THE WORLD RETURNS TO LOS ANGELES

99'FEST-WEST'87
MAY 2-3, 1987

TopIcs

LA 99ers COMPUTER GROUP

Newsletter

VOL. 6 NO. 1 JAN. 1987

Los Angeles, California

TERRIES'S CORNER

Teresa Masters President

SOMETHING OLD, SOMETHING NEW

F. Y. I. This column and this Newsletter is NOT copyright L.A. 99ers or me. It is for our members and the community at large. Feel free to use anything of value contained within, we only ask you credit the source.

Old Trash, RNK otherwise known as my infamous Runny Nose Kid comment. Gremlin has struck again. Rather than use a recognized talent, immaturity and irresponsibility prevailed. How sad, how predictable.

Thanks to Art Byers for his public recognition of "TopIcs" as the #1 Newsletter, and Tom Freeman as the most prolific contributor. George Steffen also deserves this accolade. This Newsletter is really the efforts of many, the credit goes to all.

Example of this last sentence. Last night Jan. 21, George, and I met at Tom's home. We were trying to get this edition out in a timely manner. We were three spastics. Tom getting over the loss of his Father, and multiple breakins of his home. George recovering from his Cancer surgery and now about to start another course of Radiation, and me, my Mother with 3 emergency room trips and 2 hospitalizations burning up the Ventura Freeway at 1 AM, for the last 2 weeks. Well we tried, but sorry we are a bit late. My hospital runs will determine just how late.

Elections, this is the meeting. Do your part, be there and participate. We still need a Secretary. How about a spouse or sibling or even you.

About prolific contributions. Stephen Shaw deserves a BIG thanks. His material alone is enough for 3 months. Many thanks Stephen, wish we could print it all.

OPTIONS Contained within is equal time for both MYARC AND TRITON (MG's MS/DOS interface) Both making their awaited appearance in the first quarter of '87. We really have 3 options the above mentioned 2 plus the 99/4A. Your needs alone will determine the one you select. The bileous venomous pontifications, well just consider the sources. Truly one of the Seven Sins.

L.A. 99ER'S JANUARY PROGRAM

The January meeting will begin with the President's report by Terrie Masters. Fred Moore, the club software librarian, will discuss the new additions to the collection. Following Fred's report, Gail Fair will report on the month's Marketplace specials.

The nominating committee will present its proposed slate of officers for consideration by the membership and nominations will be taken from the floor. Standing for election as President will be Tom Freeman, Vice-president Terrie Masters , Treasurer Alan Whiteman. Unfortunately no one agreed to stand for Secretary. Can anyone out there help?

Karl Reed has been using the Nararone Data Base management software in his business. He will carry on the series, "How I use my TI 99/4A Computer." at the January meeting and give us an overview of this database manager. In conjunction with this, Terrie Wilson will present a timely income tax package for Multiplan, a fairware program by James Sleeth. This talk will be followed by the social break.

The feature of the January meeting will be the second of a three part series on telecommunications. This month communications between two computers at the meeting will be demonstrated. In fact hands on participation is encouraged. Remember computers are not a spectator sport. The best way to learn how to communicate between two computers is to try. This demonstration will be conducted by Steve Chalcraft.

The January meeting will be hosted by Fletcher Wicker.

CALL LOAD TO ASSEMBLY AND BACK

by Tom Freeman

This article and the programs that accompany it are another in my intermittent series to help those interested in understanding assembly programs better. You will find XBasic programs that will convert assembly language programs in various formats to other formats, which might have one of two purposes. Either you wish to increase the portability, or printability, of a program, or you wish to disassemble it to understand what the programmer was doing.

Many programs that use assembly subprograms are published in a "CALL LOAD" format. In other words, the XBasic program directly "pokes" the assembly program into memory, byte by byte. This is done because it might be cumbersome to type in the source code for the assembly program and then assemble it, or you might not have the Editor/Assembler (everyone should, however!). Nevertheless I have published most of my programs this way. The author might also publish the (uncompressed) DIS/FIX 80 object file, but if you have ever looked at one of these, each line is just a long string of numbers and letters that make no sense, and it would be almost impossible to avoid a typing mistake! The CALL LOAD's, on the other hand, are full of commas and easily read numbers, so typing them in is easier. However that portion of the program must be run every time the program is run, which takes extra time, so it would be nice to be able to convert them to a real assembly file. Two recent examples of programs that use this method are: "Artist to XB" in Smart Programmer, September 1986 - contains two columns packed with CALL LOAD's, and Improved Unrunnable Basic in TopIcs, September 1986.

The first program, entitled CL/ASL, that follows this article (I have placed all the programs together, for neatness' sake, so that they could be in 28 column format which looks EXACTLY the way you type it in) provides a method of turning a CALL LOAD XBasic program into either a source code file, which can be run through the Assembler to produce an object file, or an object file directly. Thus there are really two programs here - lines 190-280 could be deleted if you only want to make source code, or 290-350 for only object code. I haven't been able to test this program on LOTS of files, so I suggest you use them both, in case one produces errors. Naturally I have tried to account for all the errors I could think of! One that cropped up was when the CALL LOAD began with an odd address. Assembly files normally insist on even addresses. I compensated for this by backing up to the even address one lower and beginning with the last byte from the previous line. Try this out with a sample two or three line file to see what I mean. The assembler automatically backs up one byte if the AORG or RORG address is odd, and inserts a zero byte first. This would mess up the code, which is why I retained the previous byte.

The only constraints on the input file are that it must 1) be saved in merge format (DIS/VAR 163) not as a program file, 2) contain only CALL LOAD's (delete all other lines and any other statements on the CALL LOAD lines before saving) and 3) only one CALL LOAD(address,byte,byte,...) per line. The program makes heavy use of a knowledge of how the program lines are tokenized. You can see this for yourself by running the last program in this article on a sample small file and comparing the bytes generated with the list of tokens also provided.

I found one interesting quirk in the way TI handles these assembly DIS/FIX 80 files. Normally the author of a CALL LOAD type program needs to set the REF/DEF table just below 16384 (hex >4000) byte by byte, and then insert the address of the beginning of the table into 8196 (>2004). I originally tried to do this just with AORGs, but the XB loader just won't insert the bytes there even if the assembly file tells it to! CALL LOAD works fine however. I fixed this up by assuming that all code above 16225 is for the REF/DEF table (this leaves room for 20 DEF's, and it appears that no one ever has actual assembly code at this location) and then actually construct a real DEF table. Then the loader sets the proper address into >2004 by itself.

Now when the file is ready you can replace ALL the CALL LOAD's by CALL LOAD("DSK1.YOURFILE") where YOURFILE is whatever you named your DIS/FIX 80 file (produced directly by my program, or assembled from the source code it produced). By the way, I lied a little when I wrote above that the assembly program needs to be reloaded every time you RUN the XB program. When a program is finished, the assembly code remains in memory unless you quit or CALL INIT again. So you can add a line to any such program that "PEEK"s at a couple of bytes that you know the value of (do the peeking after the program is run the first time) and then bypass the CALL INIT and the CALL LOAD if the bytes are what they should be. This works with either method of loading the assembly file (CALL LOAD(dis/fix 80 file) or CALL LOAD(address,bytes).

By the way, the program takes quite a bit of time to run, especially if the CALL LOAD's are numerous, but at least it only has to be done once!

The second program, entitled ASL/CL, reverses the process. Why would you want to do this? There are two possible reasons: one might be that you have an XB program and wish to publish it, or list it for a friend. Putting the assembly code into CALL LOAD format makes it all readable in one program. Another reason could be that you wish to have the program on tape for someone who has memory expansion but not a disk drive (my son was originally in this position). The program as listed also is a "double" program, as it allows you to construct the CALL LOAD file from a memory range, or directly from a DIS/FIX 80 file. Most object files can be simply loaded from command mode by CALL INIT :: CALL LOAD("DSK1.XXX") and my program then run with the memory range option. [This part of the program runs considerably faster.] WARNING - a few files insert the start address into the ISR hook at >83C4, and will thus auto start. You will need to run the program on the DIS/FIX file directly or use a sector editor to change that value (you would find at the end of the file something like 983C4BXXXX where XXXX is the Start address - it should be changed to 0000). Please note that the program ends with a statement on the SCREEN that you should type in one or two extra CALL LOAD's. I could have had the program do this, but I didn't get around to it and time is short! [Please note that if the program does use the above auto-start method, then you will need to add one additional CALL LOAD(-31804,x,y) where x and y are the decimal representations of the two bytes following 983C4B above, e.g. if you saw 24F4 than x and y would be 36 and 244]

If you are going to use the memory range option after loading the DIS/FIX file, there is an additional program that will help you, called ORIGINS. Many object files do not load all the bytes in the whole range of address used, but instead leave some blank, to be used later by the program (this is signalled by the BSS directive in the source code). ORIGINS will search for these breaks - actually it just lists all the origins, and you can see if there are large gaps as normally a single DIS/FIX record can only contain about 22 bytes. You can then specify each memory range separately in the program and not wasted a lot of CALL LOAD's unnecessarily.

There is one additional type of assembly file that I haven't mentioned. Some authors have written assembly code, and then "hidden" it in the XB file, using various methods such as Barry Traver's ALSAVE program. You should suspect this when the XB program as listed has more sectors than could be accounted for by the number of lines you see, or if you see a CALL LINK or a CALL LOAD(-31804,x,y) when no assembly file was loaded. The program called HIDDEN will search for the area containing the assembly file and inform you of the range. If you save the ASL/CL program in merge format, and then merge it into the hidden program, you can specify the memory range and produce a CALL LOAD file. I must warn you however, many of these are quite long and would produce a gigantic CALL LOAD file! You would probably be better off in that case to use SAVE to produce a separate program image file and then DISkASSEMBLE it! (See my article in November 1986 TopIcs to see how to use SAVE).

Finally the last program is called PRINTMERGE. This will take a MERGE type file and produce a neat listing in compressed format on a printer of each byte of each line and the ASCII representation if possible underneath it. You can do this on a few lines of code to see how program lines are tokenized. If you run it on a single CALL LOAD line for instance you would find the following: the first two bytes represent the line number (multiply the first by 256 and add the 2nd). The rest of the bytes are the tokens, or strings, and the last is always a Ø. After the line number bytes you will find: 157 (CALL), 2ØØ (unquoted string), 4 (length of next string), 76, 79, 65, 68 (L, O, A, D) 183 (left parenthesis) 2ØØ (unquoted string) x (the length of the string) x x x (the actual string, in this case the address to be loaded) etc. Have fun with this one, but DON'T use it on large files, unless you have lots of paper!

Ultimately my purpose in writing these programs was to be able to disassemble the CALL LOAD's to understand them. What I did was to produce files that DISkASSEMBLER could read. Reversing the process merely became a challenge! Here's hoping you find these programs useful. Enjoy.

