090
8811

U-G-dings

from New-JUG/North

NOVEMBER 1988 Volume 6 Number 11

Officers:

President...Bill Staedeli..384-4660 Treasurer...Frank Filice 384-8797
Veep.....Walter Macieski..667-6903 Secretary...Jim Ott...... 790-6052
Editor...Henry Hein...607-988-7789 Librarian.Andy Westner..967-9154

NEXT MEETING: November 15th, Dumont H.S. Faculty Room 7-9+ P.M.

Motto: We are a family enjoying the unspeakable peace and freedom of being orphans. (Paraphrased from George Bernard Shaw)

Deo gratias!

Fidem meam obligo vexillo civitatem Americae foederatarum et rei publicae pro qua stat, Uni nationi, Deo ducente, non dividendae, cum libertate justitiaque omnibus. (If Julius Caesar was an American)

New Jersey UG/North
P.O. Box 84
Dumont, NJ 07628

User Groups: Please Reciprocate!

TIdings from
NewJug/North
PO Box 84
Dumont, NJ 07628

Editor:
Henry Hein
RD #1 Box 343 A
Otego, NY 13825
607-528-7789

Next Meeting:
November 15th
7:30 to 9 + PM
Dumont H.S. Faculty Lounge

October's Meeting Notes:

Only 8 members attended the meeting. Treasury reported a balance of $262.41 There was a demo on setting up the configuration menu for *FUNLWEB 4.1* program. There was some discussion about buying out a local school's TI hardware and software equipment but nothing was resolved.

Next Meeting Agenda:
Discussion on club's future
Nominations of officers
Establish various committees
Demos anyone?

RAMblings
By Henry

Yes, it's that time folks, NOVEMBER! Time to be grateful for the prolific abilities of our little machine. Time to appreciate those who carried us this far by subscribing to those who extended the lives of our precious beastie! And purchasing software from those who cared very much in keeping us alive so well for so long. We are now 'orphaned' for almost 5 years now, and what an exciting time it has been!

It's time to prepare ouselves for the next year to determine the future of the club. For myself I'd like to *continue writing and putting out the* newsletter but on a limited basis, perhaps quarterly or every two months. My time has become more limited and I can't keep up the pace!

An alternative would be to join neighboring clubs such as the Central Westchester which publishes an excellent quarterly NL on disk. The Queensborough UG is another well organized group. The Northern New Jersey 99ers I suggested several times last year is another worth looking into.

The CW (above) is led by the very able and venerable Art Byers who led the organized TICOFFs for the past three years and is at it again setting the next date for March 18, 1989. He is a programmer himself and does a great job at it.

Earlier this year he put out three disks full of his and others' columns, shareware, and some of his own programs in Archived/Compressed form. One took as many as three of my ds/sd to be downloaded into normal mode. Though it took a little time to un(?)archive them and further decompress Art's disk, I had a lot of material to pass on to the club. I was surprised to see some adverse criticism from two UG editors who didn't like the practice of unarchiving and decompressing files. I disagree with them and I must say that the files and programs Art put on the disk were great examples of a creative and practical mind. Art, I salute you!

The CW UG NL on disk comes with a price of membership to save the club mailing and printing costs. Writing for them is easy to do. An associate membership fee gives you FOUR disks ds/sd per year, and on line help on their BBS. Full membership of $40/yr you'll have free access to their library.

For addresses of the above groups please write me or ask John Bonito for a printout of nearby UGs.

This newsletter will include some of the files on the latest CW disk. It is a well packed disk and can contain as much text as can be found in a NL twice or three times the size of this one with programs, too. To include as much as possible in this NL I had to use a 'microprinting' utility (a modified MULTIPRINT program) to fill these pages. I know some text will be difficult to read but please bear with me in that I feel I should print as much as possible. If you have difficulty with the tiny text feel free to let me know, meanwhile use a magnifying glass.

There were some files which were much too large for MULTIPRINT or MICROPRINT to handle and I have to reformat them for QUADCOL output. Will do for next month's NL. Some of the ones printed in this NL were not formatted at all, as you can see, due to time constraints. As I paraphrase AVIS, 'I'll try harder next time!'

en passant!

We can regret losing the wisdom, wit, and writings of Ron Albright. He is giving up his column in *Computer Shopper* which will now be done by Jonathan Zittrain and the very able Barry Traver (the TI diskazine) publisher from Philadelphia. The latter is also an expert programmer and has contributed much to Lou Phillips' MYARC computer. I'm sure he has lots to say and we welcome his support columns in a major media trade magazine! He is also, by the way, a Doctor of Divinity! He wouldn't lie to you!

On WPs FOR IBM and its clones:

Nope, there's none like TIW worth mentioning. Yep, there are some that do wonderful things that some of them do well. The price tag: $150-499 (list.) Worth it? NO! The only thing going for them is the size of files they can create, with and without graphics capability. *WORDPERFECT v. 5* for example, is about the most expensive, most versatile, and could be a desktop publishing program in itself, has an enormous manual of instructions. Maybe you don't need a PHD in *WORDPERFECT*, but it may be worthwhile to spend $300 in tuition costs to learn it HANDS ON! Great program, though! But we TIw users need not go through all that, though it can't match *WORDPERFECT*'s capabilities. A good newsletter's material does not have to be overemphasized in appearance nor overshadowed by appearance. It's the content that makes it successful!

Wish I could make ours better in content, but I do have some good contributors from time to time with some good articles in other NLs through our exchange program.

On the above matter, however, I try to present as much as possible in our ten to twelve pages without frills. Also regarding this, October's NL had 11 pages of text and to keep the NL under an ounce I had to trim 1/4" to save mailing costs. With our low reserve in the treasury, that saves a LOT!

For the club's sake, I hope you don't fold! Rebuild the club with local UG's of other computer orientations, namely IBM. You'd be surprised at the things they have in common except for the memory capacity.

A NEW APPLE:

Just announced by the APPLE manufacturer is the release of a new IIc! New? Same memory size! What jerks! No REAL improvements seen in this beast AT ALL! The LASER (clone) beats 'em all and you can opt for more mem at better price without buying expensive add ons. LASER even sells better monitors! But that's if you like APPLEs. I like the tree grown ones better, they taste good! BASIC on the APPLEs has a very sour taste. Too bad there's no XBASIC as TI has. THAT would make it 'fly' better, in MY opinion! They keep 'improving' the DOS for APPLEs. We have ours built in and, you know, it is great!

Radio Shack reduced the price of its *portable IBM graphics capability color* clone by $400 last week. Maybe they feel the competition or heard of my remarks of its useless (they don't even sell the connector) external drive port! Lots of kb's, though, and a nice machine, if it weren't for this ONE, but BIG, deficiency.

Other Radio Shack clones of IBM are NOT totally compatible with programs made for the latter. Even the hardware is NOT totally compatible. So beware! They are a bit proprietory in this regard, especially in the monitors, drive controllers, and hard drives they sell for them.

As stated above, CW's Art Byers is a good programmer. His columns and comments in this NL are a rare treat for us. I include them to shed light on that there is a great deal of talent in reach in the metropolitan area. There is more! For example, Jay Leber of Long Island, Ed Machonis of the Queensboro UG, Dennis Porpora of Northern NJUG, and some others. Lookemup!

####################EOF####################

MANIPULATING DV/80 FILES Part 1
by Art Byers

The default input or output disk file for the 99/4A is a Display Variable 80 file. It is most often used for text files generated by word processing. The WRITER files are most easy to change and reformat when they have carriage returns at the end of each paragraph and heading.

Unfortunately, this newsletter format requires that we remove the cr's from our articles. If the editor of another 99'er club wants to reprint our articles, he/she has to go through the painful process of putting back the cr's so that the article can be reworked to fit that club's newsletter format.

For this reason, we now separate each paragraph with a blank line. On this disk is a program that will read the articles and change each blank line into two cr's. Two are necessary so that when you use control/2 to reformat, one goes at the end of the paragraph and the other creates a blank line between paragraphs. This program is listed below, heavily annotated with remarks so that those of you who are not XB programmers can understand what is happening. The REMs were added on TI WRITER and are not in the actual program should you "list" it to screen or printer.

