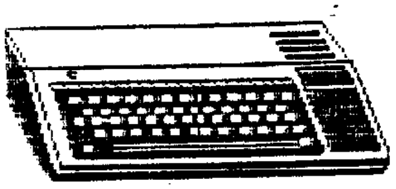
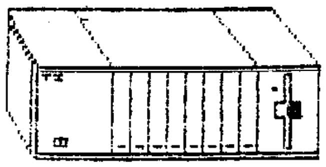
NEW JUG NEWS



NEW JERSEY USERS GROUP



Vol.5 No.4 *Monthly Publication of the New Jersey Users Group* APRIL 1986

****

MEETING

****

APRIL 14 MONDAY 7:00

7:00 - 8:00 BASIC SIG WILL MEET

8:00 - GENERAL MEETING--REPORT ON T.I.C.O.F.F
POTTER & SHULDMAN on analog and digital conversions

****
The New Jersey Users Group meets on the second Monday of each month in the Metuchen Library. Dues are $15 per year.
****

OFFICERS

President............	Steve Citron..686-5619
Vice-Presidents......	John Bonito...653-2637
	Bob Costello..663-4512
	Mel Gary......828-5407
	Bob Guellnitz.382-5963
Secretary............	Carol Sudol...494-3781
Treasurer............	Marv Shuldman.821-8158
Newsletter Editor....	Mel Gary......828-5407
Software Library.....	Dave Green....463-9133
	Leon Green....828-2435
Advanced Prog. Sig...	Jay Holovacs..356-3150
Basic SIG............	Bob Haefeli...572-2828

****
SUBSCRIPTION FREE WITH PAID MEMBERSHIP, TO USERS GROUPS AND SELECTED VENDORS
****

APRIL 1986

SUNDAY	MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
		1	2	3	4	5
6	7	8	9	10	11	12
13	14 GENERAL MEETING	15	16	17	18	19
20	21	22 STEERING COMMITTEE	23	24 ADVANCED PROG.	25 NEWSLETTER DEADLINE	26
27	28	29	30			

President:	Send Dues To:	Write For Application:
Steve Citron	Marv Shuldman	Bill Dubrow
981 Townley Ave.	28 Tyndall Rd.	21 Seaward Ave.
Union, NJ 07083	Kendall Pk., NJ 08824	Metuchen, NJ 08840

NOTICE!

The Software Library Committee, consisting of Dave Green and Leon Green, is working arduously to reconstruct the club's software library. They are now in the process of collecting all of NEW JUG's software that has been loaned out to others, either officially or unofficially. They are asking that all such material be returned as soon as possible so that they may proceed. They are also asking that you contact either of them if you have programs that you are willing to have placed in the Software Library. And they are currently in the process of formulating policies related to the functioning of the Library so as insure its efficient operation. These recommendations will be presented at the next meeting of the Steering Committee. Now that we finally have TWO persons willing to run the Library, let's give them all of the help that we can! After all, what is a Users Group without a Software Library to benefit its members?! Dave Green may be reached at 463-9133 and Leon Green at 828-2435.

MULTIPLAN

by Tom Kennedy

ELECTRONIC SPREADSHEETS...CELLULAR REFERENCING...ABSOLUTE REFERENCING...CELLULAR ANALYSIS... WORKSHEETS... FORMULAS

These are buzzwords of a form of Data Processing that on the surface appears to be incomprehensible to all but accountants and the bridge crew of the Star-Ship Enterprise. As Word Processing is to writing a letter, Data Processing is to using a multiplication table.

Many people have a hard time using spreadsheets, because working with data in this format is similar to learning a new language. But once you learn to use the commands, and the procedure of working with data in a two-dimensional row/column format instead of a one-dimensional equation, you'll find many ways in which Multiplan will allow you to "crunch numbers" faster and easier than using a calculator and notebook. More than that, Multiplan is flexible enough to be used anytime you want to display, or use, numbers or words in a row/column format. In fact, you could even adapt Multiplan to use it as a Word Processor!