Tom Freeman

CL/ASL

```
1ØØ !CONVERT CALL LOADS TO A
SSEMBLY SOURCE OR OBJECT FIL
E, BY TOM FREEMAN, LA 99ERs
11Ø BYTE$=" BYTE " :: H$="Ø1
23456789ABCDEF" :: END$="ØFF
FFF" :: ST$="ØØ" :: VALUE$="
Ø"
12Ø DISPLAY AT(2,1)ERASE ALL
:"NAME OF INPUT MERGE FILE?
 DSK1." :: ACCEPT AT(3,1)SI
ZE(-15)BEEP:I$ :: OPEN #1:I$
,VARIABLE 163,INPUT
13Ø DISPLAY AT(4,1):"PRODUCE
 OBJECT OR SOURCE      CODE?
(O/S) S" :: ACCEPT AT(5,14)S
IZE(-1)VALIDATE("OS")BEEP:CO
DE$ :: IF CODE$="O" THEN 15Ø
14Ø DISPLAY AT(7,1):"NAME FO
R OUTPUT SOURCE FILE?DSK2."
:: ACCEPT AT(8,1)SIZE(-15)BE
EP:O$ :: OPEN #2:O$ :: GOTO
16Ø
15Ø DISPLAY AT(7,1):"NAME FO
R OUTPUT OBJECT FILE?DSK2."
:: ACCEPT AT(8,1)SIZE(-15)BE
EP:O$ :: OPEN #2:O$,FIXED
16Ø DISPLAY AT(1Ø,1)BEEP:"IN
PUT FILE MUST CONTAIN ONLY'C
ALL LOAD'S, AND ONLY ONE  PE
R PROGRAM LINE"
17Ø DISPLAY AT(14,1):"RELOCA
TABLE/ABSOLUTE?(R/A) R" :: A
CCEPT AT(14,28)SIZE(-1)VALID
ATE("RA")BEEP:R$ :: IF R$="R
" THEN RFLAG=1
18Ø IF CODE$="S" THEN 29Ø
19Ø PRINT #2
2ØØ LINPUT #1:C$ :: IF LEN(C
$)=2 THEN PRINT #2:":" :: GO
TO 28Ø
21Ø GOSUB 36Ø :: GOSUB 38Ø :
: RCODE=MAX(RCODE,Q*N)
22Ø IF N>16225 THEN GOSUB 42
Ø :: GOTO 2ØØ
23Ø GOSUB 48Ø :: CALL HEX(AD
DR$,HEX$,H$,OFLAG):: PRINT #
2:CHR$(57+Q*8)&HEX$;
24Ø IF OFLAG THEN PRINT #2:"
B";ST$;:: BFLAG=1 ELSE BFLAG
=Ø
25Ø C$=SEG$(C$,L+4,155):: IF
 ASC(C$)=182 THEN IF BFLAG=1
 THEN PRINT #2:"ØØ";END$ ::
ST$=HEX$ :: GOTO 2ØØ ELSE PR
INT #2:END$ :: GOTO 2ØØ
26Ø L=ASC(SEG$(C$,3,1)):: VA
LUE$=SEG$(C$,4,L):: CALL HEX
1(VALUE$,HEX$,H$):: IF BFLAG
=Ø THEN PRINT #2:"B";:: BFLA
G=1 ELSE BFLAG=Ø
27Ø RCODE=RCODE+Q :: PRINT #
2:HEX$;:: GOTO 25Ø
28Ø RCODE$=STR$(RCODE):: CAL
L HEX(RCODE$,HEX$,H$,Ø):: RE
STORE #2 :: PRINT #2:"Ø";HEX
$;RPT$(" ",8);END$ :: CLOSE
#1 :: CLOSE #2 :: STOP
29Ø LINPUT #1:C$ :: IF LEN(C
$)=2 THEN PRINT #2:" END" ::
 CLOSE #1 :: CLOSE #2 :: STO
P
3ØØ GOSUB 36Ø :: GOSUB 38Ø
31Ø IF N>16225 THEN GOSUB 42
Ø :: GOTO 29Ø
32Ø GOSUB 48Ø :: PRINT #2:"
  ";CHR$(65+Q*17);"ORG ";N-O
FLAG
33Ø IF OFLAG THEN PRINT #2:B
YTE$;VALUE$
34Ø C$=SEG$(C$,L+4,155):: IF
 ASC(C$)=182 THEN 29Ø
35Ø L=ASC(SEG$(C$,3,1)):: VA
LUE$=SEG$(C$,4,L):: PRINT #2
:BYTE$;VALUE$ :: GOTO 34Ø
36Ø C$=SEG$(C$,1Ø,155):: IF
SEG$(C$,2,1)=CHR$(194)THEN M
FLAG=1 :: C$=SEG$(C$,2,155)E
LSE MFLAG=Ø
37Ø RETURN
38Ø L=ASC(SEG$(C$,3,1)):: N=
VAL(SEG$(C$,4,L))
39Ø Q=ABS(RFLAG*(MFLAG=Ø)*(N
>9459 AND N<16226)):: IF MFL
AG THEN N=-N
4ØØ N=N-Q*946Ø :: ADDR$=STR$
(N)
41Ø RETURN
42Ø NAME$=""
43Ø FOR X=1 TO 8 :: C$=SEG$(
C$,L+4,155):: L=ASC(SEG$(C$,
3,1)):: VALUE$=SEG$(C$,4,L):
: NAME$=NAME$&CHR$(VAL(VALUE
$)):: NEXT X
44Ø V1=ASC(SEG$(NAME$,7,1)):
: V2=ASC(SEG$(NAME$,8,1))::
NAME$=SEG$(NAME$,1,6):: N=V1
*256+V2 :: IF N>32767 THEN M
FLAG=1 ELSE MFLAG=Ø
45Ø GOSUB 39Ø :: CALL HEX(AD
DR$,HEX$,H$,Ø)
46Ø IF CODE$="O" THEN PRINT
#2:CHR$(54-Q);HEX$;NAME$;END
$ ELSE PRINT #2:" DEF ";NAME
$:NAME$;" EQU ";N+Q*946Ø
47Ø IF ASC(SEG$(C$,L+4,155))
=182 THEN RETURN ELSE 42Ø
48Ø IF INT(N/2)=N/2 THEN OFL
AG=Ø ELSE OFLAG=1
49Ø RETURN
5ØØ SUB HEX1(VALUE$,HEX$,H$)
51Ø N=VAL(VALUE$)
52Ø U=INT(N/16):: L=N-16*U
53Ø HEX$=SEG$(H$,U+1,1)&SEG$
(H$,L+1,1)
54Ø SUBEND
55Ø SUB HEX(ADDR$,HEX$,H$,OF
LAG)
56Ø HEX$=""
57Ø ADDR=VAL(ADDR$)-OFLAG ::
 ADDR=ADDR-65536*(ADDR<Ø)
58Ø FOR X=1 TO 4 :: P=16^(4-
X):: U=INT(ADDR/P):: ADDR=AD
DR-P*U :: HEX$=HEX$&SEG$(H$,
U+1,1):: NEXT X
59Ø SUBEND
```

ASL/CL

```
1ØØ !CONVERT MEMORY OR DIS/F
IX8Ø FILE TO MERGEABLE CALL
LOAD FILE, BY TOM FREEMAN, L
A 99ERS
11Ø OC$=CHR$(179)&CHR$(2ØØ):
: CC$=CHR$(182)&CHR$(Ø):: H$
="Ø123456789ABCDEF" :: A1,A2
=946Ø :: DEFADD=16384
12Ø DISPLAY AT(3,1)ERASE ALL
:"CONVERT TO ""CALL LOADS"""
:"          by Tom Freeman":"CH
OOSE":"  1. MEMORY RANGE":"
 2. DIS/FIX 8Ø FILE        1"
13Ø ACCEPT AT(7,28)SIZE(-1)V
ALIDATE("12")BEEP:CH$ :: IF
CH$="2" THEN 26Ø
14Ø GOSUB 56Ø
15Ø DISPLAY AT(11,1):"NEXT T
WO #'S MUST BE DECIMALLAST W
ILL BE included": :"FIRST AD
DRESS ";A1:"LAST  ADDRESS ";
A2
16Ø ACCEPT AT(14,15)SIZE(-6)
VALIDATE(DIGIT," -")BEEP:A1
:: A1=2*INT(A1/2)
17Ø ACCEPT AT(15,15)SIZE(-6)
VALIDATE(DIGIT," -")BEEP:A2
18Ø DISPLAY AT(16,1):"CORREC
T? (Y/N) Y" :: ACCEPT AT(16,
16)SIZE(-1)VALIDATE("YN")BEE
P:Y$ :: IF Y$="N" THEN 15Ø
19Ø GOSUB 49Ø
2ØØ DISPLAY AT(24,1):"DO MOR
E IN THIS FILE?(Y/N) N"
21Ø ACCEPT AT(24,28)SIZE(-1)
VALIDATE("YN")BEEP:M$ :: IF
M$="Y" THEN 15Ø
22Ø DISPLAY AT(24,1):"# BYTE
S IN DEF TABLE?    Ø"
23Ø ACCEPT AT(24,26)SIZE(-3)
```

```
VALIDATE(DIGIT)BEEP:BY
240 IF BY THEN A2=16384 :: A
1=A2-BY :: GOSUB 490 :: DEFA
DD=16384-BY :: GOTO 540
250 PRINT #1:CHR$(255);CHR$(
255):: CLOSE #1 :: STOP
260 GOSUB 570 :: GOSUB 560
270 LINPUT #2:A$ :: RELOC$=S
EG$(A$,2,4):: CALL DEC(RELOC
$,RELOC):: A$=SEG$(A$,14,80)
280 T$=SEG$(A$,1,1):: IF T$=
":" THEN 530 ELSE P=POS(H$,T
$,1):: ON P-1 GOTO 290,290,3
00,300,310,320,330,330,340,3
50,360,370,5
290 CALL WARN1 :: A$=SEG$(A$
,6,80):: GOTO 280
300 CALL WARN2 :: A$=SEG$(A$
,12,80):: GOTO 280
310 RLFLAG=1 :: GOTO 390
320 RLFLAG=0 :: GOTO 390
330 GOSUB 480 :: LINPUT #2:A
$ :: GOTO 280
340 RLFLAG=0 :: GOTO 430
350 RLFLAG=1 :: GOTO 430
360 RLFLAG=0 :: GOTO 450
370 RLFLAG=1 :: GOTO 450
380 CALL WARN3
390 DEFADD=DEFADD-8 :: GOSUB
 480
400 LN=LN+10 :: CALL START(L
N,DEFADD):: GOSUB 460 :: GOS
UB 470
410 NAME$=SEG$(A$,6,6):: FOR
 X=1 TO 6 :: N=ASC(SEG$(NAME
$,X,1)):: N$=STR$(N):: PRINT
 #1:OC$;CHR$(LEN(N$));N$;::
NEXT X :: A$=SEG$(A$,12,80)
420 PRINT #1:OC$;CHR$(U);U$;
OC$;CHR$(L);L$;CC$ :: GOTO 2
80
430 GOSUB 460 :: LN=LN+10 ::
 GOSUB 480 :: CALL START(LN,
ADDR):: PFLAG=1
440 A$=SEG$(A$,6,80):: GOTO
280
450 GOSUB 460 :: GOSUB 470 :
: PRINT #1:OC$;CHR$(U);U$;OC
$;CHR$(L);L$;:: GOTO 440
460 ADDR$=SEG$(A$,2,4):: CAL
L DEC(ADDR$,ADDR):: ADDR=ADD
R+RLFLAG*9460 :: RETURN
470 ADDR=ADDR-65536*(ADDR<0)
:: U=INT(ADDR/256):: L=ADDR-
256*U :: U$=STR$(U):: L$=STR
$(L):: U=LEN(U$):: L=LEN(L$)
:: RETURN
480 IF PFLAG THEN PRINT #1:C
C$ :: PFLAG=0 :: RETURN ELSE
 RETURN
490 FOR X=A1 TO A2 STEP 22 :
: LN=LN+10 :: CALL START(LN,
X)
500 FOR Y=0 TO 21 :: IF X+Y>
A2 THEN 520
510 CALL PEEK(X+Y,A):: A$=ST
R$(A):: L=LEN(A$):: PRINT #1
:OC$;CHR$(L);A$;:: NEXT Y
520 PRINT #1:CC$ :: NEXT X :
: RETURN
530 CLOSE #2 :: IF RELOC THE
N ADDR=RELOC+9460 :: GOSUB 4
70 :: DISPLAY AT(16,1)BEEP:"
REMEMBER TO ADD":"  CALL LOA
D(8194,";U$;",";L$;")"
540 IF DEFADD<16384 THEN ADD
R=DEFADD :: GOSUB 470 :: DIS
PLAY AT(19,1)BEEP:"REMEMBER
TO ADD":"  CALL LOAD(8196,";
U$;",";L$;")"
550 GOTO 250
560 DISPLAY AT(9,1):"OUTPUT
FILE? DSK1." :: ACCEPT AT(9,
14)SIZE(-15)BEEP:O$ :: OPEN
#1:O$,VARIABLE 163,OUTPUT ::
 RETURN
570 DISPLAY AT(8,1):"INPUT
FILE? DSK1." :: ACCEPT AT(8,
14)SIZE(-15)BEEP:I$ :: OPEN
#2:I$,INPUT ,FIXED :: RETURN
580 SUB START(LN,X)
590 A=INT(LN/256):: B=LN-256
*A :: Y=ABS(X):: Y$=STR$(Y):
: L=LEN(Y$)
600 PRINT #1:CHR$(A);CHR$(B)
;CHR$(157);CHR$(200);CHR$(4)
;"LOAD";CHR$(183);
610 IF X<0 THEN PRINT #1:CHR
$(194);
620 PRINT #1:CHR$(200);CHR$(
L);Y$;
630 SUBEND
640 SUB DEC(A$,A):: A=0
650 L=LEN(A$):: FOR X=1 TO L
 :: A1=ASC(SEG$(A$,X,1)):: A
2=A1-48+7*(A1>57):: A=A+A2*1
6^(L-X):: NEXT X
660 A=A+65536*(A>32767):: SU
BEND
670 SUB WARN1 :: DISPLAY AT(
22,1)BEEP:"WARNING! FILE CON
TAINS AUTO START AND MAY NOT
 BE XBASIC   PRESS ANY KEY T
O CONTINUE"
680 CALL PRESS :: SUBEND
690 SUB WARN2 :: DISPLAY AT(
22,1)BEEP:"WARNING! FILE CON
TAINS EXT  REF'S AND MAY NOT
 BE XBASIC   PRESS ANY KEY T
O CONTINUE"
700 CALL PRESS :: SUBEND
710 SUB WARN3 :: DISPLAY AT(
22,1)BEEP:"WARNING! FILE CON
TAINS A BADTAG! PROGRAM ABOR
TED"
720 PRINT #1:CHR$(255);CHR$(
255):: CLOSE #1 :: CLOSE #2
:: STOP :: SUBEND
730 SUB PRESS
740 CALL KEY(0,K,S):: IF S=0
 THEN 740
750 SUBEND
```