In the future, we will show you how to use the TAB settings of WRITER to set up the fields of a data base, then manipulate them as you wish with an XB program that reads the DV/80 file - ie: doing math calculations, global changes in the fields, sorting, etc.

PUTTING BACK THE CARRIAGE RETURNS

```
100 ! SAVE DSK1.CR/INSERT
110 ! *******************
120 ! * Program takes a *
130 ! * blank line in a *
140 ! * DV/80  file and *
150 ! * converts  it to *
160 ! * double carriage *
170 ! * returns. 2 Disk *
180 ! * Drives Required *
190 ! * by:  Art  Byers *
200 ! * CW 99'ers  club *
210 ! * May 1988        *
220 ! *******************
```

```
230 DIM A$(50) ! Storage.
In order to speed up the
run time we will read or
write up to 50 lines at a
time,
```

```
240 CALL CLEAR :: CALL SCREE
N(4):: PRINT "PROGRAM REQUIR
ES TWO DISKDRIVES" ! color
background green, start on
instructions
```

```
250 PRINT :"PLACE INPUT DISK
 IN DRIVE 1" ! Instruction
```

```
260 PRINT :"PLACE OUTPUT DIS
K IN DRIVE 2": :"PRESS ENTER
WHEN READY" :: INPUT E$ ! ok
Everything is ready, let's
go!!.
```

```
270 CALL CLEAR :: PRINT "ENT
ER INPUT FILE NAME:":"DO NOT
 ENTER DSKx." :: INPUT INFIL
E$ :: INFILE$="DSK1."&INFILE
$ ! get input filename and
concatenate the string for
drive #1.
```

```
280 PRINT :"ENTER OUTPUT FIL
E NAME":"DO NOT ENTER DSKx."
 :: INPUT OUTFILE$ :: OUTFIL
E$="DSK2."&OUTFILE$ ! get
output filename and
concatenate for drive #2
```

```
290 DISPLAY AT(10,1)ERASE AL
L:"OPENING FILES.." :: OPEN
#1:INFILE$,INPUT :: OPEN #2:
OUTFILE$,OUTPUT !let's keep
the user posted on what's
happening, that is known as
being user friendly.
```

```
300 COUNT=0 ' Initialize the
counter
```

```
310 CALL SCREEN(11):: DISPLA
Y AT(10,1)ERASE ALL BEEP REA
DING...." !change screen
color to get attention of
user. post meesage to keep
user informed.
```

```
320 COUNT=COUNT+1 :: LINPUT
#1:A$(COUNT):: IF EOF(1) THE
N 330 :: IF COUNT=50 THEN 33
0 ELSE 320 ! Increment the
COUNT, Read one line of text
from the infile, store it
in the array A$().If we have
reached the end of the file,
jump to the 'write to the
outfile' part of program. If
the count has reached 50,
jump to the 'write to the
outfile' part of program -
otherwise, go back and read
some more text from the
infile.
```

```
330 CALL SCREEN(15):: DISPLA
Y AT(10,1)BEEP:"WRITING..."
! change screen color to
get user's attention and
display a message to keep
user informed.
```

```
340 FOR LOOP=1 TO COUNT :: I
F A$(LOOP)=" " THEN A$(LOOP)
=CHR$(13)&CHR$(13) ! we have
carried the value 'COUNT'
from READ over to WRITE, so
we can use a standard FOR-
NEXT loop to write to disk.
If we find a blank line,
change it to a set of two
carriage returns.
```

```
350 PRINT #2:A$(LOOP):: NEXT
 LOOP !This does the work of
 writing to disk.
```

```
360 IF EOF(1) THEN 370 ELSE
300 ! If we have reached the
end of the file, move on to
CLOSE files and END -
otherwise, go back and read
some more text from the
infile
```

```
370 CLOSE #1 :: CLOSE #2 ::
DISPLAY AT(10,1)ERASE ALL:"T
ASK COMPLETE" :: END ! CLOSE
all files and announce that
the program is finished. and
END program.
***************EoF*************
```

MANIPULATING DV/80 FILES, Part 2
By Art Byers

Creating a data base using TI-WRITER

Texas Instrument's TI WRITER is an exceptionally powerful software tool. Most of us who use it, hardly ever access more than half of its many features, - nor do we realize how to fully exploit those features we do use.

As one example, we all use the Tab function at one time or another to set up columns. Perhaps we do it to write source code of an assembly program, perhaps to set up columns of figures on a small chart.

One more use for the TAB is to set up fields for a small data base. On this disk is a DV/80 file created on TI WRITER called TRICITIES. It is the roster of the Tricities 99'er club of Kennewick WA.

Five lines of this file are reproduced below to show you how it has been set up by columns, using the TAB function, into the fields of a data base. You will have to use the "window" feature of the the reader to see the full line.

```
DUANE/TRINA    DUSTIN       20 NUCLEAR LANE      RICHLAND      WA 99352

EUGENE J.      WALTER       1958 THAYER          RICHLAND      WA 99?52

TERRY          TERRASS      2022 WEISKOFF        RICHLAND      WA 99352

TROY           KLINGELE     808 N SYCAMORE       PASCO         WA 99301

CLINARD V.     HILL         919 S HUNTINGTON PLACE KENNEWICK   WA 99336
```

The above fields are, obviously, Firstname, Lastname, Address, City, State, and Zip Code. It is easily possible to manipulate these fields as in any other data base. All we have to know is the starting column of each field and the length of each field (or have our working program calculate the field length).

This article will demonstrate one of the things we most often want to do with a data base,- sorting. On this disk is a program call FIELDSORT. Reboot this disk, holding down BREAK Fctn/4, once it starts to boot. At the prompt, type: RUN "DSK1.FIELDSORT"

The program itself is listed below as part of the article. It is bare bones with few bells and whistles as it is just to illustrate the points above.

You can chose to print the sorted file back to disk as a DV/80 by rewriting line 200 as:
OPEN #WP:"DSK1.MYFILE",OUTPUT - but remember to put in a blank disk before the write to disk starts.

The program allows resorting the database by any field. There is nothing amazing about any of it, but you can learn by looking at how the field length is calculated. The tail remarks in the program itself should be enough for you to follow the flow. RUNning the program will show you how successfully the DV/80 manipulation has been accomplished.

Credit is due to Grillo and Zbyszynski for their tome "Data and File Management for the TI-99/4A". Much of the technique was learned from that book.

```
100 ! SAVE DSK1.FIELDSORT
110 ! multikey sort of string data saved as structured (by column) DV/80 file
120 DATA 1,16,29,52,65,68,72
130 ! **
140 DIM S(7),N$(25)! S() is starting col.of field, N$() holds list of names
150 FOR K=1 TO 7 :: READ S(K):: NEXT K ! read structure of fields. 7th is start of non existing last field
160 OPEN #6:"DSK1.TRICITIES",INPUT :: FOR J=1 TO 25 :: LINPUT #6:N$(J):: NEXT J :: CLOSE #6 ! read data off dv80 disk file
170 CALL CLEAR :: PRINT "ENTER 0 FOR SCREEN PRINT ENTER 1 FOR PRINTER 0"
180 ACCEPT AT(23,21)VALIDATE("01")SIZE(-1)BEEP:WP ! WP= Where Print? #0 automatically goes to screen
190 IF WP=0 THEN 210
200 OPEN #WP:"PIO",OUTPUT
210 CALL CLEAR :: PRINT "Select the field on which to sort 1-6": :
220 PRINT "1 = First Name"
230 PRINT "2 = Last Name"
240 PRINT "3 = Address"
250 PRINT "4 = City"
260 PRINT "5 = State"
270 PRINT "6 = Zip Code": :"What Key? 2"
280 ACCEPT AT(23,12)VALIDATE("123456")SIZE(-1)BEEP:K1
290 PRINT #WP
300 ON K1 GOSUB 500,510,520,530,540,550 ! print what field is the key
310 C=S(K1)!C=Column number of key field
320 L=S(K1+1)-S(K1)!L=length of field
330 N=25 :: GOSUB 380 !Send to sort
340 CALL CLEAR :: FOR J=1 TO N :: PRINT #WP:N$(J):: IF J/7=INT(J/7)AND WP=0 THEN GOSUB 560 ! send to hold screen for reading
350 NEXT J :: IF WP=1 THEN CLOSE #WP
360 GOSUB 560 !send to hold screen 'til press enter
370 GOTO 170 ! start over again
380 ! **Shell-Metzner Sort**
390 M=N
400 M=INT(M/2)
410 IF M=0 THEN RETURN
420 K=N-M :: J=1
430 I=J
440 P=I+M
450 IF SEG$(N$(I),C,L)<=SEG$(N$(P),C,L)THEN 480
460 T$=N$(I):: N$(I)=N$(P):: N$(P)=T$ :: I=I-M
470 IF I>=1 THEN 440
480 J=J+1
490 IF J<=K THEN 430 ELSE 400
500 PRINT #WP:"BY FIRST NAME": : :: RETURN
510 PRINT #WP:"BY LAST NAME": : :: RETURN
520 PRINT #WP:"BY STREET ADDRESS": : :: RETURN
530 PRINT #WP:"BY CITY": : :: RETURN
540 PRINT #WP:"BY STATE": : :: RETURN
550 PRINT #WP:"BY ZIP CODE": : :: RETURN
560 PRINT :: INPUT "PRESS ENTER":A$ :: PRINT :: RETURN
570 CLOSE #WP
580 END
```

