C1C1C1C1C3EE0F0!
808080818F0E0000000000000000"
)! P
320 REM INITIALIZE
330 B=99999
340 CALL CLEAR
350 CALL SCREEN(6)
360 CALL MAGNIFY(4)
370 RANDOMIZE
380 CALL CHARPAT(58,C$)
390 CALL CHAR(37,C$)
400 FOR I=1 TO 3
410 CALL CHARPAT(87+I,C$)
```

→

```
420 CALL CHAR(39+I,C$)
430 NEXT I
440 C$="FFFFFFFFFFFFFFFF"
450 C$="0000000000000000"
460 CALL CHAR(36,C$)
470 CALL CHAR(91,C$)
480 CALL CHAR(90,D$)
490 CALL COLOR(3,2,15)
500 CALL COLOR(4,2,15)
510 CALL COLOR(8,9,15)
520 REM RESTART POINT
530 N=-1
540 FOR I=1 TO 4
550 CALL SPRITE(#(I+9),124+4
*I,12,16,32+32*I)
560 NEXT I
570 FOR I=1 TO 13
580 CALL HCHAR(I+7,3,36,13)
590 NEXT I
600 DISPLAY AT(8,15):"GOAL Z
 ZZZ"
610 DISPLAY AT(9,22):"Z[Z"
620 DISPLAY AT(10,22):"ZZZ"
630 DISPLAY AT(12,15):"PRESS
Z [[["
640 DISPLAY AT(13,22):"[[["
650 DISPLAY AT(14,22):"[[["
660 DISPLAY AT(16,15):"RZRES
ET QZQUIT"
670 DISPLAY AT(18,15):"MOVES
Z     0"
680 DISPLAY AT(20,15):"BEST
Z"
690 GOSUB 1670
700 FOR I=1 TO 9
710 C(I)=9
720 CALL SPRITE(#I,88+4*I,9,
59+34*INT((I-1)/3),34*(I-3*I
NT((I-1)/3))-15)
730 NEXT I
740 P=1+INT(9*RND)
750 M=7
760 IF (P=1)+(P=3)+(P=7)+(P=
9)THEN M=10
770 IF (P=5)THEN M=11
780 GOSUB 1340
790 REM MAIN LOOP
800 GOSUB 1440
810 GOSUB 1700
820 ON K GOTO 830,890,940,10
00,1050,1120,1170,1230,1280
830 REM K=1
840 P=1 :: GOSUB 1340
850 P=2 :: GOSUB 1340
860 P=4 :: GOSUB 1340
870 P=5 :: GOSUB 1340
880 GOTO 800
890 REM K=2
900 P=1 :: GOSUB 1340
910 P=2 :: GOSUB 1340
920 P=3 :: GOSUB 1340
930 GOTO 800
940 REM K=3
950 P=2 :: GOSUB 1340
960 P=3 :: GOSUB 1340
970 P=5 :: GOSUB 1340
980 P=6 :: GOSUB 1340
990 GOTO 800
1000 REM K=4
1010 P=1 :: GOSUB 1340
1020 P=4 :: GOSUB 1340
1030 P=7 :: GOSUB 1340
1040 GOTO 800
1050 REM K=5
1060 P=2 :: GOSUB 1340
1070 P=4 :: GOSUB 1340
1080 P=5 :: GOSUB 1340
1090 P=6 :: GOSUB 1340
1100 P=8 :: GOSUB 1340
1110 GOTO 800
1120 REM K=6
1130 P=3 :: GOSUB 1340
1140 P=6 :: GOSUB 1340
1150 P=9 :: GOSUB 1340
1160 GOTO 800
1170 REM K=7
1180 P=4 :: GOSUB 1340
1190 P=5 :: GOSUB 1340
1200 P=7 :: GOSUB 1340
1210 P=8 :: GOSUB 1340
1220 GOTO 800
1230 REM K=8
1240 P=7 :: GOSUB 1340
1250 P=8 :: GOSUB 1340
1260 P=9 :: GOSUB 1340
1270 GOTO 800
1280 REM K=9
1290 P=5 :: GOSUB 1340
1300 P=6 :: GOSUB 1340
1310 P=8 :: GOSUB 1340
1320 P=9 :: GOSUB 1340
1330 GOTO 800
1340 REM SET COLOR
1350 IF C(P)<>9 THEN 1400
1360 C(P)=15
1370 CALL COLOR(#P,15)
1380 CALL HCHAR(12+INT((P-1)
/3),23+P-3*INT((P-1)/3),48+P
)
1390 RETURN
1400 C(P)=9
1410 CALL COLOR(#P,9)