ORIGINS

```
100 !DETERMINE ORIGINS OF A
D/F 80 FILE, I.E.ADDRESS RAN
GE LOADED, BY TOM FREEMAN, L
A 99'ERS
110 DEF$="56" :: ER$="78" ::
 ORG$="9A" :: DAT$="BC"
120 INPUT "NAME OF DIS/FIX 8
0 FILE TO  BE ANALYZED ":F$
:: OPEN #1:F$,FIXED,INPUT
130 INPUT "PRINTER? (PRESS E
NTER FOR   SCREEN DISPLAY) "
:P$ :: IF P$<>"" THEN P=2 ::
 OPEN #P:P$,VARIABLE 136 ::
PRINT #P:CHR$(15);
140 LINPUT #1:A$ :: RL$=SEG$
(A$,2,4):: CALL DEC(RL$,RL):
: PRINT #P:RL;"BYTES OF RELO
CATABLE CODE"
150 A$=SEG$(A$,14,80):: GOTO
 170
160 LINPUT #1:A$ :: L=L+1 ::
 PRINT :"RECORD";L;
170 T$=SEG$(A$,1,1):: IF T$=
":" THEN 260 ELSE IF POS(ER$
,T$,1)THEN 160
180 IF POS(DAT$,T$,1)THEN A$
=SEG$(A$,6,80):: BSFLAG=0 ::
 GOTO 170
190 ORG=POS(ORG$,T$,1):: IF
ORG=0 THEN 240
200 R=(ORG=2):: GOSUB 290
210 IF BSFLAG THEN PRINT #P:
"BSS";
220 IF ORG=2 THEN PRINT #P:"
R";
230 PRINT #P:AD;:: BSFLAG=1
:: A$=SEG$(A$,6,80):: GOTO 1
70
240 RDEF=POS(DEF$,T$,1):: IF
 RDEF=0 THEN PRINT #P:"ERROR
-NOT AN XB DF80 FILE" :: GOT
O 270
250 DT=DT+8 :: R=(RDEF=1)::
GOSUB 290 :: PRINT #P:"DEF "
;SEG$(A$,6,6);AD :: A$=SEG$(
A$,12,80):: GOTO 170
260 PRINT #P:" ":DT;"BYTES I
N DEF TABLE"
270 CLOSE #1 :: IF P THEN CL
OSE #P
280 STOP
290 AD$=SEG$(A$,2,4):: CALL
DEC(AD$,AD):: AD=AD-9460*R :
: RETURN
300 SUB DEC(A$,A):: A=0
310 FOR X=1 TO 4 :: A1=ASC(S
EG$(A$,X,1)):: A2=A1-48+7*(A
1>57):: A=A+A2*16^(4-X):: NE
XT X
320 A=A+65536*(A>32767):: SU
BEND
```

HIDDEN

```
32760 !QUICKLY DETERMINE THE
 ADDRESS RANGE OF A HIDDEN A
SSEMBLY PROGRAM IN AN XB PRO
GRAM, BY TOM FREEMAN, LA 99E
RS
32761 !MERGE THIS FILE INTO
THE XB FILE, THEN RUN 32762
32762 CALL PEEK(-31952,Z1,Z2
,Z3,Z4):: Z5=Z1*256+Z2-65536
 :: Z6=Z3*256+Z4-65536
32763 Z4=-32700 :: FOR Z7=Z5
+2 TO Z6-1 STEP 4 :: CALL PE
EK(Z7,Z1,Z2):: Z3=Z1*256+Z2-
65536 :: Z4=MAX(Z4,Z3):: NEX
T Z7
32764 CALL PEEK(Z4-1,Z1):: Z
4=Z4+Z1-1
32765 PRINT "PROGRAM RUNS FR
OM ";Z4;"TO -25"
```

PRINTMERGE

```
100 CALL CLEAR :: INPUT "NAM
E OF INPUT MERGE FILE?   ":F
$
110 OPEN #1:F$,VARIABLE 163
120 OPEN #2:"PIO",VARIABLE 1
32 :: PRINT #2:CHR$(15)
130 DIM A$(163)
140 LINPUT #1:B$ :: L=LEN(B$
)
150 FOR X=1 TO L :: A$(X)=SE
G$(B$,X,1):: NEXT X
160 FOR X=1 TO 33 :: IF X+Y>
L THEN 180 ELSE B=ASC(A$(X+Y
)):: PRINT #2:STR$(B);TAB(X*
4+1);
170 NEXT X
180 PRINT #2
190 FOR X=1 TO 33 :: IF X+Y>
L THEN 230
200 B=ASC(A$(X+Y)):: IF B<32
 OR B>126 THEN 210 ELSE PRIN
T #2:CHR$(B);
210 PRINT #2:TAB(X*4+1);
220 NEXT X
230 PRINT #2 :: IF X+Y<=L TH
EN Y=Y+33 :: GOTO 160
240 LINPUT #1:B$ :: IF EOF(1
)THEN 250 ELSE Y=0 :: L=LEN(
B$):: GOTO 150
250 CLOSE #1 :: CLOSE #2
```

NOTE1: In addition to the list of tokens below

199 (C7) indicates a quoted string follows
2ØØ (C8) indicates an unquoted string follows
2Ø1 (C9) indicates 2 bytes for a line number follows

NOTE2: This is Danny Michaels XB for GK. There are a few tokens not in TI XB

!	131	83	VAL	218	DA	NEW	Ø	ØØ	:	181	B5
#	253	FD	XOR	188	BC	CON	1	Ø1	)	182	B6
&	184	B8	BASE	241	F1	CONTINUE	1	Ø1	(	183	B7
(	183	B7	BEEP	238	EE	LIST	2	Ø2	&	184	B8
)	182	B6	CALL	157	9D	BYE	3	Ø3	OR	186	BA
*	195	C3	CHR$	214	D6	NUMBER	4	Ø4	AND	187	BB
+	193	C1	COPY	1Ø	ØA	NUM	4	Ø4	XOR	188	BC
,	179	B3	DATA	147	93	OLD	5	Ø5	NOT	189	BD
-	194	C2	ELSE	129	81	RES	6	Ø6	=	19Ø	BE
/	196	C4	GOTO	134	86	SAVE	7	Ø7	<	191	BF
:	181	B5	LIST	2	Ø2	MERGE	8	Ø8	>	192	CØ
;	18Ø	B4	MOVE	11	ØB	DEL	9	Ø9	+	193	C1
<	191	BF	NEXT	15Ø	96	COPY	1Ø	ØA	-	194	C2
=	19Ø	BE	OPEN	159	9F	MOVE	11	ØB	*	195	C3
>	192	CØ	READ	151	97	ELSE	129	81	/	196	C4
^	197	C5	RPT$	225	E1	::	13Ø	82	^	197	C5
::	13Ø	82	SAVE	7	Ø7	!	131	83	EOF	2Ø2	CA
AT	24Ø	FØ	SEG$	216	D8	IF	132	84	ABS	2Ø3	CB
GO	133	85	SIZE	235	EB	GO	133	85	ATN	2Ø4	CC
IF	132	84	STEP	178	B2	GOTO	134	86	COS	2Ø5	CD
ON	155	9B	STOP	152	98	GOSUB	135	87	EXP	2Ø6	CE
OR	186	BA	STR$	219	DB	RETURN	136	88	INT	2Ø7	CF
PI	221	DD	THEN	176	BØ	DEF	137	89	LOG	2Ø8	DØ
TO	177	B1	BREAK	142	8E	DIM	138	8A	SGN	2Ø9	D1
ABS	2Ø3	CB	CLOSE	16Ø	AØ	END	139	8B	SIN	21Ø	D2
ALL	236	EC	DIGIT	233	E9	FOR	14Ø	8C	SQR	211	D3
AND	187	BB	ERASE	239	EF	LET	141	8D	TAN	212	D4
ASC	22Ø	DC	ERROR	165	A5	BREAK	142	8E	LEN	213	D5
ATN	2Ø4	CC	FIXED	25Ø	FA	UNBREAK	143	8F	CHR$	214	D6
BYE	3	Ø3	GOSUB	135	87	TRACE	144	9Ø	RND	215	D7
CON	1	Ø1	IMAGE	163	A3	UNTRACE	145	91	SEG$	216	D8
COS	2Ø5	CD	INPUT	146	92	INPUT	146	92	POS	217	D9
DEF	137	89	MERGE	8	Ø8	DATA	147	93	VAL	218	DA
DEL	9	Ø9	PRINT	156	9C	RESTORE	148	94	STR$	219	DB
DIM	138	8A	TRACE	144	9Ø	RANDOMIZE	149	95	ASC	22Ø	DC
END	139	8B	USING	237	ED	NEXT	15Ø	96	PI	221	DD
EOF	2Ø2	CA	ACCEPT	164	A4	READ	151	97	REC	222	DE
EXP	2Ø6	CE	APPEND	249	F9	STOP	152	98	MAX	223	DF
FOR	14Ø	8C	DELETE	153	99	DELETE	153	99	MIN	224	EØ
INT	2Ø7	CF	LINPUT	17Ø	AA	REM	154	9A	RPT$	225	E1
LEN	213	D5	NUMBER	4	Ø4	ON	155	9B	NUMERIC	232	E8
LET	141	8D	OPTION	158	9E	PRINT	156	9C	DIGIT	233	E9
LOG	2Ø8	DØ	OUTPUT	247	F7	CALL	157	9D	UALPHA	234	EA
MAX	223	DF	RETURN	136	88	OPTION	158	9E	SIZE	235	EB
MIN	224	EØ	SUBEND	168	A8	OPEN	159	9F	ALL	236	EC
NEW	Ø	ØØ	UALPHA	234	EA	CLOSE	16Ø	AØ	USING	237	ED
NOT	189	BD	UPDATE	248	F8	SUB	161	A1	BEEP	238	EE
NUM	4	Ø4	DISPLAY	162	A2	DISPLAY	162	A2	ERASE	239	EF
OLD	5	Ø5	NUMERIC	232	E8	IMAGE	163	A3	AT	24Ø	FØ
POS	217	D9	RESTORE	148	94	ACCEPT	164	A4	BASE	241	F1
REC	222	DE	SUBEXIT	167	A7	ERROR	165	A5	VARIABLE	243	F3
REM	154	9A	UNBREAK	143	8F	WARNING	166	A6	RELATIVE	244	F4
RES	6	Ø6	UNTRACE	145	91	SUBEXIT	167	A7	INTERNAL	245	F5
RND	215	D7	WARNING	166	A6	SUBEND	168	A8	SEQUENTIAL	246	F6
RUN	169	A9	CONTINUE	1	Ø1	RUN	169	A9	OUTPUT	247	F7
SGN	2Ø9	D1	INTERNAL	245	F5	LINPUT	17Ø	AA	UPDATE	248	F8
SIN	21Ø	D2	RELATIVE	244	F4	THEN	176	BØ	APPEND	249	F9
SQR	211	D3	VALIDATE	254	FE	TO	177	B1	FIXED	25Ø	FA
SUB	161	A1	VARIABLE	243	F3	STEP	178	B2	TAB	252	FC
TAB	252	FC	RANDOMIZE	149	95	,	179	B3	#	253	FD
TAN	212	D4	SEQUENTIAL	246	F6	;	18Ø	B4	VALIDATE	254	FE

BENCHMARKS & DIFFERENT LANGUAGES

by Stephen Shaw

In the October 1986 issue of Personal Computer World a new set of benchmarks was listed, suitable to test all languages - although some may not be applicable to some languages!