What is a spreadsheet? In business, you often hear reference to "our books". The "books" that the businessmen, and we, keep are a pen paper record showing the Debits and Credits of various bills paid and assets gained, plotted against a scale of time. Each intersection of row and column contains an entry for a value. The last column and/or last row contain a summation of all previous columns or rows. In an electronic spreadsheet, you recreate the printed form with the addition that each "cell" (a row/column intersection) can also contain a mathematical equation, or "formula", that automatically acts upon pre-defined cells and displays the result accordingly. If any value in any cell is changed, the formula instantly updates displayed results. This self-maintenance ability is what pays off in using an electronic spreadsheet, such as Multiplan.

To operate the Multiplan software on the TI, you must have 32K memory expansion, and at least one disk drive. An RS232 card and a printer are also handy, but not mandatory because unlike word processing, where the end result is a printed piece of paper, the end result with a spreadsheet is useful data, which may be used many ways. Most worksheets are well over 80 characters in width, (up to 2016!) and this requires a cut-and-paste job, so a wide-carriage printer is preferable.

To load Multiplan, you insert the cartridge and program disk, select Multiplan from the menu, and press <ENTER> to load. Before pressing <ENTER> you can select one of eight screen color combinations by pressing the space bar.

The first thing you will see is a grid across the top and left side of the screen. These numbers are the row and column locations in the top left, or "HOME" position. There are 255 possible rows and 63 possible columns, with the screen framing a small section. Each "CELL" is referred to by its row/column location, such as: R1C1, R10C22, etc. In R1C1, The Home position, there is a solid rectangle, as large as the width of the cell. This is your cursor, or "CELL POINTER", which is where any entry will appear. Below the cell grid is the COMMAND LINE, which shows the primary commands you will use. There are a number of sub-commands related to each of these, but you must type the first letter of the primary command first, or place the cursor over the command and hit <ENTER>.Below the command line is the MESSAGE LINE, which prompts you for further information when needed. In the bottom left corner is the current cell pointer location, and to the right of that is the contents of the current cell. Lastly, in the bottom right corner is the available memory space remaining.

There is a file in the TI Forum Data Library that assists in learning the commands. It is called MPKEYS.HLP and it is recomended that you get a copy before continuing.

Upon selecting a command, a command menu appears with a number of response fields shown. In each response field is a proposed response, which is the default selection unless you change it. To use a command, type it's key letter and fill in the response fields. To move through the fields, use the tab key until the cursor is highlighting the correct area. Type in your response, and either tab to the next field or hit <ENTER> to activate the command.

When the necessary response is a row/column cell reference, there are two ways to respond: Absolute and Relative. Absolute referencing is a numerical definition of the cell location, such as R5C10 (the intersection of ROW 5 and COLUMN 10). A group of cells, a RANGE, is called by giving the boundary intersections separated by a colon(:), such as: R2C1:R4C10 defines a 3-by-10 cell grid consisting of columns 1 through 10 on rows 2 through 4.

Relative referencing involves identifying the desired cell by displacement from another cell, usually the one the cell pointer is currently on. As an example, if you are on row 5, column 10, (R5C10) and you wish to refer to a cell two rows up and three columns over, (R3C13) you could type in R-2C+3 or use the cursor keys to move the cursor over R3C13. The relative address will automatically update as you move. When the cursor is in place, hit <ENTER> (or tab to the next field) and the reference is defined.

So far, we have covered what you see on the screen and in response to the various commands; what the commands and key functions are; and how to fill in response fields where needed. Before going on to building a worksheet, you'll need to know how to save what you're working on, and how to load it back in. Besides the fact that you'll want to take a break occasionally, it's nice to be able to experiment with the commands, "messing up" the worksheet, and then loading the "clean" version back in to continue.