A HOMEWORK CHALLENGE!

With the example of the above fresh in your minds, let me give you club members some homework to sharpen your programming skills:

(1) Write a program that will read the data base and print out standard 3 1/2 by 15/16" mailing labels. I suggest you set up as a 3 line label with three empty lines between each name set.

(2) Take one field and fill it with varied dollars and cents numbers (ie $15.50). Next write a program that will locate that field anyway you want - for example: by column or by using POS to locate the dollar sign - and print out that field 20% higher or lower. In other words, write a program that will manipulate the money field.

(3) Last, write a program that will enable you to make global changes in ANY and all fields -that is, give everyone the same first name - OR find one particular zip code and change all examples of it to 00000.

******************EoF***************

.LM 1;RM 38;FI;AD;IN +2;CE 5
EXTENDED BASIC - STILL A GOOD CHOICE!
The case for the BASIC Language!
By ART BYERS

Part One: Introduction.

There are new guys in the 99/4A neighborhood. Among them are such stars as FORTRAN, FORTH, PILOT and SMALL C. They have lots of adherants who talk about "Like Basic" (FORTRAN), "Freedom and Exceptionally flexible" (FORTH),"Simplicity" (PILOT), and "Speed and structure" ('c'). They are Compiled languages which means they certainly run much faster than old friend XBasic. Soooooooo? Why bother with Extended Basic at all? Why not go with the New? The Better? The Faster?

One of the great things about our beloved 99/4A is that even with its limited memory, it CAN support FORTH and C and PILOT. I consider any of the computer languages that will accomplish what is needed to be fine! For me, however, Extended Basic still remains the EASIEST and BEST, most especially when coupled with Assembly Language subroutines that speed up often used important areas.

Let me try to lead you through a discussion of the pros and cons of Extended Basic without "putting down", in the slightest ANY other language for the 99/4A (including Pascal -However Pascal requires a special PEB card and those are hard to find and some early versions have bugs).

Extended Basic has many advantages from a programmer's viewpoint, not the least of which is that it is an interpreted language with a plethora of error degugging routines built in. One of the real swift pains in the neck of a compiled language is that if it is compiled containing errors or bugs, these are extremely difficult to find. This does not mean they cannot be found or that good programmers cannot produce error free compiled code. It is just that debugging, adding to, subtracting from, changing code, etc. is much easier with XB. It is a shame that TI chose to make XB a "double" interpreted language by writing it in GPL, TI's "secret" proprietary language, also interpreted, (Which to the best of my knowledge TI has NEVER released and should they have chosen to take legal action, they could make trouble for those who have violated their rights by selling GPL programs, books explaining GPL, etc. and etc.). It would have been better if the interpreter had been written in Assembly a la MYARC's XB. The added speed of MYARC's XB is a big improvement over TI's XBasic. However, The whole subject of execution speed will be covered in more detail in part 3 of this series. It deserves separate discussion because this area is what is most often raised in any and all debates on the merits of TI XB.

One of the biggest advantages of XB is its EASE OF USE AND UNDERSTANDING. BASIC itself was written just for that purpose. BASIC is suppled with such popular computers as Apple, Atari, Commodore, and IBM. This ease of use was most important in bringing better understanding of computers and use of computer languages to large numbers of Americans. For no other reason, the Basic language continues to survive.

As far as the 99/4A goes, another advantage is that the language itself resides outside the RAM areas. It is in ROM and GROM. The cover of the XB manual states that the module contains "32k bytes of preprogrammed memory". Most of the RAM is free. Additionaly, XB accesses, again with simplicity, clarity and ease, the built in ROM routines such as Device Service - printers, cassette, disk drives -, screen access and display, setting up of buffers, graphics and sprites, mathematics, etc. Many of the "new" languages save RAM memory by also accessing these same ROM routines, running at the same speed for all!

Now lets talk about available memory. Because support for Forth and 'c', for examples, must be loaded into the main 32k memory area, they do not have as much memory available as some programmers feel is absolutely necessary. This problem has been solved by using virtual memory - that is disk storage of Forth screens (blocks) or C support routines. XB support resides in console ROM and the module itself, the full 24k upper RAM is available for programs and the 8k low memory for Assembly support routines, and most of VDP RAM for string storage etc. For example, I recently purchased a Disassembler - which was written in Forth. The author plainly stated that because of the memory used by Forth itself plus the program, it was not feasible to dissassembly programs from RAM. It did its dissassembly right off the disk!. Since Basic resides in ROM, a dissassembler written for E/A or MM modules can be written in plain old BASIC, and can disassemble programs that use the 24k upper and 8k lower memory, because it resides in VDP RAM, and not overwrite the program.

Some last points! let us look at what we have to work with. We have a machine designed as a HOME computer. For almost every purpose or use at home, memory and speed available through XB are more than sufficient. We are not tracking satellites, doing high order lengthy math, searching a database the size of the national Social Security register. We have a hundred or so names on our phone list. We do not require massive spread sheets. For our normal practical purposes XB and the 99/4A can suit our needs. In fact I may be accused of HERESY, but I did almost everything with only the XB module and cassette - NO memory expansion or disk!!!

What is more, when I need a special program written to fill a personal need, I write it, dubug it and am using it in a matter of a few minutes to at most an hour. This is possible because the most frequently used XB GOSUB routines and CALL SUBs are saved on disk as MERGE files ready to be placed into a program, easily and quickly. Many programmers overlook this useful feature of XB.

The following articles in the series offer concrete evidence to backup the ideas expounded above. They are NOT a tutorial in Basic programming. Rather, they will place a point of view before you as food for thought that, hopefully, will lead to a return to some good Basic programming.
.CE 2
*********EoF*********

.LM 1;RM 38;FI;AD;IN +2;CE 6

The Case for Extended Basic Part 2
By Art Byers

EASE OF PROGRAMMING
EASE OF UNDERSTANDING
Basic Wins Hands down!!

At the start of many Assembly tutorials, authors often give examples of the same program in both Basic and Assembly. Usually, the Basic is one or two lines, but the Assembly is a screen or more. The author then points out, with pride, that the assembly code, when compiled, takes up much less RAM than does the Basic.

I suppose my reaction was not what those authors expected. I saw the programming of one line vs programming more than 20 lines. Then I remembered that in over four years of programming with Extended Basic, I have NEVER run out of memory. Basic knocks Assembly out of the ball park when it comes to ease of programming, and if you write good Basic code, you have enough RAM in the expanded 99/4A for every HOME USE. (REMEMBER my premise is that this is an excellent HOME computer).