Looking over these benchmarks for the various languages available for the TI99/4A may help you to appreciate some of the good points of some of the languaes - and some of the weak points too.

By giving listings in the various languages, you may also be helped to understand how to use them - and if you dont have them already, maybe thing about obtaining them (or not!!).

I shall look at each benchmark in turn, and start with the first, called INTMATH - here is the description:

```
DECLARE THREE INTEGER VARIABLES X,Y AND I
 ASSIGN X EQUAL TO 0 AND Y EQUAL TO 9
  WRITE "START" TO SCREEN
   REPEAT 1000 TIMES USING I AS THE LOOP VARIABLE
    ASSIGN X=X+Y-Y*Y/Y
   WRITE "FINISH" AND X TO SCREEN
      [ X SHOULD BE ZERO ]
```

In the various Basics available on the TI, Integer Math is available ONLY with version 2.1 of Myarc's Extended Basic:

```
 100 DEFINT X,Y,I
 110 X=0 :: Y=9
 120 PRINT "START"
 130 FOR I=1 TO 1000
 140 X=X+Y-Y*Y/Y
 150 NEXT I
 160 PRINT "FINISH";X
```

This took 18 seconds.

In the following timings "MYARC EXBAS" refers to Version 2.0, using all real math, unless otherwise specified.

Integer math is also available in Machine Code, C-99, and Forth. I will leave the machine code to someone else..

TI FORTH:

```
 0 VARIABLE X   0 VARIABLE Y
 0 X !          9 Y !
 : TS  ."  START"
   7 10001 1 DO DROP
    X @
    Y @ +
    Y @ -
    Y @ *
    Y @ /
      LOOP
       TS
```

You may note that I have used a 10000 loop instead of just one thousand - mainly to give me a long enough period to time! This runs in 28.6 seconds, equivalent to 2.86 seconds for one thousand.

c-99

```
/* CONIO FIRST */
#define stdin    -1
#define stdout   -2
#define stderr   -3
#define EOF      -1
#define YES       1
#define NO        0
#define NULL      0
#define EOL      10
#define FF       12
#define BS        8

int x,y,i,loop;
main()
{ putchar(FF);
 x=0;
 y=9;
 while(++loop<11)
  { puts("start");
    i=0;
    while(++i<10001)
      { x=y+y-y*y/y;
      }
   }
puts("  finish");
}
```

Notice that ten thousand was not enough for accurate timing! This ran at a rate equivalent to 0.4755 seconds for one thousand.

Before we pass to Benchmark two, a note on these timings- we know of at least 6 different console operating systems - and three different TI extended basics - so timings on other systems may differ somewhat! TI did not bother with version numbers, but the set up used was: TI99/4A console, TI peripheral expansion box, TI disk controller, Myarc ram card, TI SSSD disk drive (made by MPI). The extended basic module used was the second version of what TI called Version 110.

So, onto BENCHMARK TWO, called REALMATH:

```
DECLARE TWO REAL VARIABLES X AND Y
 DECLARE AN INTEGER VARIABLE I
  ASSIGN X EQUAL TO 0 AND Y EQUAL TO 9.9
   WRITE "START" TO SCREEN
    REPEAT 1000 TIMES USING I AS THE LOOP VARIABLE
     ASSIGN X=X+Y-Y*Y/Y
      WRITE "FINISH" AND X TO SCREEN
       [ X SHOULD EQUAL 0 ]
```

Again, we have problems with using integers... and you do not need to declare variables (why when there is only one class!) in Basic and in some languages there is no loop counter to be called I... however, this is one benchmark we can test in MANY languagse!!!

BASIC first:

```
100 X=0
110 Y=9.9
120 PRINT "START"
130 FOR I=1 TO 1000
140 X=X+Y-Y*Y/Y
150 NEXT I
160 PRINT "FINISH";X
```

Timings are:
TI BASIC: 17.7 seconds
TI EXTENDED BASIC: 22.0 seconds
MYARC EXTENDED BASIC: 31.8 seconds
(in these tests we have used Vn 2.0 of Myarcs Extended Basic - contrary to the ads, it does NOT support integer math) [Using Myarc Vn 2.1 and defining i as an integer the timing is reduced to 23 seconds - a useful speed up] Extended Basic allows multi-statement lines, and if the benchtest is compressed to a single program line in this manner, for both tested versions of Extended Basic, the running time is reduced by six per cent.

PILOT-99: In this language, use is made of 32 byte floating point math:

```
C: #X<-0
C: #Y<-9.9
T: START
LP: 100
C: #X<-#X+#Y-#Y*#Y/#Y
EL:
T: FINISH #X
E:
```

This has been looped only 100 times to save insomnia, as the test runs equivalent to 1000 loops in 576 seconds!

We now turn to BENCHMARK 3: TRIGLOG

```
 DECLARE TWO REAL VARIABLES X AND Y
  DECLARE AN INTEGER VARIABLE I
   WRITE "START" TO SCREEN
    ASSIGN X EQUAL TO 0
    ASSIGN Y EQUAL TO 9.9
     REPEAT 1000 TIMES USING I AS THE LOOP VARIABLE
      ASSIGN X=COS(SIN(ATN(LOG(Y))))
   WRITE "FINISH" AND X TO SCREEN
    [ X SHOULD BE 1000 ]
```

Now as COS cannot produce a value of 1000, we shall read this as 1.000, which still leaves a problem: the benchmark does NOT specify that LOG must be to Base 10 nor that trig uses degrees - but if you want an answer of 1.000 that is what you must use. Our TI99/4A uses BASE e for logs and radians for trig - so for many of the tests below I have used these. There is a comparison test using the "assumed" bases.

Basic first:

```
100 PRINT "START"
110 X=0
120 Y=9.9
130 FOR I=1 TO 100
140 X=COS(SIN(ATN(LOG(Y))))
150 NEXT I
160 PRINT "FINISH";X
```

The remarkable accuracy of the TI99/4A, coupled with trig routines written in long GPL sequences means TI trig is slow, hence only looped 100 times. However the results below are for 1000 loops (100 x 10 !):
TI Basic: 624 secs
TI Extended Basic: 362 secs
Myarc Extended Basic: 365 secs (Vn 2.0 AND Vn 2.1)

Using Log Base 10 and trig in degrees:

```
100 M=0.01745329251994
110 L=2.302585093
120 PRINT " START "
130 X=0
140 Y=9.9
150 FOR I=1 TO 100
160 X=COS(M*SIN(M*ATN(M*LOG(Y)/L)))
170 NEXT I
180 PRINT " FINISH ";X
```

The equivalent timings for one thousand loops are:
TI Basic: 640 secs
TI Extended Basic: 386 secs
Myarc Extended Basic: 392 secs

I will leave others the pleasure of testing this benchmark in TI Forth, C-99 and machine code, but here is what PILOT-99 can do:

```
T: START
C: #X<-0
C: #Y<-9.9
LP: 100
C: #X<-COS(SIN(ATN(LOG(#Y))))
EL:
T: FINISH #X
E:
```

The equivalent time for 1000 loops is here 1710 secs!

BENCHMARK 4 is TEXTSCRN:

```
 DECLARE AN INTEGER VARIABLE I
  WRITE "START" TO SCREEN
   REPEAT 1000 TIMES USING I AS THE LOOP VARIABLE
    WRITE "1234567890QWERTYUIOP" AND I TO THE SCREEN
  WRITE "FINISH " TO SCREEN
```

Another problem looms... the benchmark makes no mention of scrolling, a time consuming process. Not a lot of choice in TI Basic, but we do have the option in the other languages - so this benchmark has been run twice, with and without scroll. The results of this benchmark appear to be quite miraculous and will be commented on in the summary at the end:

```
100 PRINT "start"
110 FOR I=1 to 100
120 PRINT "1234567890QWERTYUIOP";I
130 NEXT I
140 PRINT "FINISH"
```

Times for 1000 loops:
TI BASIC: 260 secs

TI Extended Basic: 117 secs
Myarc Extended Basic: 73 seconds

Now, without the scroll:

```
100 PRINT "START"
110 FOR I=1 TO 100
120 DISPLAY AT(3,1):"1234567890QWERTYUIOP";I
130 NEXT I
140 PRINT "FINISH"
```

Times for 1000 loops:
TI Extended Basic: 76 seconds
Myarc Extended Basic: 34 seconds
(repeat: thirty four seconds)
but... incredible this - Version 2.1 of Myarc ExBas still manages an improvement and take it down to just 30 seconds!

Now...TI FORTH...first,scrolling:

```
 : TEST
   101 1 DO
         LOOP
```

The equivalent time for 1000 loops is 68 secs.

Next, non-scrolling:

```
 : TEST
   1001 1 DO
    4 5 GOTOXY
         LOOP
```

This routine takes 33.6 secs for 1000 loops.

Before we move on to the fastest, let's try the slowest, PILOT-99:
(non-scrolling):

```
C: #I<-0
T: START
LP: 100
C: #I<-#I+1
TC: 4,5
T: 1234567890QWERTYUIOP #I
EL:
T: FINISH
E:
```

For 1000 loops, this takes...980 secs

Moving quickly on to the fastest, first WITH scrolling, here is what C-99 can do:

```
/* console i/o first */
#define stdin    -1
#define stdout   -2
#define stderr   -3
#define EOF      -1
#define YES       1
#define NO        0
#define NULL      0
#define EOL      10
#define FF       12
#define BS        8
int I;
main()
{ putchar(FF);
  puts("start ");
   while(++i<5001)
   { puts("1234567890qwertyuiop");
   }
puts("finish");
}
```

Unfortunately my C skills have not yet become sufficiently refined to print the loop variable, but it should not make too much difference...
The time for 1000 loops here is 38.72 secs.

Now, take the scroll out - the upper part of the program up to main() is identical so I won't repeat it here:

```
main()
{ putchar(FF):
  puts("start ");
    while(++i<5001)
    { locate(3,4);
      puts("1234567890qwertyuiop");
    }
  puts("finish");
}
```

and this version runs 1000 loops in 6.82 seconds.

Now on to Benchmark Five, GRAFSCRN:

```
 DECLARE INTEGER VARIABLES I AND J
  WRITE "START" TO SCREEN
   REPEAT 100 TIMES USING X AS THE LOOP VARIABLE
    REPEAT 100 TIMES USING Y AS THE LOOP VARIABLE
     PLOT PIXEL AT SCREEN LOCATION (X,Y)
  PRINT "FINISH" TO SCREEN.
```

OK- spot the deliberate error - the top line should ask you to declare variables called X and Y and I and J! Minor point... and speaking of points, all this benchmark is for is to find out how long it takes to plot 10,000 pixels on screen...

