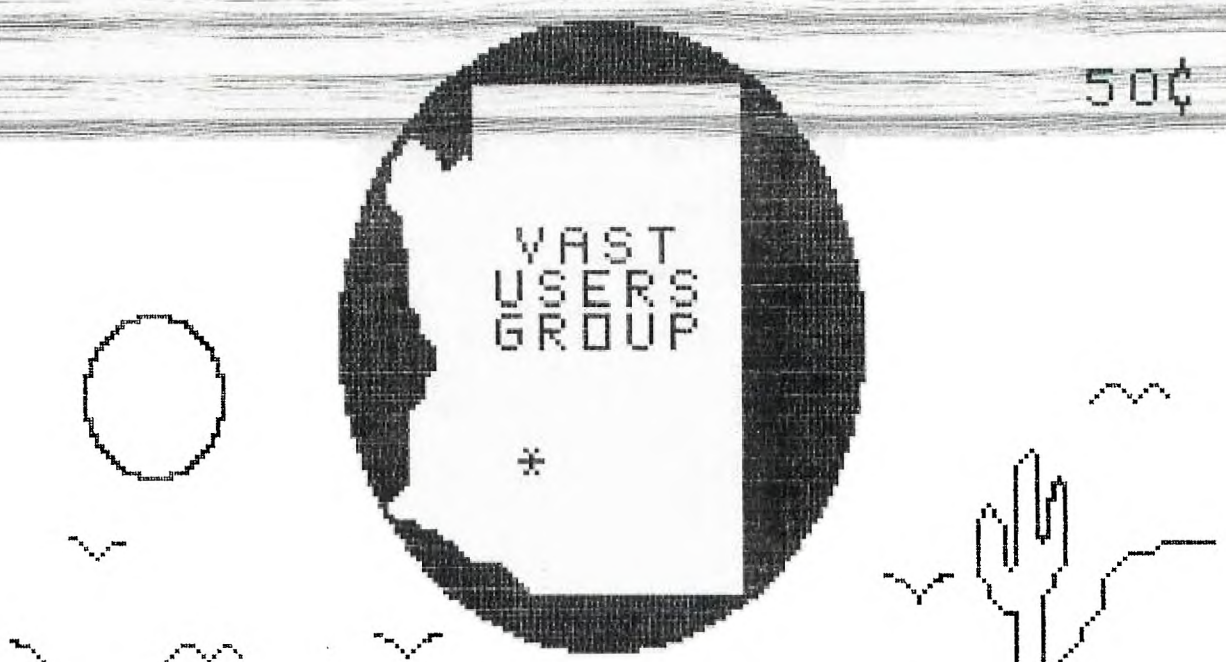
8710 099

50¢ VasA



VALLEY OF THE SUN USER GROUP newsletter

VOL.3 OCTOBER 10,1987 NO.10

* *

CONTENTS

VAST 99 INFO..... 2
SECRETARY'S SLATE. 3
COMPUTER TUTOR.. 4
VIEWS & REVIEWS. 5
WHEREFORTHS..... 6
HINTS AND TIPS..... 7
EDITORS PAGE..... 9

VAST 99 BBS 437-4335

OCTOBER, 1987

VAST USER GROUP INFORMATION

The **VAST USERS' GROUP** is a support group for Home Computer users. We primarily support the TI-99 Home computer and compatables, but all computer users are welcome. Our regular meetings are on the second Saturday of the month. We meet in downtown Phoenix at the Park Inn International Hotel at 401 N. First Street in the Phoenician Room. The meetings start at 10:00 AM and continue until 11:00 AM with socializing starting at 9:00 AM. The yearly membership fee is $6.00.

All meetings are open and anyone may attend. Only dues paying members may vote in elections and obtain programs from the Users' Group library.