There are other very good reasons for having assembly coupled to your Basic programs and I will cover these in another chapter. Now let me offer proof of the pudding with a specific example. The 9900 Assembly code, listed below, converts a decimal number into an Integer number (ie: converts 3.1416 [PI] to 3). The code is from M. S. Morley's book "Fundamentals of TI-99/4A Assembly Language", pages 119 and 120, and is written for the Mini Memory.

What the code does is to read an ASCII encoded multidigit number and replace the decimal point and following digits with blanks. If no decimal point is found, then the number remains unchanged.

```
.NF
    AORG >7DB4
    TEXT '3.1416'
    DATA >0D00
M1  DATA >7DB4
    LWPI >70B8
    LI   R0,>2ED0
    LI   R1,>2000
    LI   R2,>0D00
    MOV  M1,R3
         M1,R3
         M1,R3
         M1,R3
J1  CB   *R3,R1
    JEQ  J3
    CB   *R3,R2
    JEQ  J3
    CB   *R3+,R0
    JNE  J1
    DEC  R3
J2  MOVB R1,*R3+
    CB   *R3,R1
    JEQ  J3
    CB   *R3,R2
    JNE  J2
J3  B    *R11
    END
.FI;AD
```

Remember that all this code does not even take the result and print it out on the screen or a printer - You will need five more lines of code to write to the screen and more than double that to run it out to your printer!!

Here is the same thing done in Basic INCLUDING placing it on the screen:

```
PRINT INT(3.1416)
```

Now Come on!! Which would you rather sit down and program?? I rest my case on this point. There are very good reasons for higher languages and one of them is ease of writing source code.

At this point, you may accuse me of false-weighting the scale. You may say that FORTH, PILOT, FORTRAN and 'c' are equally as easy if you want to print the integer of PI to the screen. Yes, that is true - BUT - Basic does it in English, and because PRINT and INT are reserved words represented by tokens, does it with a minimum of memory.

In general you would be correct. Nonetheless, it is in ease of understanding, for the average home computer programmer, and therefore ease of programming, that Basic shows its advantage best.

Part 3 will examine Extended Basic's weakest point and Assembly's (and some other languages') strongest point - Speed of execution!

.CE 2

################EoF###############
##################################

A message from the Delphi TI NET forum.

5844 9-JUN-01:18: News
RE: A look at the CW99'ers mailbox (Re: Msg 5665)
From: SMICKELSON To: ARTBYERS

The Z80 Simulator seems it will come out as a card, for both the TI-99/4A as well as the Geneve. It will have the latest Z80 upgrade CPU, which will permit a clock speed of 12 MHz, enabling software to run 3 to 4 times as fast as it origionally did on the ADAm. The Nintendo software, if ever made available, will be through a royalty arrangement with the manufactures of same.

Back to the Z80 card, though it,(i.e. the z80 simulator software), runs on both the TI (in a Maximem) or on the the Geneve, it runs better on the Geneve, though I understand still slower than on the ADAM, which brings the necessity of having a Z80 (I think a Z8000)CPU with 64K RAM. The Z80 card will also hav e an assembler utility, to enable the user to write/program their own CPM software. Gary Bowser, maker of the card, thinks the unit can run MSX software in the Geneve. The TI or Geneve will control the I/O to disk, joysticks, screen, while the software will use the hardware environment provided by the car d, to operate a speed close to origional. Gary has made one such card on a proto board.

From CW UG

TI-Sings · P. 6
11/88

MY DEBT TO THE TI-99/4A
By Carney W. Mimms.

What do I owe the Central Westchester 9 9ers besides next quarter's dues? With apologies to our treasurer, this is no mere rhetorical question. Like many of us, my life and work have been changed by my developing interest in computers in ways I never could have anticipated a few years ago: none of this could have happened without the support and encouragement of the members of the Central Westcher 99'ers.

For a long time, I was ignorant of computers even when surrounded by them. A few trips to the computer center in college were enough to convince me that this was an experience reserved for the initiates, white-robed technicians and pale, scruffy-looking engineering types hovering around the glassed-in., airconditioned central Machine. When I next encountered computers, they had become a lot smaller and more approachable. Unfortunately, I completely failed to recognize them for what they were! By this time, I had gone to law school and become a practicing lawyer. Thinking myself reasonably well-informed and up-to-date, I was excited by the labor and time-saving potential of the then-new word processing technology.

Unfortunately, nobody told me that these things were computers, or that anybody but secretaries could benefit from using one. On reflection, I remember that one of my more intrepid colleagues confessed that he had learned to use his secretary's Wang on the weekends, and even seemed a little awed by the experience, but I must have dismissed this as meaning nothing.

My next encounter with the computer was on a more remote and rarified plane--too remote. In 1978, after reading a rave review in Scientific American, I bought and read a truly wonderful book called Godel, Escher, Bach,: an Eternal Golden Braid, by Douglas Hofstadter. In the book Hofstadter interwove fascinating information about mathematics, music, Esche r's self-referential drawings, and exciting computer programming concepts like recursion and levels of reference. I was hooked! All I wanted to do was run out and start trying these wonderful things on a computer--some computer, any compu ter! Alas, though, there were no real computers to be found: certainly none to match the machines Hofstadter's book and my overheated imagination had conjured up. Out in the real world, the available alternatives were still the monster mainframes of my college days, the inaccessible and severely functional minicomput ers of the scientific and technical world, and the dumb but functional word processors I had already passed over. Just on the horizon, though, were some simple, accessible and relatively inexpensive machines (thousands of dollars rather than tens of thousands) like the Altair and the Apple II. Since nobody was waving the future under my nose, yet, I failed to buy one.

There matters stood until the arrival of the TI 99/4A in my life. Of course, by 1981 it would have been hard to miss t he thunder of publicity surrounding Home Computers of all varieties--unless you lived in a cave somewhere in the Himalayas. I went out and signed up for an adult Ed course in computers, the usual how -to-program-in basic stuff, and began shopping for a cheap computer. The first one I picked, the Timex Sinclair, was cheap, all right, but definitely not the stuff of Hofstadterian dreams. It was slow, crude, had a lousy video display (on my giant old tube TV) and worst of all had a tiny non-standard membrane keyboard. After a frustrating week with the keyboard, I sent it back. Fortunately, by now the TI 99/4A, which had a real key board had come down to under $200, with rebate. Not only did it have a real key board, it boasted real color! graphics a n exceptionally nice dialect of Microsoft Basic and a growing collection of interesting software, much of it in audio cassettes.

At this point, the happiest computer accident of all took place, As I was browsing through the latest hardware and software for my new TI at Sears in White Plains, I encountered another new enthusiast, Art Byers. The rest is, as they say, History! Along with two others, Art and I founded the CW 99'ers.

Although I didn't now it yet, I had already gone about as far as solitary enthusiasm would take me. I was dimly aware from my experience in the adult Ed course that I had a lot to learn about the practical uses of computers, something Dr. Hofstadter had rather little to say about. Over the next few years, the member s of the Central Westchester 99ers helped me struggle to master the TI 99 4A, and even more important, to get a sense of what personal computers could and could not do. The programs we tinkered with, though they seem primitive now, actually did useful work, like estimating taxes or laying out income and expenditures in tabular form. Not to mention that greybeard of computer applications, word processing (Art Byers even wrote his own word processing program in TI Basic!). In time, we began to tackle other new and challenging applications, including telecommunicatons, another area in which turned out to be most fortunate for me.

At this point, I had made a major career shift from law to publishing and had joined a company that was just beginning to use computers, in this case, the then -new Apple Macintosh. Many of the skills I had acquired with the help of the Westchester 99ers soon became crucial. And even the skills I hadn't yet acquired, I had learned enough to fake. Telecommunications, for example, came up for the first time when I was asked if I could conduct on-line database research for the company's health newsletter. "Of course", said I, and dashed off to buy a modem and a Compuserve kit so that I could learn enough to put on a convincing display of my on-line prowess before my ignorance caught up with me. Fortunately, Art Byers, Bob Cataldo, Al Trudeau and others in the club came to my rescue and endured protracted on-line sessions and inflated phone bills while I learned the ropes. Looking back from the secure vantage point of hundreds of hours of on-line information searches and an office Compuserve account, I am grateful for all the patience everyone displayed with me.