Not available in TI Basic or TI Extended Basic, but it can be done with Myarc Extended Basic:

```
100 CALL GRAPHICS(3)
110 CALL WRITE(1,22,2,"START")
120 FOR X=1 TO 100
130 FOR Y=1 TO 100
140 CALL POINT(1,X,Y)
150 NEXT Y
160 NEXT X
170 CALL WRITE(1,23,2,"FINISH")
180 GOTO 180
```

This took 268 seconds.
Running the program in Version 2.1 and using intergers, the time is reduced to 150 seconds.

FORTH also allows bit map graphics, so here is TI FORTH:

```
: TEST   PAGE  GRAPHICS2
    101 1 DO
     101 1 DO
       I J DOT
     LOOP
    LOOP
      TEXT ;
```

Notice that TI FORTH forces us to use those errant variables I and J!
This routine takes just 20.1 seconds.

Now, PILOT-99 also allows us to use bit map graphics, but those 32 byte floating point numbers seem to get in the way:

```
IG:
GC: 2,16
C: #X<-30
C: #Y<-0
TG: 12,9,START
LP: 50
LP: 25
PP: #Y,#X
C: #X<-#X+1
EL:
C: #X<-30
C: #Y<-#Y+1
EL:
TG: 13,9,FINISH
E:
```

Right- I have not used two loops of 100 each, instead, just 50 and 25, to give a total pixel count of one eighth of what it should be...and here is why- to plot 10,000 pixels would take one hour five minutes and 36 seconds.

Let's finish this benchmark off with something a might faster:
9900 Source Code.
VSBW, VSBR, and VWTR are externally referenced to routines in the Editor Assembler module, and also available in the disk file EAU with FUNLWRITER.
The routines SETUP, CLEAR and PLOT are not shown here - they were written by Graham Marshall and appeared in the diskazine 4FRONT published by New Day Computing:

```
***************
* bit map
* benchmark 5
***************
        DEF   BITMAP
        REF   VSBW,VSBR,VWTR
***************
ROW     DATA  >0000
COLM    DATA  >0000
BITMAP  BL    @SETUP
        LI    R2,>F100
        MOV   @ROW,R8
        MOV   @COLM,R9
POINT   MOV   R8,R0
        MOV   R9,R1
        BL    @PLOT
        INC   R8
        CI    R8,100
        JGT   ROWINC
        JMP   POINT
ROWINC  MOV   @ROW,R8
        INC   R9
        CI    R9,101
        JLT   POINT
END     LI    R0,>F100
        BL    @CLEAR
        JMP   BITMAP
* graham marshalls routines here
****************
```

This routine took 8.66 seconds.

FINALLY:
BENCHMARK 6: STORE:

```
 DECLARE AN INTEGER VARIABLE I
 WRITE "START" TO SCREEN
  CREATE A DISK FILE "TEST"
   OPEN "TEST" [READY FOR] INPUT
    REPEAT 1000 TIMES USING I AS THE LOOP VARIABLE
     WRITE THE RECORD "1234567890QWERTYUIOP" TO "TEST"
   CLOSE "TEST"
  DELETE "TEST"
 WRITE "FINISH" TO SCREEN
```

This also gives us a problem or two, mainly because the TI Disk Operating system automatically does some of the work for us - we don't have to "create a disk file", the system does that for us when we open the file!
And if you are used to opening a file "OUTPUT" when you write to it, the reference to "open "test" for input" just might throw you there...
I've only run this benchmark in the Basics, for to TI Disk Drive and to Myarc Ram Card, I'll leave the other languages to better programmers...
In the listings below, the file is opened as the default of DISPLAY, UPDATE, VARIABLE 80, and a well used disk was used - the same one for every test, in exactly the same condition for each.
Now, DV80 may not be the fastest way of running this benchmark! but the conditions WERE identical for every run:

```
100 PRINT "START"
110 OPEN #1:"DSK1.TEST"
120 FOR I=1 TO 1000
130 PRINT #1:"1234567890QWERTYUIOP"
140 NEXT I
150 CLOSE #1
160 DELETE "DSK1.TEST"
170 PRINT "FINISH"
```

Using the MPI/TI SS.SD. disk drive:
 TI BASIC: 166.5 secs
 TI Extended Basic: 131.5 secs
 Myarc Extended Basic: 82.8 secs

Using the Myarc Ram Disk:
 TI Basic: 112.6 secs
 TI Extended Basic: 68.6 secs
 Myarc Extended Basic: 34.6 seconds.

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

NOW... a brief summary of all those timings.

BENCHMARK:	INTMATH	REALMATH	TRIGLOG	TEXTSCRN	GRAFSCRN	STORE
---all times for 1000 loops as definition---						
LANGUAGE:						
			TI Norm	SCROLL		DISK
TI BASIC	NP	17.7 SEC	624 SEC	260 SEC	NP	166 SEC
			Base 10	NOSCROLL		RAMDISK
			640 SEC	NP		113 SEC
TI EX BAS						
Vn 110	NP	22.0 SEC	362 SEC	117 SEC	NP	131 SEC
			386 SEC	76 SEC		69 SEC
MYARC						
EXT BASIC						
Vn 2.0	NP	31.8 SEC	365 SEC	73 SEC	268 SEC	83 SEC
			392 SEC	34 SEC		35 SEC
Vn 2.1				untried		
	18 SEC	23 sec	as 2.0	30 sec	150 sec	untried
PILOT-99						
	NP	576 SEC	1710 SEC	untried	65.5 MINUTES	untried
			untried	980 SEC		untried
TI FORTH						
	2.86	untried	untried	68 SEC		untried
	SEC		untried	34 SEC	20.1 SEC	untried
C-99						
	0.48	untried	untried	38.7 SEC		untried
	SEC		untried	6.8 SEC	untried	untried
9900						
MACHINE	not	untried	untried	untried	8.66 SEC	untried
CODE	tried		untried	untried		untried

We long ago became accustomed to the TI99/4A being placed very very close to the bottom of the PCW benchmark listings.... but take another look at the times for TEXTSCRN above.

In the October 1986 issue, PCW gave the following timings, for these very new and very fast machines:

COMMODORE AMIGA BASIC.....	150.3 seconds!
IBM PC + BasicA...........	100.0 seconds
IBM PC + Turbo Pascal......	76.4 seconds
IBM PC + PC Forth..........	67.0 seconds
ATARI ST + Atari Basic.....	44.8 seconds
Atari ST + Megamax C.......	39.6 seconds

Well well well....

What can we deduce?

Extended Basic tends to be faster than TI Basic, and while Myarc Extended Basic can be much faster than TI Extended Basic, it is NOT the rule for that to be so!

PILOT-99 is really held back by those 32 byte numbers and is not a language of choice for speed!

TI FORTH can be fast, and C-99 is well worth looking at if you can't make it all the way to 9900 machine code.

STEPHEN SHAW SEPTEMBER 1986

Did you know that...?

by Chick De Marti

OOPS?

If you think my eyes look bad, you should se how my face looks with my foot in my mouth! Last month I stated that I had loaded FORTH with my MINIMEM. Well since someone called me and said I couldn't...I CAN'T ??? I'll figure how or what I did and will keep you informed.

Chick

<*> <*> <*> <*> <*>

From the SUNCOAST BEEPER:

According to the Smart Programmer, Fast term users should note that the program was written for XMODEM dis/fix 128 file transfers in order to facilitate transfers between various types and brands of computers. This means that a TI file header is not sent on such files. But Barry Travers, of Smart Programmer and the GENIAL TRAVelER, has announced that protecting a file prior to uploads will trigger sending the TI file header.

<*> <*> <*> <*> <*>

A REMINDER (I just remembered!)

Instead of defining CHR$(130), the left harp, as "FFFFFFFFFFFFFFFF" to produce a solid square, just use CALL COLOR(13,2,2).

NOTE: this works well if no other characters from set 13 are to be used in the program.

<*> <*> <*> <*> <*>

Here's a game "FOR EXPERTS"

WE have 9 slips of paper with the numbers from 1 to 9 on them. Two people play the game...each take turns picking a number. The first one to pick slips who's figures total 15 wins the game.

P.S. With a little clue, we can make this "EXPERTS GAME" childs play. (Our clue can be found on the next page.)

"DIAL A WORD" ERROR

An error in the "dial a word" program by Joey Latimer, was loacated by Mark Wannop. "Line 20 sets string A$ to the entire alphabet. However, there is no "Q" or "Z" in the telephone dial. As a result, the letter combonations do not come out right. To correct, simply delete Q and Z from the string".

<*> <*> <*> <*> <*>

from the MID-SOUTH 99ers

Many users of Funlwriter have discovered...after returning from certain sections...when EDITOR is selected, the word wrap is no longer working. This is because you are now using Editor/Assembler Editor where there is no word wrap. To get the regular TI Writer Editor, make sure the first two selections on the menu are Editor and Formatter. Selection #2 can (sometimes) be changed by pressing selection #6 ("SWITCH"). REMEMBER! Selection #2 MUST say Formatter...for the current Editor to be used.

Did you know you can have more than one RS232 in the PEB? However, thee catch is the second card must be modified to be accessed as RS232/3 and RS232/4 and PIO/2. This modification will be done to TI brand RS232 cards FREE of charge if you send the card to TI. With a "Y" cable on the serial port you will have RS232/1,RS232/2, RS232/3, RS232/4, PIO/1 and PIO/2. Call 1-800-TI-CARES for details.

Thanx to Gary Cox for the above two items.

<*> <*> <*> <*> <*>

MISC. STUFF

Another puzzle for you math geniuses.

```
      B E S T

    + M A D E
    ---------                BOTTOM
    M A S.E R   (Answer this page.)
```

WHY_ARE_FIRETRUCKS_RED?

Well, fire trucks have four wheels and eight men. Four and eight make twelve. There are twelve inches in a foot, and a foot is a ruler. Queen Elizabeth is a ruler, and it's also one of the largest ships on the sea. The sea has fish and fish have fins. The Finns fought the Russians once and the Russians are red. That's why fire engines are red, 'cause they're always rushin'.

If you think this is illogical, you should hear some of the excuses people give for not participating in club activities.
(Author unknown)

<*> <*> <*> <*> <*>

GOOD_ERROR_ROUTINE

Error routines are somewhat confusing. I thought I'de search for one to use -as a demonstration. While this routine is not unique, the program "DISK LABEL MAKER" by J.D.Gootee, in the Nov. Jackson County 99er's Newsletter, brought it to my attention again...for I, in turn offer it to you as an example of a "GOOD ERROR ROUTINE"

```
150 ON ERROR 500 ! used prior to an
500 REM ERROR TRAP ROUTINE
510 CALL ERR(C,T,S,L):: IF T<0
<0 THEN 520 :: IF T>0 THEN 540
520 CALL CLEAR :: DISPLAY AT(12
,1):"PROGRAM ERROR#";C;"FOUND
IN ";L :: DISPLAY AT(13,1):
"PLEASE TRY AGAIN"
530 FOR X=1 TO 1000 :: NEXT X ::
RETURN 140
540 CALL CLEAR :: DISPLAY AT(12
,1):"I/O ERROR ";C;" FOUND IN"
;L :: DISPLAY AT(13,1):"PLEASE
TRY AGAIN"
550 FOR X=1 TO 1000 :: NEXT X ::
RETURN
560 END
```

COLOR_COMBINATIONS_IN_FORTH

This little hint comes from AVTI of Lancaster, Ca.

In the immediate mode of FORTH (no program running) you can type in this little ditty:

```
: SEE 252 22 D0 I DUP . 7 VWTR
KEY 2 = IF ABORT ENDIF LOOP ;
```

After you get the ok type SEE. Don't worry if you can't read anything at times, the FG and BG colors may match and there will be nothing to read. When you see a combination that gives you a good screen display, write down the last number... (bottom of the screen) and continue to step through the loop...(or exit via the FCTN-2).
(...you will have chosen the best color combination available for your set...Ed).