The LOAD and SAVE commands are under the command TRANSFER. Hit "T" and the menu will be displayed. Since the first option is LOAD, hitting <ENTER> now will prompt for a filename. To select SAVE, (or another option) hit the first letter and <ENTER>, or tab through to the desired item and hit <ENTER>. When loading or saving, you'll be prompted for a filename the first time, which will become the default response.

Multiplan also incorporates an extensive helpfile contained on disk. When the command line is displayed, you select HELP with either the Help action key (<AID> or "?") or by typing "H". The worksheet will be replaced by the beginning of the helpfile. If a command has been selected, hitting the help key will display the specific section of the helpfile that pertains to the command you are using. The help menu allows you to RESUME (return to command menu), START at the top of the helpfile, or move to NEXT or PREVIOUS page of information. The first step to creating a worksheet is to decide how many rows and columns you'll need, and how the data will be displayed. It is best to sketch this out on paper to get a feel for how it will look. Also, you'll need to decide what formulas will have to be created that use the data contained in the worksheet. Lastly, you will probably want to change the format of many of the cells, usually by rows or columns. Most often, the formatting required is for display purposes. Cell width, alignment of the data within the cells, etc.

Now that you know how everything will look, begin by formatting the cells. Upon start-up, the cells are set with a number of defaults. You may want to change the widths of some columns, to between 3 and 32 columns, to show all of the entry for the cells. If the data in a cell is too large to fit the width of the cell, it will be truncated to fit, unless it is a numerical entry, where it will be replaced by a sting of "#"'s.

FORMAT CELLS is used to set cell alignment and display format. A cell can be aligned to either center text for columnar headers, etc., or to align data displayed in tables. For instance, a table of dollar values could be shown with a "$" in front and decimal points aligned. The display formats are used to show how the data appears in a cell. CONTinuous allows the text in a cell to run over the right boundary to the next cell. If all cells are made continuous, you have a word processor-type format. EXP displays numbers in scientific notation. Fixed Point rounds off decimals to a defined number. GENeral is as you see when starting up, values displayed as entered. INTeger rounds off all numbers to integers. "$" (Dollar) adds a dollar sign to numbers and rounds to two decimals. "*" Replaces the number with an equivalent number of asterisks, to use like a bar graph. "%" displays the number in percent form. Lastly, the "-" just leaves the setting at the previous option.

Now that the cell formats are defined, it's time to start entering data. Begin by labeling your rows and columns, as necessary. To enter data, either text or values, move the cursor to the desired cell and hit either "A" or "V", depending on the type. The command line will disappear and you'll be prompted for either text or value. Type in your entry and hit enter either <ENTER> to return to the command line, or use the appropriate FCTN-ARROW key to move to the next cell. With the FCTN key, when you land on the next cell, you are prompted only for text/value entry. In this case, you do not hit A or V to declare type, but when you begin entering data, Multiplan decides what style the data is, and responds accordingly. The only disadvantage is that there's a slight delay between the first character of your entry and the remainder, so if you type in, for instance, the word "TOTALS" too quickly, all you'll see in the cell is "TTOLS". After a bit of use, a "stutter" habit is developed in how you enter data, so this becomes less apparent. When entering data, if an error is made, do not use the FCTN-S key to backspace for correction (as programmers are used to), the backspace key is CTRL-H (as telecommunication folks are used to).

If, after creating part of a worksheet, you need to add or delete rows or columns, three commands apply. DELETE completely removes any number of rows or columns. BLANK just removes the data in the cells, the row/columns remain and retain their formats. INSERT creates a new row or column set to the default settings.