I now appreciate the help I have received from the Westchester 99ers more than ever, since my publishing job has now evolved so that, not only is much of my time taken up with the office computers, but a substantial part of that time is spent training computer novices. Whenever my patience and good cheer begin to flag, I try to remember the patient and generous spirit of my fellow novices and former novices at the Central Westchester

To use one of my few bits of publishing jargon, more of this saga TK (to come). Ok. got it wait til I open a second log name

From CW UG

TI-dings P 7
11/88

NEWS BULLETINS FROM THE TI WORLD. New bulletins are at the top of the stack, older ones at the bottom.

08/05/88 MYARC MOVES

MYARC has moved from Basking Ridge to Martinsville, NJ. New Telephone number is 201-805-0006

08/05/88 ASGARD announces a new magazine for the TI-99/4A.

It will be issued quarterly and have from 16 to 32 pages. In a publicity release, it's described as:"an indispensible source of tips, update notices" - "New product information"- "unbiased -reviews" and the latest NEWS from the TI world.

Price: $6.00 for four issues. Write Asgard News c/o Asgard Publishing, P O box 10697, Rockville, MD 20850

08/05/88: PAUL CHARLTON, MIKE DODD AND PETER HODDIE JOIN DELPHI's TI NET SIG.

PAUL CHARLTON, the well known programmer (Fast-Term for the 99/4A and MDOS for the 9640) is now EXCLUSIVELY available on the DELPHI TI NET sig. He has been answering questions on technical topics for both computers and has had 3 lengthy conferences where everyone has been able to "talk" to Paul on line.

Paul has graduated college and is now living and working in Denver. Paul's Delphi username is TI994A.

Mike Dodd, who has been working on several programming projects for MYARC's 9640, some in collaberation with J Peter Hoddie, is also now on Delphi's TI NET and available to help you. Mike's username is MY9640.

Last, J. Peter Hoddie, author of MY-WORD, PRESCANIT, and currently working on running software for the Myarc HFDC (Hard Floppy Disk Controller) is available under the username GENEVE.

07/20/88 TIBASE being shipped.

TIBASE data base has been shipped to 9/4A owners. See June Micropendium in the club lending library for an extensive review. Even though announced long after FIRST BASE, TIBASE has reached market first. That is a coup for TEXAMENTS, the publisher.

FIRSTBASE - now long delayed, supposedly is awaiting the 9640 version (possibly being worked on by Mike Dodd and J Peter Hoddie) so that the 99/4A and 9640 versions can be released at the same time.

07/12/88 BILL KNECHT dead at age 41.

Bill Knecht, best known for his Faire-ware disks of music for the 99/4A has passed away after a long illness. He was a member of the Houston UG and had been sysop of their BBS. He is missed by the TI community.

07/15/88 TURBO PASCAL for the TI-99 Now Being shipped.

L L Conner Enterprises has announced that they are the exclusive North American distributor for TURBO-PASC 99 AND THAT THEY ARE SHIPPING: Cost 59.95 It runs out of EA option 5 or TIW opt. #3. For more info, write L L CONNER at 1521 Ferry St., Layfayette, IN 47904 of phone 317-742-8146

06/12/88 TICOFF sets '89 date.

Saturday March 18th, 1989 is the date for TICOFF'89. This year Art Byers will be in charge of speakers and events, including classrooms for the IBM as well as 99/4A. As in the past, Carney Mimms will be TICOFF's legal advisor. More news of TICOFF will follow closer to fair date.

05/15/88 Chicago TI Fair Set for Nov 12

Place: Holiday Inn, 3505 Algonquin Road
Rolling Meadows, Ill.
312 259 5000
Chicago TI Club hot line fair phone:
312 775 0051
Schedule: Social Mixer Friday Nov 11
8pm to 12 pm $4.00 admission
FAIRE Sat Nov 12. 9am to 6pm
DINNER Sat Nov 12 7pm to 9:30 pm
admission $10.00
Hotel rates $55.00 single/double
65.00 tower
Limosine service: FREE to and

NEWS OF TICOFF'89
FAIR DATE: March 18, 1989

On Sunday, June 12th, the first organizational meeting for TICOFF'89 took place in Montvale NJ. Here is the leadership for '89.

Overall Director, Robert Guellnitz of the NEWJUG club. Bob is also in charge of Publicity and public relations. Bob is "where the buck stops"!!! His right and left hands are:

(1) Operations Coordinator, Dave Green. Dave belongs to 3 different Jersey TI Clubs and has a Geneve. Operations covers the set up, physical layout of tables, admission setup, vendor unloading and loading assistance, parking - the day of show work.

(2) PreShow Coordinator, Dennis Porpora of NEWJUG-NORTH. Dennis is in charge of most of the work that must be done before the show. Reporting to him are:

(a) User Group Chairman, Jay Leber of the TI Tex Club, LI, NY. Jay is one of the Sysops on the TI SOURCE BBS run out of Texaments shop on Long Island. Jay will also moderate the User Group open discussion that will be held at TICOFF'89.

(b) Vendor Chairman, Frank Bubenik. Frank is President of the Long Island 99'ers club.

(c) Speakers and special events Chairman, Art Byers of the Central Westchester 99'ers.

Legal advice is supplied by Carney W. Mimms of the CW 99'ers. Art Byers also acts as "advisor in general" because of his experience as Co-director for two years.
###############EOF###################

Excerpted from TId Bits #19
-By Steve Mickelson, President 9T9
Users Group, Toronto

Z-80 Update:

Got a request for more information re: Jim Ballentine's Z80 Simulator, from Greg McGill, President of the Tri-Valley 99ers UG, via GEnie mail. The best person to answer any questions, would be Jim. However, suffice to say, the simulator will run on the TI, as well as the Geneve. As I understand it, the higher clock speed of the 9918 video IC on the Adam, results in a slower execution of the ROM dump on the 4A. The 0-WAIT Geneve, on the other hand, allows the software, Antarctic Adventure, to run closer to the way it would on the Adam. Gary Bowser has made, ala Proto-board, a device to enable ROM dumps of Adam cartridges. It seems that those with 16K, two pages of memory, will run o.k. in a regular Geneve. Cartridges that use more,(32K), will reqiure a Geneve with expanded memory.

At April's meeting, I brought my Geneve, and we had the penguin strutting his stuff in the Z80 Simulator. As to whether this software will be released as freeware, commercial software or whatever, has still to be determined. Date of release, also, is to be announced. Stay tuned.
##################EoF##################

PRBASEOLOGY

by Stan Corbin

Some find PRBASE confusing - as I did, however when the light comes through it is found to be a very fine and useful program - for those who have use for a data base.

I will try to take you from start up to shut down in a series of steps.
It is anticipated that you have the documentation for PRBASE to refer to, throughout this exercize.

DATABASE CREATION
First of course is to place your PRBASE program into your disk drive #1. You will need an extended basic module in the grom port. Turn your system on, press number 2 twice and the program will auto-load provided you have a program called LOAD on the PRBASE disk. When the program has loaded, a menu screen will be displayed, as follows.

PRESS:
 1 FOR DATABASE CREATION
 2 DATA MANAGEMENT
 3 EXIT

Number 3 will exit back to the TI title screen.
Number 2 is for managing the data on an already created data disk.
Since we haven't produced a data disk format we will press number 1.
This brings up the PRBASE logo screen of version 2.Ø, requesting a $1Ø.ØØ donation to William Warren, which is well worth it to anyone using the program. Hitting the space bar takes you to the ---
CREATION MENU
PRESS:

1 to Select Data Disk Drive
2 Format Data Diskette
3 Design Data Screen
4 Design Tabular Reports
5 Design Mailing Labels
6 Set Printer Codes
7 Set System Options
8 Exit

PRESS NUMBER 1 FIRST, a subscreen will be introduced, with the legend "Enter Data Disk Drive (1-5) 1". Pressing number 2 will display a 2 and the subscreen will disappear. You have now selected the drive into which your data will be input. Install a blank initialized diskette into drive #2.