1420 CALL HCHAR(12+INT((P-1)
/3),23+P-3*INT((P-1)/3),91)
1430 RETURN
1440 REM CHECK FOR SOLUTION
1450 N=N+1
1460 DISPLAY AT(18,26-LEN(ST
R$(N))):N
1470 IF C(5)<>9 THEN 1660
1480 FOR I=1 TO 4
1490 IF C(I)<>15 THEN 1660
1500 NEXT I
1510 FOR I=6 TO 9
1520 IF C(I)<>15 THEN 1660
1530 NEXT I
1540 B=MIN(B,N)
1550 GOSUB 1670
1560 DISPLAY AT(22,3):"SOLUT
ION !!"
1570 DISPLAY AT(23,3):"ONL)"
;N-M;"E(TRA MOVES"
1580 CALL SOUND(500,440,0)
1590 CALL KEY(0,K,S)
1600 IF S=0 THEN 1590
1610 IF K=81 THEN 1790
1620 IF K<>82 THEN 1590
1630 DISPLAY AT(22,1):""
1640 DISPLAY AT(23,1):""
1650 GOTO 520
1660 RETURN
1670 REM BEST SCORE
1680 DISPLAY AT(20,22):SEG$(
"     ",1,5-LEN(STR$(B)));ST
R$(B)
1690 RETURN
1700 REM WAIT FOR KEY
1710 CALL KEY(0,K,S)
1720 IF S=0 THEN 1710
1730 IF K=81 THEN 1790
1740 IF K=82 THEN 520
1750 IF (K<49)+(K>57)THEN 17
10
1760 K=K-48
1770 IF C(K)=9 THEN 1710
1780 RETURN
1790 REM QUIT
1800 CALL CHARSET
1810 END
```

FOR SALE AND WANTED

TI*MES BACK ISSUES. The following issues are still available, some in limited quantities:- Nos. 4,8,9,10,11,12,13,14,16. Prices are £1 per copy, plus the following charges for post&packing:-

Inland: 1st Class Mail 32p, 2nd Class Mail 24p.
Overseas: Surface Mail 70p, Air Mail £1.85.

APPLY: Bryan Cloud, 27 Whitehouse Road, Ipswich, SUFFOLK 1P1 5LS

****

MODULES for sale: TI INVADERS £4.95; MUNCHMAN £2.95. Prices include post and packing. Cheques to Bryan Cloud, 27 Whitehouse road, Ipswich, SUFFOLK 1P1 5LS.

****

WANTED: Extended Basic Module, also Chess module (Basic or Ext. Basic). Telephone 0494-23771 (work), 0494 32309 (home). Barry Anderson, 63 Linchfield Flats, Totteridge, High Wycombe, Bucks HP13 7QH.

****

FOR SALE: Alphacom 42 printer £50 o.n.o. Also, T.I. thermal printer (or please can someone tell me where I can get the paper for it). Also for sale the book "Initiation into the Texas Instruments TI99/4a Assembly Language" (for use with Mini Memory) £10 o.n.o. Apply: Ian James, 36, Harrison Crescent, Bedworth, Warks CV12 8SL.

****

URGENTLY WANTED! Disk Controller Card for the T.I. Peripheral Expansion Box. Any reasonable price paid. Contact; Geoffrey Coan, 76, Roundcroft, Romiley, Stockport, Cheshire SK6 4LS. Phone: 061 427 4440.

****

WANTED: Extended Basic, Educational and games modules. Plus any books, programming books relating to the TI99/4a. Contact: Ned Robinson, 9 Westgate Road, Belton, Doncaster, South Yorkshire DN9 1PY.