The current officers are:

President
Bob Nixon.....897-6204
Vice-President
Ike Van Kampen.....934-5164
Secretary
Stu Olson.....846-7624
Treasurer
Earl Bonneau.....269-3802
User Group Librarian
Earl Bonneau.....269-3802
Newsletter Editor/BBS SysOp
Jim Ely.....921-3375

****

A FORTH Tutorial is being conducted by Rene' LeBlanc in this newsletter. It consists of a continuing series of articles relating to his version of FORTH which is available from the User Group Library. For more information, please contact him at (602) 991-1403.

The Users' Group's BBS is now in operation 24 hours a day. Contact it at (602) 437-4335. There is a lot of interesting conversation and information available here so give it a try.

Deadline for submission of articles or advertising for the Newsletter is the last Saturday of every month. Articles may be submitted in any form, however, the preferred method is by phone transfer directly to the Editor.

****

Advertising rates are as follows:

Commercial:

Full Page $10.00
Half Page $ 7.00
Quarter Page $4.00

Personal:

Four lines,
30 Characters/line
$1.00
$.20 per line
over four.

All rates are for **ONE** issue only!

****

Programs are available from the USERS' GROUP LIBRARY at the following rates:

SS/SD Disk $2.00
DS/SD Disk $4.00

If copying of documentation is required, it will be at the rate of $.10 per page. If the User Group supplies the disk, please add $1.00 to the above charges. An exchange program for free programs is also in effect. Please contact the librarian for further information. A complete list of what is in the library is available on 2 disks free of charge if you supply the disks or for $1.00 per disk if the User Group supplies the disks.

****

Valley of the Sun T Users Group

****

MINUTES
VAST User Group
September 12, 1987

The meeting was called to order by Bob Nixon, President, at 10:03. A motion was made and passed to dispense with reading of the minutes from the August meeting.

Bob called for old business and Bob Koons announced that he had not found an attorney to work on the project of getting the U.G. registered with the state. Gerry Kennedy indicated he has possibly found one to help out with the legal matters and will report on the progress at the next meeting. There was no other old business to discuss.

New business was introduced concerning the various committees that are needed. The various committees were announced and Bob first asked for any volunteers for the positions of chairman for the committees. A discussion followed as to what the basic responsibility of the chairman (within the different committees) would include.

The following people volunteered for the chairman positions of:

TI-Committee - Bob Noel
Other computer committee - Stu Olson

Bob Koons was selected as the chairman of the Information committee. The committee would be responsible for the gathering of news and info pertinent to the group (ie, upcoming computer related events, new equipment releases, etc).

Although Jim Ely was not present, Mike Marfisi indicated he would help Jim with both the BBS and newsletter committees.

A discussion was brought up concerning the use of the BBS to post a listing of the various committees and their responsibilities. This way, users logging onto the BBS could read through them and make a decision as to which area they would like to help out in. Mike indicated he would help put together some info to post on the BBS concerning this.

A discussion on club dues was brought up. Several items were discussed basically revolving around the price we are currently requesting, $6.00 per year. An idea to increase the dues and include the newsletter as a free item was brought up. Also, some members indicated they may prefer to have the newsletter mailed to them which would be beneficial if they could not attend the meeting to obtain one. A suggestion was made that those members desiring special mailings could possibly have the cost of postage included in their dues.

Due to the relatively light turn out (27 members in attendence), a motion was made to table the discussion until the following meeting in hopes that more members would be present to participate in the discussion.

Bob Koons suggested a topic for the next two meetings. He discussed a similiar program given by the Sun City Computer Group where they invited a local distributor to a meeting and had him assemble an IBM clone type computer from scratch. Since this falls into the area of the Other computer committee, Stu announced he would see to the details and have it ready for the next meeting.

The meeting was closed at 10:45 a.m.

Stu Olson
Secretary, VAST User Group

The Elements of Basic

Part 3

By Dave Howell

Courtesy "Erie 99'er Newsletter"

NUMERIC DATA

Most computers use the floating decimal point as the method of representing numeric data. Any number of digits can be placed on either side of the decimal point. Even with numbers having no decimal position, a decimal point is always assumed following the number's last digit.