Failing to select the disk drive for your data diskette at this time, results in not being able to save your formatted screen to disk. Now select item #3, to design your data screen. The Screen you are now seeing with numbers Ø through 9 quadruplicated across the top of the screen, and a series of graphic characters at the bottom is where you will set up your particular format. We may as well follow the format in the PRBASE DOCUMENTATION. Your screen can be personalized by framing your title within a framework of the graphic figures obtained by using CONTROL A,B,C or whatever pleases you, from the graphic character display. When you are satisfied with the title display, you can proceed with setting up the rest of your data format.

Pay particular attention to the special key functions listed under "DESIGNING YOUR SCREEN" in the PRBASE documentation (page 6). You will find Function E and X and ENTER to be very useful in moving around the screen.

Type in the format as shown in the documentation. The brackets shown, are obtained by typing R,T,F, or G, while holding down the function key. These brackets are used to define the length of the fields. That is, how many characters can be placed in each field such as the name field which has 17 spaces allocated. The brackets produced with Funtion R and T are used where you want the item in upper case (CAPITAL) letters. The brackets ARE included in the field size. Once you have completed the formatted screen, press Function and 6 (PROC'D) and if the total of spaces used in your fields do not exceed 255 character spaces, you will be asked for your data base filename, type in a name which is not more than ten characters in length. For this exercize let us use "PERSFILE" for the filename. Press ENTER and you will then be asked for your output device. This could be PIO, or RS232(include baud rate etc.). After entering your output device, you will be asked "print this screen (Y/N)", your answer is "Y" as you will want that information later. Printer must be ON! The screen will then be printed by the printer. You will then be told to "press PROCD to write to Disk". The number of fields you have used and number of characters used will be shown. IF you have used more than the allotted 255 spaces for your fields, you will be so informed and you will then have to go back and adjust your fields so that the total character spaces used do not exceed 255 (does not occur with this screen). Pressing PROC'D (function-6) will print the data to your Disk #2, and on completion will return you to the CREATION MENU.

Now you may want to design mailing labels (for an address file), in which case you would select item "5 Designing Mailing Labels". You are then asked "Number of Lines per Label?". If you are using the standard 1 x 3.5 labels, you would enter 6. Now you are presented with a screen indicating the starting location and the size of each field in your address format. You are asked for your "Log Device:". Type in PIO or RS232(baud etc.) and ENTER. If the program does not accept PIO and reverts back to the same screen, try a Function "3" (ERASE), and then type in PIO. You are now asked "Print This Screen (Y/N)"; type Y, and the screen will be dumped to the printer. Another screen is presented and you are asked the same question and you type "Y".

A new screen is facing you, titled, "DESIGN MAILING LABELS". Here is the nitty-gritty! Item number is the number position of your fields, ie. (1)LAST NAME, (2)FIRST NAME, etc. Screen location is the starting position of each field. Number of characters is the number of character spaces available in each field. Report line is the line on which the field (name, address etc.)will be printed on your label. Column position is the position on the printed line where each field starts.

The cursor position is at the start position (bracket) of LAST NAME.

Refer to the first screen you printed out and read the number at the far left, add the number from the column at which the first bracket is located. This will give you the number shown, which is the number 213.

If you look under the title (page 1Ø of PRBASE documentation) "A DISCUSSION ON THE REPORT VECTOR TABLE", you will note, that the screen shown, puts the LAST NAME starting bracket position at 255, that is incorrect for the original screens used. If you copied the design, illustrated under, "DESIGNING YOUR DATA SCREEN", then the position of the bracket will be 213. The number of characters would be 17, (columns 13 through column 29, the ending bracket).

The printer would print the Last Name at the START of the first line of the label. The first name would be printed LAST. To print the first name first, one would INVERT column positions. Item #1 would be in column position 21 and item #2 would be in column position 1.
As you can see, the line an item is written to, OR to which column, is accomplished by manipulating the line and/or column positions. Bear in mind that one should start at column position 1 and add the number of characters for the item plus two (for leeway) to get the position for the next item appearing on that same line. For instance, if item 1 has 17 characters, the position

From RDM UG, California 6/88

TI-dings 11/88 p. 9

for item #2 on the same line would be 19 (17+2).

Now let us proceed to set up our labels. let's print the FIRST NAME FIRST, so item 1 (Last Name), COLUMN POSITION will be 24 (22+2). The first name uses 22 characters and is being placed before the last name. Item 2 (First Name), COLUMN POSITION will be 1. We want the street address to be on the second line all by itself, so change item #3, REPORT LINE to 2. City, State and Zip should be on line three. Change item #4 (City), REPORT LINE to 3, and COLUMN POSITION to 1. Change item #5 (State), REPORT LINE to 3, COLUMN POSITION to 19 (17+2). Change item #6 (Zip), to REPORT LINE 3, and COLUMN POSITION 24 (19+3+2). NOTE: When an item is not on position 1 (start of printed line), it becomes necessary to add the number of characters that preceed it. In item #6 we added 17 the number of characters in "CITY" plus 2 for leeway which gave us 19. We then add the 19 to the 3 spaces allotted for "STATE" plus 2 for leeway. That put the start of the zip code at 24.

We have completed the information for printing our labels, however in order to keep the next lines from being printed, let us replace the "Screen Location" numbers for items 7, 8 and 9, with zero's. Do the same for "No of Chars".

You are now ready to follow the instructions at the bottom of the screen. Press Function "6" (PROC'D). You are informed (at bottom) "Standby-initializing data screens. Once initializing is complete your screen will show "Number of lines used: 3", "Number of lines wanted: 6".

Since this is what we want, press ENTER and your label format will be written to disk and you will be returned to the main menu.

Select number 7 SYSTEM OPTIONS. Your file name and printer default (PIO) will be shown. Press ENTER for each of these. Next is "Number of sides on data disk, enter the appropriate number for your system (usually 1). This program allows the use of two across labels, most of us use one across. For left label starting column the options are Ø to 5. The choice is yours, however I use zero as you are given the same choice in the PRBASE Management Program. If you are using one across labels then Zero is appropriate for the right label. After completing these entries you are returned to the main menu. Now you can select "8" to exit, you have designed your Data Screen and Labels. You are back to the TI title screen.

DATA MANAGEMENT

Press 2 twice to load PRBASE and select option 2 (DATA MANAGEMENT) from the PRBASE title screen. After it loads, you will be asked for the disk number and the name of your data disk. Enter that data (DSK2.PERSFILE) and your formatted screen will be displayed. Press H for HELP and all the command letters will be displayed. These are the same commands listed and explained in the PRBASE documentation. Pressing any key will return you to the command mode. Unless you ARE IN the command mode, there will be no response to pressing any command letter (other than returning you to command mode).

Since our data base is empty input the command "A" for adding data and input the appropriate data. Do this for each data screen. When you have finished, select the command "I" for index and select the field you want indexed. Let us use the first field (last name) and press FUNCTION "6"(PROC'D). Select the command "S" for sort, the computer will sort the first field. Now if you select the command "U", and type in the string you want to find in that first field, and press ENTER the name will come up immediately, unless you typed a name that is not in that field.

This is a good time to select the command "O" (drive output) and input the DRIVE number and PIO. You will be asked "SELECTIVE INDEX?: (Y/N)" the place to learn about this feature is under the title "BUILD MEMORY INDEX". Normally your reply to this would be "N".

The "G" or global search will find any string that is in any field.

To print labels it is necessary that your printer is on and the indexed field has been SORTED. Press "L" to print labels, the legend "Start at: #", followed by a flashing cursor appears. You can enter the screen number you wish to start with. The labels will be in alphabetic order and NOT numerical. In other words when you select a starting screen number the next screen will be by alphabetical order and not necessarily the next numerical screen.