Formulas are used to perform a mathematical computation upon the data in a cell or group of cells. One example is in a sales order form, where you have a column of data that is totaled at the bottom, multiplied by a tax percentage, and the tax added to the result. The cell in which the sub-total is to appear would contain a formula describing a sum of the data in the columns, expressed as either a chain addition problem, (R3C5+R4C5+...+R10C5) or using the SUM() function and a range of cells. (SUM(R3C5:R10C5)). The formulas can become quite complex, depending on the work performed. Appendix C contains a list of the mathematical functions that can be used in building formulas. Formulas can also consist of names of cells as the opperand, as in "SUBTOTAL x .079", to calculate the entry for a cell named TAX. Names are assigned with the name command. Names can be any continuous string of alpha- numeric characters, but must begin with a letter. Simply place the cursor over the cell to name and press N. Type in the desired name to the response field, and TAB to the next field. The current cell will be shown as the proposed response. If a range of cells is desired, hit the FCTN key, at the cell response, to move the cursor from the current location to the end point, then hit <ENTER>. In this manner, a whole row or column can be named. Names can also be used in the GOTO command to aid in moving quickly to a location. "GOTO TOTALS" for example.

Windows allow you to view more than one area of your worksheet at one time. You can split a row or column of titles to form a window over the data, so as the cursor is moved throughout the worksheet, the headers remain in place to see what data is shown. Also, separate worksheets can be developed in one and divided into windows so all can be seen ar once. After selecting the window command, four options are shown. SPLIT is what opens the windows, either horizontally, vertically, or at preset titles. LINKing two or more windows scrolls them together as you move through the worksheet. BORDER is used to put a border of any character surrounding the windows, to make them easier to read. A window is cancelled with the CLOSE option.

Once you have finally created the worksheet, and all the data has been entered, what do you do with it? In a sense, the end product is the worksheet, because you may refer to it constantly as new data is applied, and a printed copy might become outdated quickly. After all, that's part of the reason you are working on an Electronic Spreadsheet in the first place, the instant and easy update of information.

In some cases though, a printout is desired, either in the form of a disk file that can be incorporated into a document on a word processor, or a hard-copy printout for reference. The printer command has four options used in printing the worksheet. FILE prints the worksheet to disk in display variable 80 format, which can be loaded into a word processor. Before printing a hard copy, you must first set margins and print options. The MARGINS option sets the limits of rows and columns in the printout, along with indentations and paginations. OPTIONS defines the portion of the worksheet to be printed, using a range of cells. The set-up field contains te device name of your printer. The last two fields let you print the formulas "hidden" in cells, and whether or not to print the row/column numbers. After margins and options are defined, select the PRINTER option to begin the print-out. If the width of the worksheet exceeds the width of your printer carriage, the left half will be printed entirely, then the right half below that, so the two can be cut--pasted together.

In some cases, you may be working on a number of worksheets that are related to each other, such as in a business with SALES/PAYROLL/INVENTORY spreadsheets. These separate files can be linked together so data can be drawn from, as an example, the INVENTORY file to be used in the SALES worksheet and information from SALES could be used in PAYROLL.

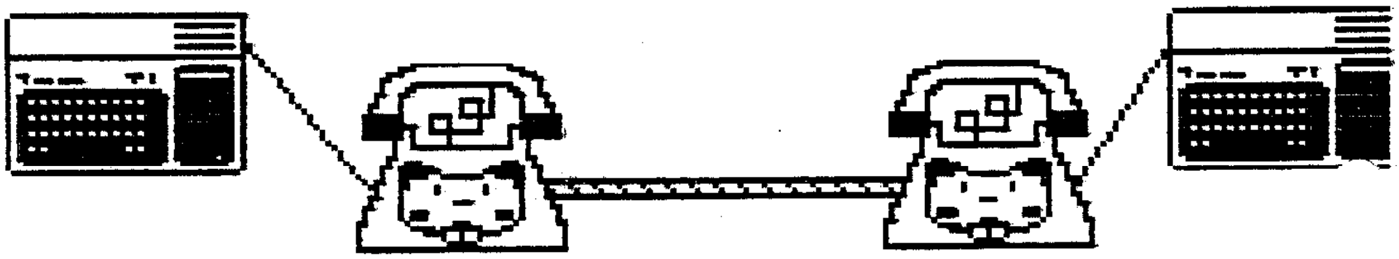
The eXTERNAL command, (press "X" at command line) is used to COPY data from an inactive sheet into the active one. You are prompted for the filename of the source sheet, the name (or R/C reference) of the source cell, the destination cell of the data, and the LINK option. If LINK is selected, then the two sheets will become linked so that when the destination sheet is loaded, the source sheet will automatically be used to supply data where needed. The LIST option displays the names of all sheets supporting the active sheet. The USE option allows you to switch which inactve sheets will support the active sheet, so long as they are in the same format. As an example, the SALES sheet would call upon different INVENTORY sheets for each month, all created in the same format, with different data.