Floating point numbers are displayed with 10 digits of accuracy. For example, the following entry of a 10 digit number would generate a 10 digit display:

```
PRINT .5666666666
 .5666666666
```

If an 11 digit floating point number was entered, the last digit would not be displayed and the number would be rounded as follows:

```
PRINT .56666666666
 .5666666667
```

Commas may not be included with numeric data. For example, 109700 would be a valid entry, while 109,700 would be treated by the computer as two separate entries -- 109 and 700.

Floating point numbers include signed integers as well as numbers with decimal points. Examples:

```
-.0789 5 7000 0 +.000001 67.98
```

Negative floating point numbers must be preceded by a minus sign (-). Positive floating point numbers can be preceded by a plus sign (+) but a number is assumed positive if it doesn't have a sign.

TI BASIC uses scientific notation to express integers containing 11 or more digits. The following are examples of floating point numbers expressed in scientific notation:

```
PRINT 12345678901
 1.23457E+10

PRINT -575000000000
-5.75E+11

PRINT .00000066666666
 6.66667E-07
```

In the first and last examples, the number is truncated at the 6th non-zero digit. Any additional digits are rounded off.

Numbers cannot be displayed with an exponent greater that 99. Numbers that have exponents greater than 99 are displayed with two asterisks in place of the exponent. Example:

```
2.47685E+**
```

Numbers that require an exponent that is less than -99 are automatically converted to zero. Further the computer cannot manipulate numbers that have an exponent greater than 127. As a result, the following warning will be displayed:

```
* WARNING
NUMBER TOO BIG
```

DH

OCTOBER 1987

VIEWS AND REVIEWS

REVIEW OF CERTIFICATE 99
GREAT LAKES SOFTWARE

By Mike Marfisi

Certificate 99 is designed to allow you to create and print an unlimited variety of certificates, awards and diploma style printed pages. You can also make signs and posters. It operates out of XB, E/A or TI Writer and requires an Epson/ Gemini type printer. You can print your certificates using any of 6 built-in fonts, 24 graphic and 12 borders.

The program comes to you on a SSSD "flippy" disk. The main program is on side 1 and the files are on side 2. Once the program is loaded, the title screen appears and you are told to flip the disk over to continue. Pressing enter at this point brings the "font selection" menu into view. Select one of the 6 fonts. You are then asked if you would like the text that you will enter to be automatically centered. Border selection is next. Choose one from the menu or use none, as you please. Hitting the space bar will cycle you through the selections on the border and graphic menus. You get to choose from 24 graphics. You may magnify them and place them anywhere on the screen.

Provision is made to enter a signature line with one of several gag signatures like the "Prez" and other "notables". Or leave a blank line for your signature or don't use one at all.

After all these choices have been made, it is time to compose your prose. The text screen is made up of two sections. These sections are miniature versions of the final product so you can see just what your end result will look like. The top section is used to place and enter large type and the lower for small type. The screen shows the position of any graphic or signature line so you can't overwrite these areas. Just type in the text you want and you are all set.

When you have reached the last line of text, hitting the enter key will take you to the print screen where you are asked to input your printer command (PIO, RS232, etc.). That's all there is to it. Sit back and wait while your certificate is formatted and printed.

After the page has printed, the program asks if you want to continue or quit. Whichever option you choose, the program will then save the last certificate to memory. When you boot the program next time, you will have the last thing you produced as your starting point. You can either print it again or overwrite it and start over.

The program is very easy to use. There are no complicated steps. The instructions are easy and straight forward. The program comes to you with a dozen sheets of "parchment" paper and some gold seals to jazz up your certificate. If you have a need to produce certificates, awards, quick signs etc., this may be a good program to add to your collection. Price tag is $19.95 from Tex Comp or Great Lakes Mfg.

Overall rating for this program is B+ (some of the graphics are dumb! Why would anyone want a Pepsi can on an award?). I will say that the manufacturer says you can add to ... change the graphics with a prog... like Joy Paint 99.