****

WANTED: 1. Information on the logic of the Shugart SA455 half height drive. 2. Details of the various moveable jumpers on the main PCB. 3. Has anyopne one of these drives that they wish to part with, faulty or otherwise? Contact: John Butcher, 8 Kelso Close, The Ridings, Worth, Crawley, Sussex RH10 4XH.

****

FOR SALE: C15 and C20 cassettes, new, in library cases, 35p each plus 50p post per 10. Original type Minimem batteries with solder tags £1 each, post included. £2 fitted, post included. Cassette leads made to order: £5 single, £6.50 double, post included. Apply: Mike Goddard, "Sarnia", Cemetry Road, RHOS, Nr. Wrexham, Clwyd, Wales. Tel. 0978-843547.

****

MIMIMEM Conversion to Rechargeable Battery. Send Minimem and crossed cheque £7.50, to: N.J. Petry, Tensal Technology, 15 Penrice Close, WORLE, W.S.M., AVON BS22 9AH.

****

TI MES

more →

FOR SALE:

Hardware

TI-99/4A		30.00
Expansion Box	PHP1200	60.00
32k Expansion	PHP1260	45.00
RS232 Card	PHP1220	40.00
Disk Controller	PHP1240	40.00
Disk Manager 2	PHP3089	5.00
Disk Drive internal SSSD	PHP1250	35.00
Horizon Ramdisk SS		100.00
Speech Synthesiser	PHP1500	15.00
Wired Remote Controllers	PHP1100	10.00

Cartridges

Extended Basic V110	PHM3026	25.00
Minimemory	PHM3058	35.00
Editor/Assembler	PHM3055	25.00
LOGO II	PHM3109	25.00
TI-Writer updated	PHM3111	25.00
Multiplan updated	PHM3113	25.00
Forth (with manual)		15.00
Terminal Emulator II	PHM3055	20.00
Zero Zap	PHM3036	5.00
Personal Record Keeping	PHM3013	10.00
Personal Report Generator	PHM3044	10.00
Alpiner	PHM3056	10.00
TI Invaders	PHM3053	5.00
Adventure	PHM3041	10.00
The Count	PHT6049	5.00
Savage Island Series	PHT6054	5.00

TEL. 0733 42642

FOR ALL ON THIS PAGE APPLY:

HENRY CLARK,
60, St Pauls Road
Peterborough
PE1 3DW

Public Domain Software
small c
PR Base
CFS
Games from TI*MES Library and S.Shaw
Screen dump
Neatlist
Disk Index

Mission Impossible	PHT6047	5.00
Household Budget Management	PHM3007	5.00
The Attack	PHM3031	5.00
Parsec	PHM3112	8.00
Demonstration	PHM3001	1.00

Also:
TI-99opoly
RLE
Super Bugger
DM1000 3.1
4FRONT Magazine 1-4.

Postage and packing are not included. I would be willing to deliver within 100 miles to anyone who bought the lot.

Software

Programming Aids 1	PHT6004	2.50
Programming Aids 2	PHD5005	5.00
Programming Aids 3	PHD5012	5.00
Mailing List	PHD5001	5.00
Personal Financial Aids	PHT6003	2.50
Utilities 1 (Republic)		10.00
Display Enhancement Package (Oaktree)		10.00
Advanced Diagnostics		10.00
Graphex		20.00
Disk Catalogue (Sam Pincus)		2.00

Books

Programs for the TI Home Computer	5.00
Intro to Assembly Language	5.00
Fundamentals of TI-99/4A Ass Language	5.00
Assembly Language on the TI-99/4A	5.00
Smart Programming Guide	4.00
TEII Protocol Manual (disk and paper)	5.00
Dynamic Games	2.00
Tantalizing Games	2.00
Getting Started	2.00
Initiation into TI-99/4A Ass Lang	5.00
Using & Programming the TI-99/A	2.00
Games for your TI-99/4A	1.00