One thing I have not mentioned is control codes (command "C") . This program allows one to print the labels in the type font you select. In the CREATION menu there is item #6 "Set Printer Codes". I did not mention it when we were in "CREATE", as I felt it was not necessary at that time. You have five lines to format with the printer codes you may wish to use for your labels. There is a sixth line available from the DATA MANAGER. Pressing "C" from the command mode displys "CODE CONTROL SELECTION" with choices of zero to 5. One to five have to be installed in the "CREATE" program. Selecting Zero (Ø) will bring up "Control Code ASCII Values" and below that, a line of six zero's with spaces between them. You would refer to your printer manual for the ASCII control code characters to use fo, whatever printer features you desire. The first character is usually the escape character 27. For printing in double width mode (Gemini 1ØX), input 27 and ENTER, then 87 and Enter, then 1 and Enter, then Enter. When you have a zero under the cursor and press ENTER the control codes are read to the printer and the printer is ready to print in your selected style. NOTE: your printer must be on and attached.
I believe you now have sufficient information to use PRBASE. Read the PRBASE documentation carefully in conjunction with this information and you should have no problems.

SCR #57

```
 Ø \ BASES and BINARY 5/7/88 EGR
 1 : B  2 BASE ! ; \ BINARY
 2 : O  8 BASE ! ; \ OCTAL
 3 : D 1Ø BASE ! ; \ DECIMAL
 4 : H 16 BASE ! ; \ HEXADECIMAL
 5 \ 111111
 6 \ 54321Ø987654321Ø BIT NUMBERS
 7 \ 31                 W V\
 8 \ 268421           B E A \ POWERS
 9 \ 731ØØ521         I I L >  OF
1Ø \ 689942152631     T G U /  TWO
11 \ 8426842684268421   H E/
12                      T
13 B 1111111111111111 \ D -1
14 B 1ØØØØØØØØØØØØØØØ \ D -32768
15 B Ø111111111111111 \ D  32767
```

TRITON, TENEX AND TEX-COMP

At the May meeting, we promised to publish their phone numbers:

TRITON	1-8ØØ-227-69ØØ
TEXEX	1-8ØØ-348-2778
TEX-COMP	1-818-366-6631

USING THE TI
By Stephen Nickerson

This is an attempt to give club members some insight as to how I am using my TI at work.

To start the story one neede to realize that untillrecently I didn't own a disk drive,thus all my work was done with a console and cassette. I carried the console and cassette player to and from work because my parallel interface and printer were at home. It was too much trouble to carry the printer to work. I also use the printer for other computers at home.

After developing an interest in computers and discovering word processing (Data Biotics Wordwriter Module with word wrap!!!!) I have spent a lot of effort trying to have the County buy me a computer for park work. They have of late purchased a Panasonic KX-P1091i printer (will report on at a later time) for me; just this has helped a lot. In the past I would write out my memos and forms at work, save to cassette, take home, print it out, and refine it. This has meant a lot of lost time and homwork to get things just right.

Now using a disk drive and the printer at work (still using the Wordwriter mod_le) things have changed to say the least.

I have started a collection of memo spacing forms, now I just type over old work, save new, print, I'm done, and no more working out the spacing each time. Need a change in an an old memo or letter? Call it up, make changes, print, and it's done. No more having to take it home and bring it back to make changes.

I have used TIcalc to work out park budget' needs for seasonal park employees and rework the numbers after the final budget appropriation. Once set up a lot of time is saved and a nice professional looking print out is obtained on just this application.

This year I used the Personal Record Keeping module to keep a list of 273 names and location of park employees as I was the Park Dept.'s campaign coordinator for the United Way. After making up the files and typing in all the names, I realized if I had changed a couple of the fields to integer (number fields) I would have saved myself some time as the module could have done several math calculations for me, but it still helped to keep track of many things.

(Editor's note: About two and a half years ago, we had a program on how to modify PRK files using the seven additional BASIC language commands that are hidden in the PRK module. It is actually possible for Steve to change the fields without re-entering all his data. In my opinion, aside from its severe '::K limit (all data must fit into VL: RAM) and the slowness of its sort routine, the PRK module is an excellent data base. AJB)

Now with the disk drive I have started to play around with PR Base a data base from our library. This is an open screen and you put the files in were you want. It's not as structured as PRK. First impression is it's good , but lacks math computations. When I am more familiar with this data base, I hope to give a report at a future meeting.

****************EoF*****************

SPAD XIII Mk 2
By Harry Wilhelm, West NY 99'ers

This is a review of the SPAD XIII Mk. 2 flight simulator published by Not-Polyoptics. (A strange name if ever I heard one!) According to the ads, this is an updated version of the SPAD XIII program that features better response, improved flight response, etc. Since I never owned Mk. i I will have to confine my remarks to the current version.

One's first response to any program is their reaction to the owners manual. In this case, it was a highly favorable response. Not counting the covers, the manual is 20 informative and interesting pages. It covers the history of WW I fighter planes and the SPAD aircraft in particular. There is a brief description of how aircraft fly. Controlling the SPAD is covered in greatest detail, as you might expect. The objects you will see on the ground and in the air are described. First flight, aerobatics, your mission (which is to destroy "hunnish" aircraft and hangars) are all gone into. Five separate scenarios are described towards the end of the book. Throughout the manual, there are informative pictures demonstrating the stall, the immelman turn, the loop, etc. These help give the flavor of the era in which you will be flying. All instruction manuals should be this good' On a scale of one to ten, this manusl rates a 12!!

The disk is protected, and requires Extended BASIC to run. It will NOT run with the E/A or MiniMem modules. (I tried.) It autoloads quickly from Extended BASIC in about 20 seconds or so. The master title screen gives you the option of having a "Red Baron" adversary. Beginners are best advised to press "N" as the red baron is an aggressive and troublesome opponent even for the best of pilots. The screen then clears and is replaced with a forward view sitting in the cockpit of the SPAD. There are four instruments: the altimeter, the air speed indicator, the compass, and the fuel guage. There is also a throttle lever on the left side of the cockpit and your joystick in the middle and bottom of the screen. The nose of the plane is up, since the tail is resting on its skid, and you see that your machine guns and gunsight are aimed at the sky.

Pressing the keys from 1 to 6 gives all s(x possible views out of the plane: 2 looks to the left, 3 looks to the right, 4 looks back, 5 looks up, and 6 looks straight down. This last option is the only unrealistic one, since in the real plane you wouldn't be able to see out the bottom unless it was heavily damaged. All these views show the appropriate parts of the plane; ie. the wings, the rear fuselage with vertical fin and horizontal stabilizor, etc. Keys 7, 8, 9, and 0 control the engine speed. Key 9 will gun your Hispano-Suiza V8 to full throttle. The 0 key kills the throttle, and the 8 and 7 keys move the throttle up or down a notch.

Press 9 to gun the engine and your plane commences on its takeoff roll, gathers speed, and becomes airborn. You can be sure that the real plane was a lot trickier in it's ground handling characteristics, since it was a tail-dragger with only a skid (not even a wheel) under the tail. You don't have to do anything except maybe pull some up elevator after takeoff to start climbing. You can control the aircraft with either joysticks or else with the keyboard. I recommend the latter, since you have more control over the plane. You can get full left, right, up or down controls from the keyboard, as well as rudder control. These functions are very valuable in a dogfight!

You have full control over the aircraft and that includes all the common aerobatic maneuvers such as loops, barrel rolls, split S turns, and immelman turns, and inverted flight. You see the horizon tilt. If you make a right hand turn over one of the towns, you can look to the right and see the town pivot underneath your wings. When making a landing approach you see your field loom ever larger as you approach it in your descent. Meanwhile your altimeter unwinds, and if everything is set up right it arrives at zero at the same time that you arrive at your field.

At this point I should point out that this program is one of the strangest that I have ever seen. Strange because it is at the same tiee very crude and very elegant. Crude because of the graphics which are highly simplified. For example, clouds are represented by white diagonal lines in the sky. The german observation balloons are represented by strange polyhedral shapes in the sky near the trenches. The ground is the same color as the sky (cyan) but with the addition of some sparse vegetation such as hedgerows and an occasional tree. The horizon is a dark green line with a back of white clouds over it. This, and the way the plane handles, is your only key to whether or not you are inverted.