Multiplan is one of the most powerful tools to be used on any computer. It's versatility allows it to be used in many different applications. Word Processing, record keeping, budget/accounting, etc. Any application that requires storing data in a tabular format. The instant update of information and the advanced mathematics capability can be used in a variety of ways.

Versatility is the main attraction of the many spreadsheet programs used on various machines, and in fact, Multiplan can even use files stored in VISICALC(tm) format. VISICALC, one of the "first" major spreadsheets, is similar to Multiplan in many ways: the screen display; cursor positioning; error correction; and entering data and formulas. The referencing of cells is more detailed with Multiplan, including the ability to name cells for ease of use. It has been shown that Mulitplan can be easier to pick up and use for the person not familiar to speadsheets, although once the concepts are mastered, the usage is similar in all. With a familiar knowledge of a program like Multiplan, you could do away with a word processor, a database manager, or even a pocket calculator, although each has it's specific advantages.

We have tried to cover the basics of getting started in working with spreadsheets, but this has still only scratched the surface of the wealth of information within the manual supplied with Multiplan. A walk-thru in the first half provides a very good introduction, and the second half documents each command and function in detail. There are also a number of good books available on Multiplan, and the software is the same on nearly every machine.

(d/l from CompuServe by mel gary)



LOCAL BBS

BILL REISS 679-0549 24 HRS

BILL WRIGHT 257-2607 24 HRS

BEAVER BRD. 238-8170 9PM-9A
TECHIE 300-1200

C TUTORIAL-2

By Warren Agee 70277,2063

Second in a series of tutorials on the C language

In my first tutorial, I covered storage classes, something necessary to know before you even start programming in C. Functions are another basic concept which must be grapsed before writing C programs. Simply put, a function is a subroutine designed to perform a specified task. In some cases, values are passed to and from functions, while other functions require no communication. Numerous functions are part of the standard C library, like gets() and puts(), which allows input and output of strings, respecitively. Others, like fopen(), are kept in function libraries and stored on disk. And, of course, you may write your own functions. Indeed, the process of writing a C program involves writing user-defined functions, then putting all these functions together into a runnable program.

So, where do we begin? First of all, naming conventions. Although a function may have a name of any length, the c99 compiler only recognizes the first six characters, and they may be only alphabetic. Unlike most other compilers, the underscore (_) is not allowed. Secondly, what distinguishes a function name from a variable is the presence of parentheses. Depending on the purpose of the function, the parentheses may be empty, like getchar(). If the function requires values to be passed to it, these are placed inside the parentheses, as in puts("\nHello there!") . Now that the cosmetics are out of the way, let's get down to creating a function.

As I mentioned in the last tutorial, to call a function, merely type in its name, followed by a semicolon. To alert the compiler that you are creating a function, omit the semicolon.

```
clr()
{
  int c;
  putchar(12);
}
```

Here we define function called clr(). Note the missing semicolon. Also note that since the parentheses are empty, we are not going to communicate any values to the function. Next we have an opening brace, which signals the beginning of the function body. Note that the brace aligns with the first letter in the function name above; this is a standard C convention to make programs easier to read. Then we indent a few spaces, another convention. We then define the integer variable "c." Because this statement occurs inside the function body, it is local to that function (See Tutorial #1 for more info). The next statement is a standard console i/o function which prints a character to the screen whose ASCII value is in parentheses. In this case, putchar(12) simply clears the screen. We then find a closing brace which ends the function. Notice that the two braces line up.