<*> <*> <*> <*> <*>

DOUBLE-STRIKE AND ELITE

Here's a quickie for you. With my Epson FX-80 when you try to print in ELITE and then try to double-strike ... the printer will only single-strike! Try entering in the immediate mode (still in XBasic):

```
OPEN #2:"PIO"
PRINT #2:CHR$(27);"E";CHR$(27);"G";
```

It works...(he says not to confidently)

<*> <*> <*> <*> <*>

ANSWERS...

```
#1                     #2

    2 1 9 1 4              9 5 6 7
   -----------
    7 1 5 1 3              1 0 8 5
   -----------            ---------
    6 1 1 1 8            1 0 6 5 2
```

Well i'm out of coffee...see you next month.

Chick

List of BBS's from the Gardena Area

Zone L = Local calls
" 2 = .08 per Min.
" 3 = .10 per Min.

ADDITIONAL BBS INFORMATION PAGE 17

Compiled by
Chick De Marti
LA99ers U.G.

	Hrs.?...Redondo Beach	14th DIMENTION.....	541-6250	L
	24 hrs...Haw/Comp	ACCESS ONE.........	537-3378	L
	Hrs.?...Norwalk	APPLE FORTRESS.....	924-5644	3
*	24 Hrs...TI or XModem	99 BBS (LaHabra)...	947-7777	3
	L.A. Sports BBS	AWARE II...........	851-0780	3
	Hrs.?...Alamitos	BANKERS & HACKERS..	498-6581	3
	Hrs.?...Lomita	BEACH BUM BBS......	325-3237	L
	Hrs.?...LOMITA	BEST OF BOTH.......	325-3237	L
	Hrs.?...EL SEGUNDO	BILBO'S HIDEAWAY...	640-6104	L
	Hrs.?...INGLEWOOD	BLACK KNIGHT INC...	641-6373	L
	Hrs.?...Redondo Bch	BRIDGE OF CY'S.....	372-2282	L
	Hrs.?...Santa Monica	BR'S BBS...........	394-5950	3
	Hrs.?...West L.A.	BULL BOARD.........	473-3128	3
	L.A. for Handycappers	BULLETIN BOARD SYS.	934-6026	3
	Hrs.?...West L.A.	BUY PHONE..........	474-0270	3
*	(TI-EXPRESS) Local	CALIFORNIA 80......	589-0372	L
	Hrs.?...West L.A.	C.C.NETWORK........	479-4074	3
	Hrs.?...L.A.	CHARLATAN'S CABIN..	658-4192	3
	Hrs.?...Beverly Hills	CINIMA BOARD.......	273-1314	3
	300-1200... West L.A.	CIRCUIT............	477-7465	3
	Hrs.?...L.A.	COLOR-80 BBS#44....	258-0640	4
	300/1200...Redondo Bch	COMMODORE CONNECT..	370-4997	L
*	TI Programs_Torrance	COMP.CONF.CENTER...	212-5267	L
	FOR COMP.PROFESSIONALS	COMP.CONSULTANT NET	479-4074	3
	Hrs.?...Mar Vista	COMP.CONSPIRACY....	398-3674	2
	Hrs.?...Long Beach	COMPULSIVE COLLECTER	428-5971	2
	Hrs.?...Redondo Bch	COMPUTERLAND COMF..	371-8825	L
	Hrs.?...Beverly Hills	CONNECTOR..........	859-0894	3
	Hrs.?...L.A. Strip	CYBORG'ASTLE.......	581-5573	L
	Hrs.?...Redondo Bch	DARK DIMENTION.....	541-1667	L
	Hrs.?...? ? ?	DARK SIDE..........	698-3306	3
	Hrs.?...Downey	DOWNEY CONNECTION..	563-6357	L
	Hrs7-11pm...Gardena	DYM(Dial Your Match)	397-6300	L
	Hrs7-11pm...San Pedro	DYM " " ".......	549-3584	L
	Hrs7-11pm...West L.A.	DYM " " ".......	479-2550	3
	Hrs.?...Long Beach	DRAGON'S LAIR......	595-9346	2
	Hrs.?...L.A.	ED GEIN DATA DATING	655-1841	3
	Hrs.?...L.A.	ENTERPRISE.........	386-9525	3
	300/1200...Redondo	EPSON..............	375-1391	L
	Hrs.?...Redondo Bch	FASTFIND CO........	377-0804	L
	Hrs.?...Gardena	FIDO BBS...........	329-4603	L
	Hrs.?...Lackwood	FLIGHT DECK........	421-1041	2
	Hrs.?...West L.A.	FREEHOLD...........	474-1075	3
	Hrs.?...West L.A.	FUTURE SHIELD......	208-8255	3
	Games only...Redondo	GAMENET............	541-8039	L
	300/1200...L.A.	GARY'S RBBS........	779-7471	L
	Hrs.?...Redondo Bch	G.F.R.N............	541-2503	L
	Hrs.?...Redondo Beach	HITCHIKERS A.E.....	541-7348	L
	Hollywood Commodre U.G.	HUG1...............	874-8256	3
	24 Hrs...San Pedro	HUGHS Msg. SER.....	549-9640	L
	Hrs.?...L.A.	IBM PC BBS.........	779-7471	L
	Hrs.?...Bevery Hills	INSANE ASYLUM......	276-0805	3
	24 hrs...Culver City	INT. COLLECTORS NET	204-0646	2

AN AID FOR RETURN TO PIRATES ISLE
By John Floyd

Well, let's face it!!!! Mr. Adams out did himself on this one! This adventure has got to be the most difficult of the entire series.

With a few sneaky tricks I have compiled a list of what I beleive to be ALL of the nouns and verbs used within the adventure. There are a few words that I am unsure of as to their use.

The word AUTO may be a verb rather than a noun and the words HAND, HYPE and SHOR can mean several things.

I hope that this list of words can be of some assistance to those of you who have become stumped while playing the adventure. If you are one of those that becomes offended by someone else offering hints, I recommend that you destroy this list immediately! (Or at least put it someplace that you cannot refer to it until you decide that you really need it!)

I have divided the list into nouns and verbs and have alphabetized each catagory for ease of locating words. All of the treasures are noted with an asterisk so that you can easily refer to them.

Now I myself have not been playing this adventure very long and by no means know the solution to the game. The list has however helped me in a few situatons.

Okay now! Try all of the verb/noun combinations as you trek through this fantastic adventure but don't get discouraged, you will die many times in this adventure and each is a learning experience.

HAPPY ADVENTURING

Directions
==========

ANY
NORTh
SOUTh
EAST
WEST
UP
DOWN

Nouns
=====

ALARm
ALGAe
AMBEr *
AROUnd
AUTOpilot ?
BACK
BATTery
BEACh
BEAM
BED
BLACkbeard
BLADe
BOAT
BOOK *
BOX
BREAth
BROOch *
BUNKbed
BUTTon
CABIn
CEILing
CHESt *
CLOCk
CRACk
CRAWlway
CREVice
CURRent
DECK
DEEP
DOCK
DOLLar *
DOOR
DUBLoons *
EARRing *
ENGIne
EYES
FACEmask
FAN
FLOOdlights
FLOTsam
FRAMe
FUEL
GAME
GAS
GLASses
GLUE
HAMMer
HANDle ?
HELM
HILL
INVEntory
ISLE
IT
JETSam
LADDer
LEDGe
LENS
LIGHts
MAP
MASK
MATTress
OBJEct
OCEAn
OILSkin
OPENing
OYSTer
PAINting
PEARl *
PICTure
PILIng
PIN *
PIRAte
POOL
PORThole
RAINcoat
REMBrandt*
RIM
RING *
ROCK
RUM
SCREwdriver
SEA
SHED
SHIP
SHOVel
SIGN
SILT
SNAIl
STAMps *
SUMMit
TOOLshed
TOP
TORCh
WATCh *
WATEr
WIRErim
YOHO

Verbs
=====

BREAthe
BURN
CHARge
CLEAn
CLIMb
CRANk
CRAWl
DESCribe
DIG
DIVE
DON
DRINk
DRIVe
DROP
EAT
EMPTy
ENTEr
EXAMine
FEEL
FIND
FIX
FOLD
FOLLow
GET
GIVE
GLUE
GO
GRAB
HEAR
HELP
HOLD
HYPErventilate
IN
INVEntory
Inventory
JUMP
KILL
LEAP
LEAVe
LIFT
LIGHt
LISTen
LOOK
Look.
MAKE
MIX
MOVE
OPEN
FAUSe
PICK
PRESs
PROBe
PULL
PUSH
PUT
PUTOn
QUIT
READ
RECHarge
RELEase
REMOve
REPAir
RINSe
RIP
ROLL
RUB
SAIL
SAVE
SAY
SCORe
SHORt ?
SHUToff
SLEEp
SPIT
SQUInt
STAB
STANd
STARt
STOP
SWIM
TAKE
TEAR
THROw
TO
TOUCh
UNLOck
UNROll
UNSCrew
USE
UWRap
WAIT
WAKE
WALK
WASH
WEAR
WITH
WRAP
YAWN

THANX Northwest Ohio 99ers

WHAT IS A NIBBLE, ANYWAY?

BY JIM SWEDLOW
FROM THE USERS GROUP OF ORANGE COUNTY
A9CUG CALL NEWSLETTER

This month I am going to try and explain all of the various number words we run across. With luck, after you finish reading this, you will have some understanding of bit, byte, nibble, word, hex, binary, and where -31952 really is in memory.
With luck.
Computers really think in binary. In this numbering system there are two numbers, 0 and 1 (or, if you are a computer, off and on). While this works for your 4A, binary is cumbersome for humans. For example, in binary 41,576 is 10100110001 1100.
Hex, or hexidecimal, has sixteen numbers from zero to F. Here are the first sixteen numbers in binary, decimal and hex:

DECIMAL	HEX	BINARY
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

The next number would be 16 or >10 or b10000 (> means hex and b means binary).
One binary number is a bit. Four bits is a nibble. So, 10, A, or 1010 takes four bits or a nibble to express.
A byte is eight bits or two nibbles. With a bit you can count from zero to one. A nibble gets you from zero to fifteen. The range of byte is:

BASE	LOW	HIGH
Binary	0	11111111
Hex	0	FF
Decimal	0	255

You have probably noticed the numbers 16 and 255 when using your TI. ASCII character run from 0 to 255. There are sixteen colors (1 to 16, really 0 to 15). A string can be up to 255 characters long. And on and on.
Before tackling the next thing, a word, lets see if we can decode something. Lets take b10100 or >14. To convert either number to decimal, we need a method:

>14 is >10 plus >4
>10 is 16 and >4 is 4
16 plus 4 is 20
Hence, >14 is 20

b10100 is b10000 plus b100
b10000 is 16 and b100 is 4
16 plus 4 is 20
b10100 is 20

Futher than that I cannot go in this space.
A word is sixteen bits or four nibbles or two bytes. The range of a word is:

BASE	LOW	HIGH
Binary	0	1111111111111111
Hex	0	FFFF
Decimal	0	65,535

But there are no negative numbers. Since we need them, we use something called twos compliment !(which is way beyond the scope of this column and this writer). I can tell you, however, the impact;

Hex range	Decimal Range
0-7FFF	0 to 32,767
8000-FFFF	-32,768 to -1

Remember that >8000 is the next number after >7FFF.
Some examples:
7FFF is 32,767
8000 is -32,768
FFFF is -1
0 is 0
Confused? So was I until I worked with it for a while. These conversion rules may help:
>>Any number less than or equal to 32,767 requires no conversion.
>>Subtract 65536 form any number over 32,767.
>>Add 65536 to any number less than zero.
This conversion process can be expressed in basic:

```
AD=AD+65536*(AD>32767)
```

If AD is the address, this returns the same number if AD is less than or equal to 32767. If AD is greater than 32767, the test returns true (-1) and a negative 65536 is added to AD. Try it on your computer.

Bottom line time. Suppose you see CALL PEEK(-31952,A,B). Where is -31952? Well, since it is less than zero, we add 65536 and get 3584 or >8330. NOW YOU KNOW!

LIBRARY CORNER

This month January meeting the Library will have a GRAB BAG sale. We we are no longer going to keep extra copies of disks programs. Every extra copy of disk programs will be let go at $2.00 per disk.