Value rating...B+ (great if you need this type program! questionable at $19.95 if you just want to play.

Documentation...A+.

MM

OCTOBER, 1987

WHEREFORTHS OF FORTH

This will be the final issue of WHEREFORTHS OF FORTH. The last several issues have described a program to copy disks using only a single drive. This was meant to be a fairly realistic example of Forth programming that re-used a number of utility words which were introduced in earlier issues of WHEREFORTHS.

WHEREFORTHS #19, published in the August, 1987 issue of the VAST newsletter, finished the description of the main words used by the COPY-DISK program.

I told you that in this issue of WHEREFORTHS I would review some of the other supporting words of COPY-DISK that provide some of the user input/output niceties of the program.

Well, I have changed my mind. I think the words used are all simple enough that if you are really interested in the details, you will be able to figure them out for yourselves. If you can't figure something out, you can feel free to call me up or see me at a future meeting and ask about it.

Instead of explaining a word-by-word description of how the various words of COPY-DISK work, I feel more like waxing philosophical about programming in general and FORTH in particular.

Many of you are computer USERS, but do not get into the details of writing your own programs. That is a perfectly valid choice. Computers can do all sorts of amazing things just using programs that have already been written by others. As long as SOME people write good general-purpose programs the rest of us can buy or "borrow" or out-right steal, we can spend most of our computer time using those programs written by others.

Writing programs is a very technical activity, and it takes a lot of time, knowledge and skill. But it is also a very satisfying activity, and we don't have to start out by writing a large complex application program. We can write small and simple programs just for the intellectual exercise, like working cross-word puzzles, and gradually work our way up to specialized useful programs that do exactly what we want them to do for our own purposes.

One of the reasons BASIC is a popular language is that it is INTERACTIVE. You can type in a BASIC statement and when you hit ENTER it will try to execute. Most languages need to be compiled into machine code before they can execute. Another strength of BASIC is that it is a HIGH-LEVEL language that provides very easy access to your computer's input/output functions, file system, graphics and sound, and powerful string-manipulation, and mathematical operations.

BASIC is an INTERPRETED language, meaning that in the computer is ALREADY a program running that is watching your inputs and interpreting them as you enter them. What you pay for this convenience is SPEED. The computer does a lot of extra work, first figuring out what you have typed in, and then DOING it.

With COMPILED programs, like assembly language and C, you must perform an intermediate step of transforming the source code into machine code. Then, when you LOAD the machine code into the computer, you can execute it and have it do what the program was designed to do.

Since the machine code can simply be executed directly, there is no intermediate step of interpreting some human-oriented ASCII text and then translating that into the proper actions. This makes compiled languages MUCH FASTER than BASIC.

The Forth language has some of the advantages of BOTH compiled AND interpreted languages. It has an interpreter which very efficiently executes words which have previously been compiled into the Forth "dictionary". Forth source code that is written to compile new words into the dictionary can be compiled a word at a time: unlike assembly language and C where you need to have a complete source program all written and packaged up as a source file before you can begin to compile it.

Typical Forth systems compile so quickly that it isn't even necessary to keep object code around. You can simply LOAD it in from source, thereby compiling it as you LOAD, and then run the program. TI-FORTH, which is based on the FIG (Forth Interest Group) Forth primitives, is a notoriously slow compiling system, as Forth systems go, but even so, it compiles very quickly compared with most languages.

Because TI-Forth does compile relatively slowly as Forth systems go, TI provided the binary save (BSAVE) and binary load (BLOAD) functions which allow you to save a compiled object form of your program, and then load it in later to run it when you wish without recompiling it again.

Due to the threaded-dictionary structure of Forth object code, it occupies less memory than most programs of comparable functionality written in other languages, and Forth execution speed is quite fast. It is a bit like "having your cake and eating it too."