Suppose that we want to pass one or more values to a function. Look at this:

```
add(n1,n2)
int n1,n2;
{
  int sum;
  sum=n1+n2;
  return(sum);
}
```

The first line tells the compiler to expect 2 values in the parenthesis when this function is called. We give these two values the names n1 and n2. When one calls this function, two numbers may appear in parentheses [like add(1,2)] or two variables [like add(a,b)]. The next line is a variable declaration, which was described in the first tutorial, but the purpose here is a little different. The function add() receives two values; now the compiler has to know what KIND (class) of values they are. Since we are passing numbers, we declare them as integers. also notice that this must come *before* the opening brace. We then declare another variable, sum, to hold the sum of the two integers. We perform the addition just as one would do in BASIC. The next line is very important.

When this function is called, we give it two numbers, and we want back the sum, right? Since the variable "sum" is local to add(), once we return to the calling program, the value of sum is lost. "Sum" only exists in add() and nowhere else. What we have to do is artificially send the value of "sum" back to the calling program, and we do this with the return statement, as shown above. Now, when we call add(), we will get back the value of sum, like this:

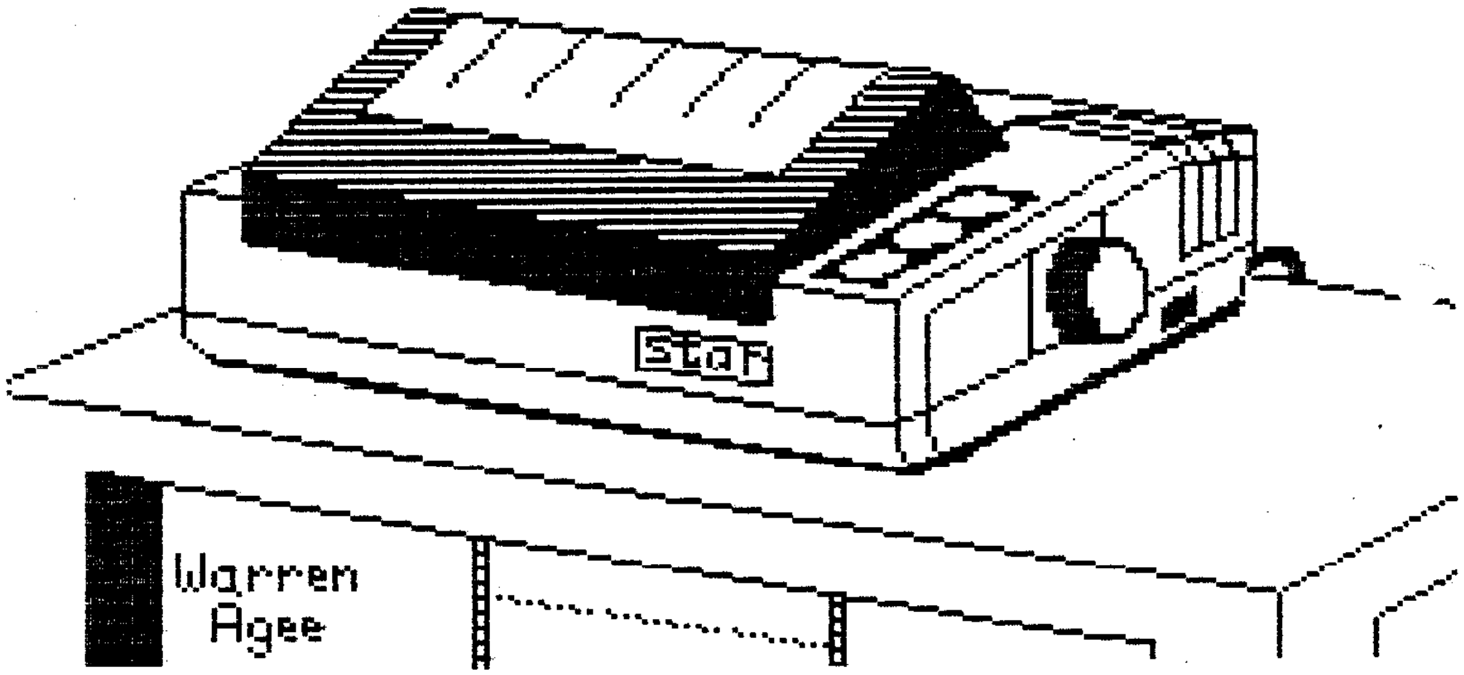
```
main()
{
  int c;
  c=add(5,2);
}
```