At the same time the program is extremely elegant. Everything is represented 3 dimensionally. Nearby clouds pass through your field of view more rapidly than distant ones. Objects get larger as they get closer. All the fundamental laws of aerodynamics seem to be obeyed by the plane. Changing your engine speed primarily controls whether you climb or descend, while the elevator control is what actually determines your airspeed. When the plane stalls it often goes into a spin, which takes some practice to recover from. Even the enemy planes that you will encounter seem to be under intelligent control, although they appear to be less maneuverable than your own. You can shoot down an enemy plane and immediately go into a vertical dive. Descend a thousand feet or so (don't hit the ground), then level out and look behind you. You'll see the flaming wreckage of your victim falling out of the sky and through your field of view' When you shoot down an observation balloon it bursts into flame and slowly descends to the ground. Around the observation balloons and the enemy field is plenty of anti-aircraft fire. It seems to be mostly a matter of luck whether you survive it or not. One of your missions is to bomb the two german hangars, which takes a great deal of practice to do reliably. Another realistic point is that you can cross the rudder and aileron controls and do a sideslip when making your final landing approach.

I keep finding new subtleties in the program. For example, sometimes when your plane is hit it will be damaged in the engine. You can restore partial power by holding full throttle, which causes the engine to put out about half it's usual amount of power, and laboring and sputter markedly. The subtlety is that you are also hit in the fuel tank when this happens. Your fuel guage comes around to empty MUCH faster than it would ordinarily and you have to head back for your field as soon as possible before you run out.

Most programs that I use seem to be sufficiently straightforward that I could picture myself writing them if given enough time and motivation. But SPAD XIII seems so complex that I couldn't even imagine where to begin'

In case it isn't apparent, for the most part I really like this program. The proof of that is that I have spent more time with it than with any other game type program that I have. I would recommend this program to anyone who has any interest in flight and is willing to learn a l(ttle bit about how planes fly. - HWW

*******************EOF*********************

TI-dings P. 11 11/88

FROM CW UG

THE US ROBOTICS SPORTSTER 2400 MODEM
By Carney W. Mimms

Those of you who are old-timers in the telecommunications field (in other words, you've been on-line for more than a year), will have already recognized that 2400 baud is well on its way to becoming the new standard for everyday microcomputer telecommunication, just as the 1200 bd standard replaced 300 baud not so long ago. One happy result of this transformation is the explosion of low-cost 2400 baud modems now reaching the market. You could of course choose on price alone, but as we all know, many of the companies in the peripherals business have about the same half-life as a microgram of einsteinium. This can be particularly distressing in the case of modems, since there is basically no way to assess the quality or reliability other than to. rely on the reputation of the manufacturer. And the list of reliable manufacturers is a short one. There is, of course, Hayes. the company that created and defined the "smart" modem and still the maker of top-flight modems at top-flight prices. At the same level of price and quality is US Robotics, whose Courier 2400 helped define the 2400 baud modem. The Courier 2400 baud modem is a solid, durable device. but at $450.00, it is not within the budget of most of us.

Fortunately, US Robotics has recently decided to change this situation by offering a low-price version of the Courier, the Sportster 2400. This model, which lists for $250.00 and can be found by mail order for less, uses the same chip set as its more expensive brother. For this reason, it supports the entire Hayes command set, as well as US Robotics' highly- regarded proprietary error-correction protocol, called MNP. To produce the Sportster at a lower cost, US Robotics has changed to a smaller (about the size of a paperback book) case, cheaper switches, a cheaper speaker, and included a much less comprehensive, but still adequate, manual. By way of compensation for the reduced manual, however, the complete command set has been placed on a label on the bottom of the modem--much tougher to misplace. All in all, the Sportster represents an outstanding combination of quality and value, backed by one of the few stable big name companies in the modem business.

We can only hope that its presence will raise the standards of the low-priced 2400 baud modem market generally so we can all start uploading and downloading at 2400

I had the pleasure of using the Corcomp 512 Ramdisk for the month of April. I was very impressed with its operation and ease of use. The first thing I did was to load up Funnelweb and try its functi ons. I was surprised when I tried to use DM1000 to transfer files to the ram disk. I was able to copy the programs I wanted to the ramdisk, which I design ated as DSK1. When I did a listing of the files on DSK1., every file showed up twice. When I saw it was an error, I unplugged the Ramdisk and started over. I used the Ramanager to copy the programs I wanted and it worked fine.

When I restarted my system, the Ramdisk would not start up Funnelweb. I then went through the configuration set up and turned off boot tracking and the program worked great. I loaded up the Ramdisk with all the games I could find and let my two sons (7 & 9) to use the TI as much as they could. They were excited with the ease of operation and the fact that they could just choose a game and it came up real fast. The only problem they had was with the shut off switch on card. Since I only had the card on Ioan for one month, I did not wire the button to a remote switch. As a result, they just turned off the machine without reset After a few days of use, programs were starting to be corrupted by this misuse. It appeared that EA5 type games were most affected by this. This may be due to the fact that these games were the ones most used and therefore most likely to be corrupted.

I was very pleased with the card's overall operation. The only limiting factor I really encountered was the amount of programs I could cram onto the Ramdisk. This was no real problem, because I could put my most used programs on it with room to spare. I would rate this product a B Bob Cataldo.

WHAT'S NEW??

Two new data bases for the TI 99/4A.

TI-BASE - the following is a PRESS RELEASE, not a review.

TI-BASE. New from INSCEBOT, Inc.

TI-BASE is the place to collect and organize your data. Record Definitions are customized to your exact needs. Data can be freely interchanged between all types of records. Simple structured command language allows an easy implementation of needed functions. Address lists, checkbooks, business applications, all easily performed and maintained on diskette by TI-BASE.

DATA DEFINITION:	MATH CAPABILITY:
Character	Arithmetic
Numerical	Trig
Date	Logical
Literals	Boolean
DISK DIRECTIVES	SORTED RECORDS:
Initialization	Nested command files
Catalog	Structured Directive
File Copy	
File Deletion	

SORTED RECORDS:
Sequential
$Finds$

LOCAL VARIABLES, ON LINE HELP, CHARACTER MANIPULATIONS

$24.95. Florida Residents add 6%. $1.50 postage and handling. Quantity prices on request.

INSCEBOT, Inc. P O Box 291610, Port Orange, FL 32029

Requires 32k memory expansion, a minimum of one disk drive and either Extended Basic, Editor Assembler of Mini/memory modules for tha TI-99/4A

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

FIRSTBASE - Warren Agee's long worked upon and long awaited data base. The following is from a publicity release and is NOT a review:

-IBM style query commands

-Batch processing with four-function floating point math

-Multipie keyword searches

-Extremely large capacity

-Handy Macros

With FIRSTBASE, the new data base manager from Olympys Technologies, you can organize, manipulate, and report data like never before. We listened to what users missed in other database managers, and what we've created is quite impressive. We increased record size to 3000 bytes, allowing you to store much more information than ever before. We allowed you to search for data using powerful query commands, similar to ones used on IBM compatible programs, so you can find exactly what you are looking for with a single command. We added a batch-processing capability, so you can perform "search and replace" functions or delete a group of records at once. You can ever perform MATH on any field. We added a MACRO feature so you can store frequently used commands on disk for quick recall.

SPECIFICATIONS

720 bytes/fields
3000 bytes/record
75 fields/record
32,767 records/databases
100 megabytes/file

-Search on multiple fields using AND and OR, or on keywords

-Search on multiple keys

-Quiries sent to screen, printer, new database, existing database

-Flexible report generator

-Written in c99

-Requires 32k, disk and either XB, E/A, or TI-WRITER Cartridge.

-Myarc Geneve 9640 Compatable

SUGGESTED RETAIL PRICE: $49.95

Available exclusively through GENIAL COMPUTERWARE, P.O. Box IB3, Grafton Ma. 01519

Developed by Olympys Technologies, Advancing software technologies.

Scheduled release date: Late June or early July, 1988.

FIRSTBASE written by Warren Agee. c99 language by Clint Pulley.
$$$$$$$$$$$$$$$$$$$EOF$$$$$$$$$$$$$$$$$
$$

From CW UG

TI-dings p12 11/88