P.S> This letter was written useing PRINT-IT (4016)

NEW ADDS FOR JANUARY

4106 **PRINT IT** $4.00 /// FREEWARE BY RODGER MERRITT -LABELS, LETTERS, BANNERS, GRAPHICS, BIG, SCRIPT WRITTING. DSSD (525)

4107 **CHARACTERS** $4.00 /// FREEWARE BY RODGER MERRRITT - TWO DISK FULL OF CHARACTERS ,GRAPHICS, SCRIPT & FONTS, CAN BE USED WITH PRINT-IT

4108 **STAR TREK** $2.00 /// FREEWARE BY KEN GILLILAND - 4 STAR TREK MUSIC - 1.THE TV SERIES 2. THE MOTION PICTURE 3. THE WRAP OF KHAN 4. THE SEARCH FOR SPOCK. (SSSD) 358

4110 **WAGNER** $4.00 /// FREEWARE BY KEN GILLILAND FOUR RICHIE WAGNER'S HITS, 1. SIEGFRIELDS FUNERAL MUSIC, 2. PRELUDE TO LOHENGRIN, 3. PRELUDE TRISTAN UNO ISOLDA, 4. TRAFT IHE DAS SCHIFF UN MEESE AN? (DSSD) 609

4032 **FUNNWRITER/3.4** $4.00 /// FREEWARE BY HUNTER VALLEY 99 USER GROUP UP DATE OF THE POPLAR DISK, HAS DM1000 3/5 INCLUDED PLUS OTHER. (DSSD) 648

2374 **XMAS BANNER** $5.00 SIX FOOT BANNER "MERRY XMAS" AND "SEASONSAL GREETING" WITH SANTA GRAPHIC (DSSD) 561

2649 **GAMES #49** $5.00 MANY GAMES WELL DONE, AIRDEFENCE, BUTTERFLY, CRAZYCLIMB, DEVASTATOR, FIDDLESONG, IN 80 DAYS, MARS/REVNG, MARTIANS, MYSTERY SPELL, ORBIT/DEF, R/DUAL, ROBOTCHASE, SPEECH, SPEECH/AID, TALKING/CALC, TI-TOWERS

2650 **GAMES #50** $5.00 GAMES BY TIGERCLUB -AHAMUR, AIR RESCUE, AIR DEFENCE, AIR TRAFFIC, ALIEN DESTOYER, AROW ZAP, ARTILLERY, ATTACKER, BAT ATTACK, BONKERS, BOWLING, DRACULAR, EGG WARS, (SSSD)351

2651 **GAMES 51** $5.00 MORE GAMES FROM TIGER-CUB BATTLESTAR, BLACK JACK2, BOMB SQUARD, FUN HOUSE, GALACTIC, PLANE, PLANES, ROMEO & JUL, SKY RESCUE, SLOR MACHINE, TROGMAN, VERSAILES, ZANQUEST, (SSSD) 360

4111 **STAR** $4.00 FREEWARE BY MICHAEL RICCIO EXECELLENT UTIL PROGRAM, IN ASSEMBLY, TO BE USED IN XBASIC. YOU CAN MOVE LINES, CHANGE LINE NUMBERS, CREATE ASSEMBLY SUBPROGRAMS. 2 (SSSD)673

4112 **FRAC** $2.00 FREEWARE BY STEVE LAUGGUTH -32K, E/A, CREATES MULTICOLOR FRACTAL IMAGES, FANTASTIC COLOR PATTERNS,CAN CREATE YOUR OWN. (SSSD)152

4113 **CATLUG** $2.00 FREEWARE BY MARTY KNOLL JR. - NEED 32K, GRAM KRACKER - CALALOG UP TO 350 DISKS, 2380 FILES, SORTS, PRINTS 1,2,OR 3 COLUMNS, SEARCH AND MANY OTHERS COMMANDS (SSSD)159

2017 **FAST PRINT** $5.00 BY BILL HARMS -A COUPLE OF ROUTINES TO ENTER YOUR SPECIFIC PRINTER CODES FOR SPECIAL PRINTING. USE AS IS OR MERGE. (SSSD)105.

2016 **TEST SYSTEM** $5.00 BY TEXAS INSTRUMENTS- X/BASIC AND MIMI

MEMEORY- TWO DISKS AND 20 PAGE MANUAL FOR TESTING THE 99/4A AND ITS HARDWARES. 2(SSSD)632

LIBRARIAN FRED MOORE
7730 EMERSON AVE.
LOS ANGELES,CA 90045
213-670 4293

```
****************************************
*                                      *
*           BULLETIN BOARD             *
*                                      *
*  99 BBS-- 300/1200 BAUD --24 HOURS   *
*                                      *
*      Five Upload/Download Disks      *
*     (Changed daily, if remembered)   *
*      using TI or Xmodem protocol     *
*                                      *
*          Full Message Base           *
*                                      *
*     (213) 947-7777  (La Habra)       *
*                                      *
****************************************
```

```
****************************************
*                                      *
*           BULLETIN BOARD             *
*            ANNOUNCEMENT              *
*                                      *
*      The first for the LA 99'ers     *
*                                      *
*                                      *
*         Uploads & Downloads          *
*                                      *
*        300/1200 BAUD   24 hours      *
*                                      *
*         Sysop John Williams          *
*           714-495-3964               *
*                                      *
*                                      *
****************************************
```

You sign onto the BBS through your computer. Most of the boards ask users to answer an electronic questionnaire. Some limit access until a prospective user's telephone number is verified.

Some boards specialize in dating or discussion of current events. Most of these boards have listings of other numbers the computer user can call using the modem. Many of the boards are accessible 24 hours a day.

If you need more details before signing up, most of the systems have a "chat" mode in which you can electronically exchange messages with the system's operator. The numbers below will emit the electronic signal needed to get your modem operating.

Marina Match-Up, (213) 397-6300. Dating is perhaps the most widespread activity undertaken on local boards. Like most of the others of its type, this BBS asks callers to fill out a questionnaire. The answers are matched with those of other users, giving the person requesting the match a select group of persons to whom he or she can send an electronic note.

The Talk Channel, (818) 506-0620. Although it has a match function, this BBS stresses exchanging electronic messages. Persons requesting an account fill out a 34-question form.

The Lyceum, (213) 594-9062. Named for the ancient Greek academy where Plato, Aristotle and their brainy friends batted around lofty thoughts, this Lyceum is also dedicated to debate on weighty matters. Computer discussions range from current political issues, such as the U.S. role in the Middle East and Central America, to the sorts of things that probably kept Aristotle up nights, like the nature of the relationship between God and man.

Videoman, (213) 666-8588. This BBS bills itself as "A public-access computer for Hollywood's entertainment industry." Although most of the persons with accounts on the system appear to be from the technical end of show biz, there are a few recognizable names on the user's log, including Ed Shaughnessy of the "Tonight Show" orchestra and Bob Claster, host of "Funny Stuff" on KCRW-FM. This BBS also features a computerized guide to 553 local restaurants.

The Handicapper's Log, (213) 934-6026. This BBS is a computerized racing form for horse players.

The Ground Zero Lobo EMS, (213) 430-0079. This "electronic message system" also has games on line for the computer user who yearns for new types of aliens to zap.

BR's BBS, (213) 394-5950. This BBS offers users the opportunity to collaborate on science-fiction stories. It also has boards for users to comment on politics or other matters.

Mary Jo's Real Estate Genius, (213) 370-0893. One of a number of business-oriented BBS's that are cropping up, this one is a guide to properties in the South Bay.

Fantasy Plaza, (818) 840-8252. This BBS invites computer users to take a stroll through a computer mall. The merchandise ranges from computer accessories to cardboard-and-cellophane "rainbow glasses."

Buy Phone, (213) 470-4679. This BBS likens itself to an electronic Yellow Pages, but its offerings are far fewer. It does, however, have substantial theater, movie and restaurant listings and where to buy, rent, lease or fix an automobile.

MARKETPLACE

(the marketplace is a fund raiser for the club, that is, the "profit" goes to maintain the quality of this Newsletter. In general the price listed splits the difference between cost and retail. Please help your Club.)

MILLERS GRAPHICS

DISkASSEMBLER	18.50
ORPHAN CHRONICLES (priceless)	9.95
ADVANCED DIAGNOSTICS	18.50
EXPLORER	22.50
NIGHT MISSION	18.50
GRAM KRACKER (80K EXPANDED)	185.00
GK UTILITY I	10.00
SMART PROGRAMMING FOR SPRITES	6.25

NEW RELEASES

PRE-SCAN IT! (J.PETER HODDIE)	10.00
GRAM PACKER "	10.00
FONT WRITER "	19.00
PRINTER'S APPRENTICE (M.McCANN)	19.00

MYARC

RS232	82.00
D/D DISK CONTROLLER	155.00
128K RAM DISK/SPOOLER	175.00
512K RAM DISK/SPOOLER	280.00
EXTENDED BASIC II LEVEL IV	80.00
128K RAM DISK W/XBASIC II	235.00
512K RAM DISK W/XBASIC II	340.00

INSCEBOT

TI-ARTIST	15.00
DISPLAY MASTER	12.00
ARTIST EXTRAS	6.00

MEGATRONICS

EXTENDED BASIC II PLUS	72.50
INTERN (BOOK ON GPL)	16.50
128K GRAM CARD	227.50

HARDWARE & SUPPLIES

TEAC 55BV DSDD DRIVES	90.00
DISKETTES DSDD	.50
64K EPSON INT. PRINT BUFFER	45.00
COLOR RIBBONS (EPSON)	4.00

BACK ISSUES

SUPER 99 MONTHLY	1.25
MICROPENDIUM	1.25
SMART PROGRAMMER JUNE 1986	1.50

BEST OF NEWSLETTERS W/DISK	5.00
FORTH NOTES VOL 1-5 (2.50 EA)	10.00
BEGINNER'S FORTH NOTEBOOK	2.50
ASSEMBLY NOTES VOL 1	2.50
TECHNICAL AND BUSINESS BOOKS	5.00
SAMS BOOKS (VARIOUS)	5.00
SAMS BOOKS WITH CASSETTES	7.50

(please send your order to the CLUB address, not the Librarian, and add $1.00 per disk for postage and handling. CA residents add 6.5% tax).

MYARC
T.M.