The expression "add(5,2)" is replaced by the answer, and we assign that value to c. If we just wrote "add(5,2)" and did not assign it to anything, the sum would just be discarded.

But why do all this? We could just declare "sum" as an external variable in main(). That way "sum" would retain it's value throughout the entire program. In very large programs, you can run into difficulties if you use only external variables. Stick to local (automatic) variables whenever possible.

Well, there you have it! There is a lot more to cover as far as functions go. The return statement only returns ONE value, no more. If you need more than one value back from the function, you have to use pointers. Pointers can be quite sticky and confusing to beginners, so I will be spending quite some time on them in the next few tutorials. So stay tuned, and experiment! It's the only way to learn! (Well, reading my tutorials may help a bit!)

(d/l from CompuServe by mel gary)



FOR SALE

EPSON MX-80 PRINTER-PARALLEL

GOOD CONDITION

[illegible]00 DOLLARS

CONTACT MEL GARY AT 828-5407

SOLD

ANNOUNCING THE TI-99/4A TRAVelER!

An exciting new magazine-on-disk, which will include over 700 sectors of programs and articles in each issue, featuring authors like Mack McCormick, Jonathan Zittrain, Tom Weithofer, Tom Kennedy, Todd Kaplan, and Barry Traver.

The first issue (September 1985) -- which has already appeared -- includes an article on TI ARTIST and GRAPHX by Ron Albright, assembly language articles by Mack McCormick on accessing the RS232, an article "TI&ME: Saving Tips on CompuServe" by CIS Wizop Jonathan Zittrain, two utilities by Tom Freeman (one to print DV80 files _sideways_ on your dot matrix printer, and the other to print two-column newsletter-style sheets with full justification), a versatile assembly language utility accessible from Extended BASIC called "RAW" (for "Read And Write") for single-sector disk access by Barry Traver, another utility by Barry to show DV80 files on your screen in 40-column text mode, a colorful game called HoleyMoley by John Behnke, plus much, much more.

The "diskazine" is actually priced less than some "freeware": you get a six-issue subscription for only $30 (that's over 4000 sectors, so that you are only paying less than 3/4 of a penny per sector!). This special price is only guaranteed through the end of 1985, after which time subscription rates may increase, so _now_ is the time to send in your check!

The TI-99/4A TRAVelER is edited by Barry Traver, whose programs have appeared in various publications ("Giant and Dwarfs" in 99'er Home Computer Magazine, "Merge/Read" in Craig Miller's Smart Programmer, and "Numb/Conv" in Super 99 Monthly, where he is now a regular contributing staff writer). He was recently maed a full sysop on the TI FORUM on CompuServe, where he earlier received the honor of being asked to serve as Chairman of the Expert Member Board.

The "diskazine" is being "published" on commercially-made SS/SD "flippies," so that the same format will work on everyone's disk system. Because of the medium employed (disk rather than cassette), it will be assumed that most subscribers will also have at least a 32K RAM card, Extended bASIC, and probably Editor/Assembler as well, and the contents of TRAVelER will be chosen accordingly. (In other words, you can expect something a bit more sophisticated than the TI BASIC programs which are generally all that is available in magazines and books in bookstores!)

TRAVelER SUBSCRIPTION FORM

Name____

Address______________________________________

City_____________________ State________ Zip Code__________

Please send your check to the following address: Barry Traver, Editor, GENIAL TRAVelER, 835 Green Valley Drive, Philadelphia, PA 19128. Thanks!