However, Forth is not as easy to learn as BASIC. It has a lot more primitive functions, and you need quite a lot of practice to be able to make effective use of them. You need more insight into what is really going on down in the machine. Even so, it is still much less tedious than as-

CONTINUED ON PAGE 8 >>

OCTOBER,1987

BITS AND PIECES

TMS 9900 ASSEMBLY LANGUAGE TUTORIAL

PART 9

By Steve Royce

Courtesy WNY 99'ers

A DSRLNK FOR EXTENDED BASIC

AUTHORED BY JOHN CLULOW

Device Service Routine Link by Texas Instruments, Modified by John Clulow, 1982.

Because Extended BASIC utilities don't include a DSRLNK, such a utility must be provided separately for Ext BASIC routines which access, for instance, RS232 I/O. The routine shown below was produced, in part, from a source program supplied by Texas Instruments and in part from a consideration of the DSRLNK utility in the Mini Memory Module.

If the DSRLNK object code is loaded in the Extended BASIC program IMMEDIATELY after the CALL INIT statement, then the assembly program utilizing DSRLNK should use the equate DSRLNK EQU >2532

```
        DEF  DSRLNK
SCLEN   EQU  >8354
SCNAME  EQU  >8356
CRULST  EQU  >83D0
SADDR   EQU  >83D2
GPLWS   EQU  >83E0
VSBR    EQU  >2028
FLGPTR  DATA 0
SVGPRT  DATA 0
SAVCPU  DATA 0
SAVENT  DATA 0
SAVLEN  DATA 0
SAVPAB  DATA 0
SAVVDP  DATA 0
NAMBUF  DATA 0,0,0,0,0
DLNKWS  DATA 0,0,0,0,0
TYPE    DATA 0,0,0,0,0,0,0,0,0,0,0,0
C100    DATA 100
H20     EQU  $
H2000   DATA >2000
DECMAL  TEXT '.'
HAA     BYTE >AA
DSRLNK  DATA DLNKWS,DLENTR
DLENTR  MOV  *R14+,R5
        SOCB @H20,R15
        MOV  @SCNAME,R0
        MOV  R0,R9
        AI   R9,-8
        BLWP @VSBR
        MOVB R1,R3
        SRL  R3,8
        SETO R4
        LI   R2,NAMBUF
LNK$LP  INC  R0
        INC  R4
        C    R4,R3
        JEQ  LNK$LN
        BLWP @VSBR
        MOVB R1,*R2+
        CB   R1,@DECMAL
        JNE  LNK$LP
LNK$LN  MOV  R4,R4
        JEQ  LNKERR
        CI   R4,7
        JGT  LNKERR
        CLR  @CRULST
        MOV  R4,@SCLEN
        MOV  R4,@SAVLEN
        INC  R4
        A    R4,@SCNAME
        MOV  @SCNAME,@SAVPAB
SROM    LWPI GPLWS
        CLR  R1
        LI   R12,>0F00
NOROM   MOV  R12,R12
        JEQ  NOOFF
        SBZ  0
NOOFF   AI   R12,>0100
        CLR  @CRULST
        CI   R12,>2000
```

CONTINUED ON PAGE 8

HINTS AND TIPS CONTINUES FROM PAGE 7

```
        JEQ   NODSR
        MOV   R12,@CRULST
        SBO   0
        LI    R2,>4000
        CB    *R2,@HAA
        JNE   NOROM
        B     @TYPE,R2
        JMP   SGO2
SGO     MOV   @SADDR,R2
        SBO   0
SGO2    MOV   *R2,R2
        JEQ   NOROM
        MOV   R2,@SADDR
        INCT  R2
        MOV   *R2+,R9
        MOVB  @SCLEN+1,R5
        JEQ   NAME2
        CB    R5,*R2+
        JNE   SGO
        SRL   R5,8
        LI    R6,NAMBUF
NAME1   CB    *R6+,*R2+
        JNE   SGO
        DEC   R5
        JNE   NAME1
NAME2   INC   R1
        MOV   R1,@SAVVER
        MOV   R9,@SAVENT
        MOV   R12,@SAVCRU
        BL    *R9
        JMP   SGO
        SBZ   0
        LWPI  DLNKWS
        MOV   R9,R0
        BLWP  @VSBR
        SRL   R1,13
        INC   @CRR
        RTWP
NODSR   LWPI  DLNKWS
LNKERR  CLR   R1
IOERR   SWPB  R1
        MOVB  R1,*R13
        SOCB  @H20,R15
        RTWP
        END
```

WHEREFORTHS OF FORTH CONTINUES FROM PAGE 6

sembly language. With POWER comes RESPONSIBILITY!