GENEVE TM

IMPORTANT NEW FEATURES OF MYARC'S Model 9640 FAMILY COMPUTER

TMS9995 u-Processor Runs 3-4 Times Faster Than the 99/4A

- Runs same instruction set as 9900 used in 99/4A plus 4 new ones
- Pipelined processor (i.e. u-processor performs several functions simultaneously

V9938 Advanced Video Display Processor

- Is software compatible with TMS9918A (used in 99/4A)
- Uses 46 registers for high speed "HARDWARE" graphics commands
- Commands include: DRAW SEARCH POINT(status) BLINK FILL MOVE Animation And more
- Uses Color Pallet of 512 colors on the screen at a single time
- 7 modes of graphics operation; some modes allow 256 colors
- True BMG (Bit-Mapped-Graphics) operation
- Both composite (like the 99/4A) and analog RGB outputs (like the Atari ST and Commodore Amiga)
- Supports up to 256 colors per screen in the 256 by 424 mode or 16 colors in the 512 by 424 mode
- Comes with 128K bytes of video RAM (8 times the amount of the 99/4A)

Real-Time Clock Chip

- Gives you and your program instant access to date and time

Sound Chip

- Compatible with 99/4A (i.e. 3 simultaneous tones, 1 noise)

Awesome amount of RAM

- 512K of CPU RAM (user configurable between CPU-RAM, RAM-DISK or PRINT-SPOOLER)
- Expandable to 1 megabyte with MYARC 512K Card
- In 99/4A mode 64K of the 512K becomes GROM and 16K CARTRIDGE ROM

Built-In Mouse Interface

- Installed hardware allows for the MS mouse to be connected directly to the 9640 board
- Basic language support for the mouse built in. Uses the industry-standard MacIntosh mouse commands

(over)

MYARC, INC., P.O. BOX 140, BASKING RIDGE, NEW JERSEY 07920 (201)766-1700
MICROCOMPUTER ARCHITECTS

Standard Joystick Interface

* Joystick interface is compatible with the one used by the 99/4A

Hardware and Software Support for the most commonly-used peripherals

* Floppy Disk Controllers include MYARC, Texas Instruments, and Corcomp
* RS232 cards include MYARC, Texas Instruments, and Corcomp
* Ram-Disks include Horizon

Software Support Supplied with the 9640

* MYARC DOS (similar to MS-DOS 2.1)
* MYARC ADVANCED BASIC
    - Compatible with TI Extended BASIC and MYARC Extenced BASIC II
    - Supports all modes of the Video Processor (including 80 column)
    - Supports Windows
    - Supports easy to program Mouse Commands
    - Combined Text and Bit-Mapped-Graphics Modes
    - Drawing Commands such as Circle, Rectangle, etc. are built-in
* Program patches to make TI-Writer 1) more powerful
2) display 80 columns
* Program to SAVE your 99/4A cartridges to disk

SOFTWARE SUPPORT FROM OTHER SOFTWARE VENDORS

PECAN Systems

* UCSD Pascal Runtime (included free with the 9640)
* UCSD PROGRAMMING LANGUAGES (at additional cost)
    - UCSD Basic
    - UCSD COBOL
    - UCSD Fortran
    - UCSD Pascal
* Plus thousands of other applications ranging from Pig Management to Office Management to Home Education

DataBioTics

* The Music Shop
* Super-Super 4TH
* Super Word
* The Terminal Connection
* The Professional Business Assistant
* Macro Assembler
* Pilot
* Lush Brush

Paul Charlton

* Fast-Term II

Clint Pulley

* Big C Compiler

INSCEBOT

* TI-ARTIST
* MacPaint Equivalent

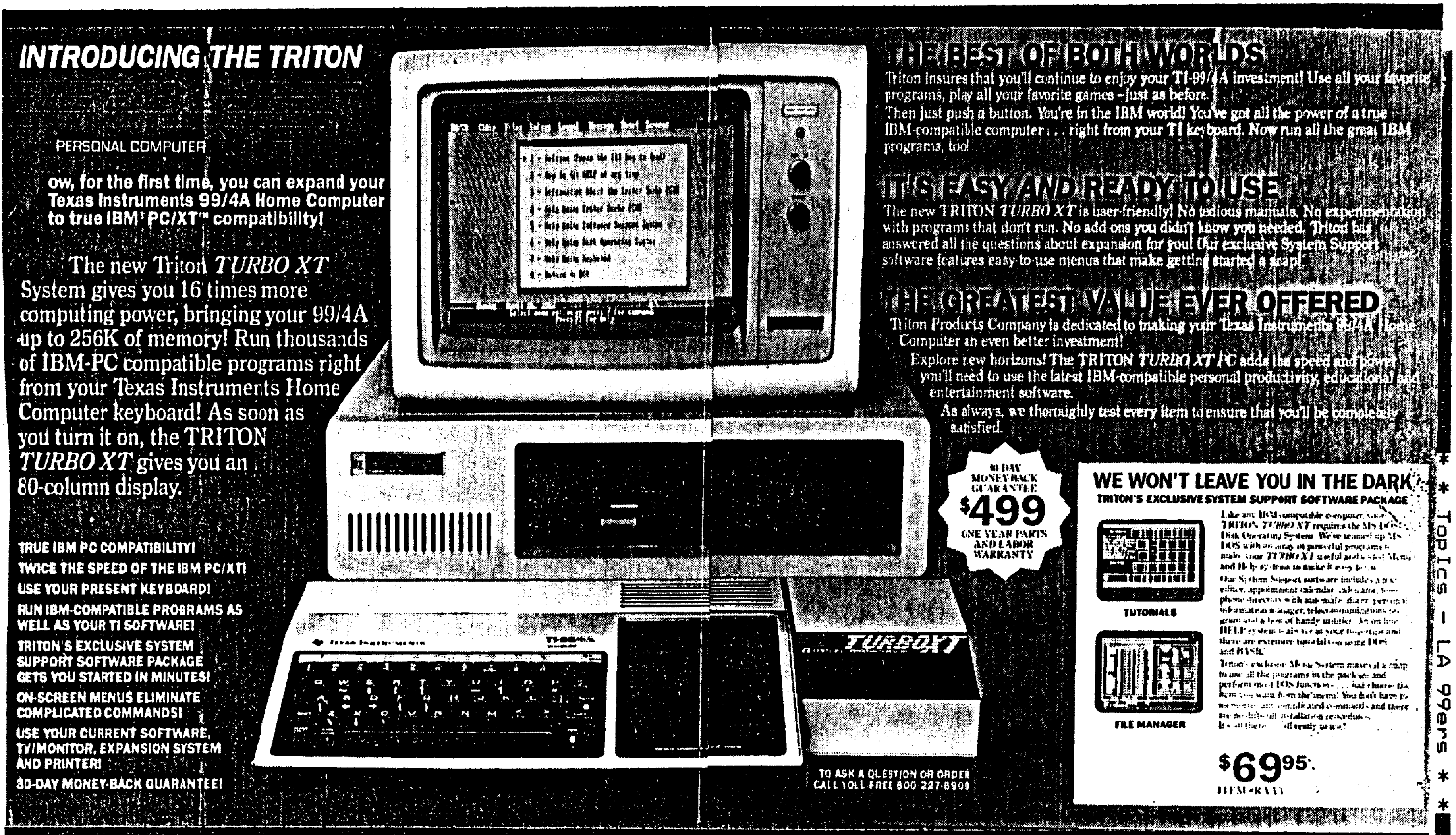
Pike Creek Computer Co

* General Purpose Accounting Software

BRYGHT-DATA

* Professional Business Acc'ting Software

CSI Design Group



CLUB OFFICERS

President:
Terrie Masters
271-693Ø

Vice President:
George Steffen
329-3527

Secretary:
Terry Wilson
563-3869

Treasurer:
Margaret Hutton
541-5359

Editor:
Terrie Masters
271-693Ø

Membership Chairman
Tom Freeman
454-1943

Librarian:
Fred Moore
67Ø-4293

Library Assistant:
Chick De Marti
532-8499

Library Assistant:
Alan Whiteman
379-8Ø31

Equipment Chairman:
Joe Fierstein
377-9834

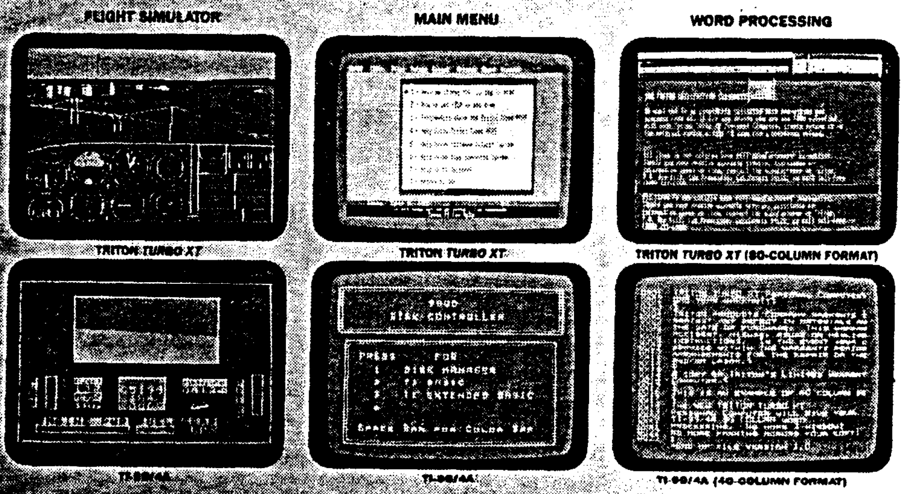
MISCELLANEOUS TI EQUIPMENT FOR SALE

Please Call
Bill Goyette

213-215-19Ø9

A PICTURE IS WORTH A THOUSAND COMMANDS!

COMPARE THE DIFFERENCE THE TRITON *TURBO XT* PERSONAL COMPUTER MAKES.



SPECIFICATIONS

TRITON *TURBO XT* Personal Computer

MICROPROCESSOR: Intel 8088, 8/4.77 MHz clock speed (software selectable).

OPERATING SYSTEM: MicroSoft* Disk Operating System (MS-DOS)

MEMORY: 256K RAM (Expandable to 640K).

DISK DRIVE: One 5¼" double sided, double density, 360K thin-line mini-floppy, 48 tracks per inch.

VIDEO: RGB/composite color graphics display adapter.

INTERNAL EXPANSION: Eight standard user-accessible IBM PC card slots.

EXTERNAL CONNECTIONS: Standard parallel printer port, composite video, RGB, AC outlet.

POWER: 120VAC, 50-60 Hz.

BRIDGE BOX

4A MODE: Common video out, 99/4A video in.

XT MODE: Keyboard out, video in, XT Video in. Selectable power-up mode callable from TI BASIC or Extended BASIC. Concurrent processing. Five LED status display.

TRITON GUARANTEES THAT YOU'LL BE SATISFIED. WE HAVE OUR NAME ON IT.

Hundreds of thousands of satisfied TI-99/4A owners have found Triton the best source for peripherals, software and helpful advice. We back every purchase with our toll-free *Help Line*. It's very important that you have support. Should you ever encounter a problem with anything you purchase from Triton, our friendly and knowledgeable operators can usually sort things out right over the phone.

THE TRITON MONEY BACK GUARANTEE

We want you to be completely satisfied with everything you purchase from us. If for any reason you are not 100% satisfied with your TRITON *TURBO XT* Personal Computer, simply return it in the original carton within 30 days for a courteous refund.

THREE YEAR EXTENDED WARRANTY

Another Triton first! For less than one hundred dollars, you can insure that you'll always have your *Turbo XT* up and running! Our extended second and third year warranty simply continues the first year's coverage two more years. Wouldn't it be comforting to know that you don't have to worry about breakdowns, labor or repairs . . . for three full years! It's only $99.95. (Order Item #RAAB)

THE FUTURE GENERATIONS

Because of the tremendous growth of the personal computer industry, Triton has a full-time product management department dedicated to researching and testing all of the new programs and peripherals that flood the marketplace. Only the very finest are selected and recommended to you.

As in the past, you will be kept informed of new and exciting products for your new TRITON *TURBO XT* PC with our full-color catalogs. As always, each issue features the same great values that you've come to expect.

WE'RE AS NEAR AS YOUR PHONE. TO ORDER OR ASK A QUESTION, CALL TOLL-FREE

800-227-6900

30-DAY MONEY-BACK GUARANTEE
THE NEW TRITON *TURBO XT*
Personal Computer
ITEM #RZAA
$499
PLUS $19.99 SHIPPING CHARGE
Due to the anticipated demand, orders will be filled on a first-come, first-shipped basis. [illegible]

Board Meetings for the LA 99ers is held on the 1st Tuesday of each month at MERIT SAVINGS at 18501 Western Ave. Gardena The meetings starts at 7:15. *** EVERYONE IS INVITED ***

Los Angeles 99er Computer Group: Forth Wednesday of each month, 7:15 PM, at Torrance Library, 3031 Torrance Blvd., Torrance

Pomona Valley 99ers Computer Group: Second Monday of each month, 7:00 PM, at Cortez Elementary , 12750 Carrisa Ave, Chino. Call Joy Warner, 982-9971, nights.

San Fernando Valley 99er Computer Group: Second Tuesday each month, 7:30 PM, Doctor's Conferance Room, Sherman Oaks Community Hospital, 4929 Van Nuys Blvd., Sherman Oaks.

San Gabriel Valley 99/4 Users Group: First Wednesday of each month at West Covina Public Library, 1601 W. Covina Parkway, West Covina.

Users Group of Orange County: Third Thursday of each month, 7:30 PM at Westchester Community Service Center (1 block east of Beach Blvd.), Jackson and Westminster, Westminster, CA.

* The first line on the address label shows the last issue you will receive for members or the last issue received by us for exchange.