Forth is FAR more powerful than BASIC in the sense that it can access ALL the capabilities of your computer as can assembly language. It does not try to give you an over-simplified view of the machine, and thereby hide much of its power from you. But to use that power, you have to understand it! Ah, there is no free lunch!

The nice thing about Forth that helps counteract its potential complexity is its extreme modularity. In Forth, we write large programs by creating layers of small procedures (called "words") which can be coded and tested interactively, usually by themselves. As we code each new word, we then test it by typing the required parameters on the stack and then the word name and see what it does, all interactively. If it performs the desired action, we go on to the next word. If not, we interactively debug the word under test. Most Forth words comprise only a few lines of code, so we always deal with a little piece of the program at a time. When we get the underlying pieces coded and debugged, we then build higher-level words upon them.

I tried to demonstrate this in the various examples I have provided you throughout the WHEREFORTHS series. I hope I have been able to provide you with some insight into this unique and very practical programming language.

Rene' LeBlanc

NEXT MEETING NOV. 19!

From the Editors Desk

THANKS.....

To Mike Marfisi, who actually volunteered to do a Newsletter for September! Pretty nice Newsletter at that.

IN THIS ISSUE....

Mike Marfisi has done a review of a new program he has received called CERTIFICATE 99. It is on page 5. Computer Tutor on page 4 talks about how the TI handles Numeric Data. On page 6 is the last edition of WHEREFORTHS OF FORTH by Rene' LeBlanc. Rene' has been a regular contributor to the Newsletter since the beginning. Thanks, Rene'. Sorry to see the WHEREFORTHS articles end, but when the response to your work is so little, I can sure understand. Part 9 of our 10 part series on Assembly Language is on page 7, this time with a DSRLNK from Extended Basic. Hope you enjoy this issue.

NEXT MONTH.....

We will have the last part of our Assembly Tutorials, an Extended Basic program that turns your TI into a burglar alarm system and some more In the Computer Tutor series on programming in Basic and Extended Basic. You know, YOUR article could be included here, also.

MYARC 9640 COMPUTER UPDATE

What I would really like to see here is an article written by Jerry Liddell or Dan Shell (both have 9640's and are keeping up with all the goings on). But I never hear from either of them, so for now, some info taken from MICROpendium.

MY-WORD, the MYARC version of TI-Writer for the 9640, and Multiplan are said to be 100 percent complete. Multiplan has as much speed as the IBM version and 56K of memory for spreadsheets. Also, some bugs are gone -- the tabbing key can now be used between fields. The artist program, MY-ART, is also complete.

DOS is said to be complete except for batch commands and if no major errors occur, will be available in early October (anyone heard anything about this yet?).

10 of the new floppy-and-hard disk controller cards are being Beta tested. Production was due to start October 1, and shipments to dealers by Mid October.

MYARC is planning a one time direct mailing of software to everyone from whom it received a warranty registration (MDOS, MY-WORD, Multiplan, Advanced Basic and Pascal). This mailout is scheduled for late October. Documentation updates will also be available (how the new software varies from TI-99/4A software).

More info as it is becomes available...

ONE LAST THING...

I heard a rumor that Gerry Kennedy's modem broke late last month and that he was having such withdrawl pains that he called Stu Olson and asked him to make "modem sounds" to him over the phone. Any truth to that rumor, Gerry? Glad to see you got your modem fixed!! By the way, was that number a 976- type number?

That's it for this month

Jim Ely, Editor