DALLAS TI HOME COMPUTER GROUP
OCTOBER, 1985

Meetings: 7:00pm, 3rd Friday each month
At Northlake College, Room B-241, Irving, Texas

NEW ROOM NUMBER!
SEE STORY, BELOW.

>>> NEXT MEETING: Friday, 18 October, 1985 <<<

President: Richard Roberts (SOURCE: TI3552; STARTEXT: 8762)
Vice-President: Dan Johnson
Secretary: Louis Guion (STARTEXT: 77536)
Treasurer: Earl Bullock
Editor: Robert Lee Hoffpauer

This newsletter is the official publication of the DALLAS TI HOME COMPUTER GROUP, a non-profit organization serving member/users of the Texas Instruments 99/4A HOME COMPUTER. For more information you are invited to attend our next meeting or send a SASE to: DALLAS TI HOME COMPUTER GROUP, [redacted], Irving, Texas 75061.

= =

PRESIDENT'S SCREENDUMP: Richard Roberts

It had to happen some time, right? Good things just don't last forever! I am referring, of course, to the sudden uprooting of our meeting room at North Lake College. It seems that the good scholars have found something else they would like to use the facility for on Friday evenings, and so we have been relocated to another room. For the balance of the year, or until further notice, we will be meeting in room B-241.

By comparison, it is going to seem like we will be meeting in a closet, but I believe the room will be adequate for our regular attendance. It will certainly enhance the definition of the word "TOGETHERNESS"! The seating is what is called "table-top" chairs, so you might plan to bring as little as possible, since you won't have as much room to spread out. The room is located in the same hall as B-206, but the number plate is not visible from the hallway. As you walk down the hall, look for our sign-up table, which will be prominently positioned.

The program for October will consist of the third installment of Robert Hoffpauer's "Programming in BASIC" lesson, and will expand on the topics covered during the first two sessions. The second part of the program is just now jelling, and will not take it's final form until the night of the meeting. It's rather broad in concept, but I'm working on paring it down to keep it as simple as possible.

If you had to give it a name, it would be something like "Care and Maintenance of TI Hardware". Ideas already suggested include the dis-assembly of an expansion box, with tips on cleaning the bottom contacts, and how to replace the airplane propeller with something quieter. Also planned is coverage on the dis-assembly of a console, with tips on replacing the GROM connector, the keyboard, and checking for loose chips. And then, there is the ever popular "solid state" cartridge. I know some people who were surprised they could be opened, much less need cleaning. This, and much, much more is in the works. It should be a very informative program, regardless of your computer expertise.

During the Executive Board meeting this past month, it was recommended that we find someone willing to take the responsibility for setting up, and maintaining a Cassette Software Library. Since our entire library is on disk, it would be most helpful if the volunteer has a disk system, but not mandatory. If you are interested in this position, please give me a call, or catch me at the meeting.

I'm not for sure if everyone understood the details concerning Navarone's agreement to sell their software to User Groups at dealer cost, so let me mention it again. Basically, the only requirement is for the group to solicit a list of programs desired by club members, collect the total amount to cover the order, and send it in directly. There does not have to be a "minimum" number of any particular program on the order. It may be for as little as one program, or a dozen.

The FOR loop is a counted loop, in that it will iterate a specified number of times. There are also indefinite loops that iterate until a specific condition is met, and infinite loops that iterate forever. Since nobody sets out to create an infinite loop, let's cover indefinite loops. There are two types of indefinite loops -- top-tested and bottom tested. A top-tested loop will evaluate the 'condition' set up by the programmer before any of the loop-statements are executed; a botom-tested loop will perform the loop-statements before checking the condition. A top-tested loop may never execute any of the loop-statements if the condition for entry into the loop is never met; a bottom-tested loop will always execute the loop-statements at least once, and, if the conditions are met, will repeat them one or more times. Diagrams 1 & 2 illustrate the two looping constructs.

In designing an indefinite loop, whether top or bottom-tested, it is often the practice to set a flag immediately before or just after entry into the loop. The flag might be set initially to zero, then later in the loop-statements it can be incremented if some action has been taken, and then used as the test-condition for exiting or continuing the loop. A flag is simply a variable set up by the programmer for use as an indicator of what has happened. As an example (example #3) of a programmer designing his own indefinite loop, and using a flag as the condition for exit, I will re-write the BUBBLESORT subroutine from the August meeting. In this example I have an indefinite outer loop containing a counted inner loop. FLAG is set to zero at entry into the outer loop. If during one pass thru the data, an exchange is made, then FLAG is incremented. After the counted inner loop is complete, FLAG is checked to see if there were any exchanges during the pass. If there wereany exchanges made, the outer loop repeats by re-setting FLAG to zero. If there were not, the list must be in sorted order, so we can leave the loop in line 1110, jumping to the first statement following the loop. Happy trails!

EXAMPLE #1:

LOOP

```
140 FOR COUNT=1 TO 8
150 READ A(COUNT)
160 NEXT COUNT
```

EXAMPLE #2:

OUTER LOOP

INNER LOOP

```
1010 FOR PASS=1 TO SIZE-1
1020 FOR I=1 TO SIZE-PASS
1030 IF A(I) > A(I+1) THEN 1040 ELSE 1070
1040 TEMP=A(I)
1050 A(I)=A(I+1)
1060 A(I+1)=TEMP
1070 NEXT I
1080 NEXT PASS
```

DIAGRAM #1:

TOP-TESTED (WHILE-DO)

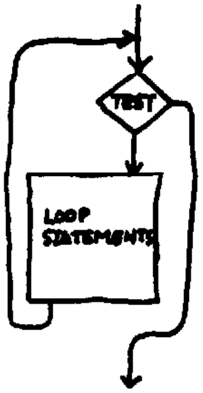
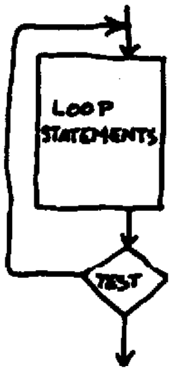


DIAGRAM #2:



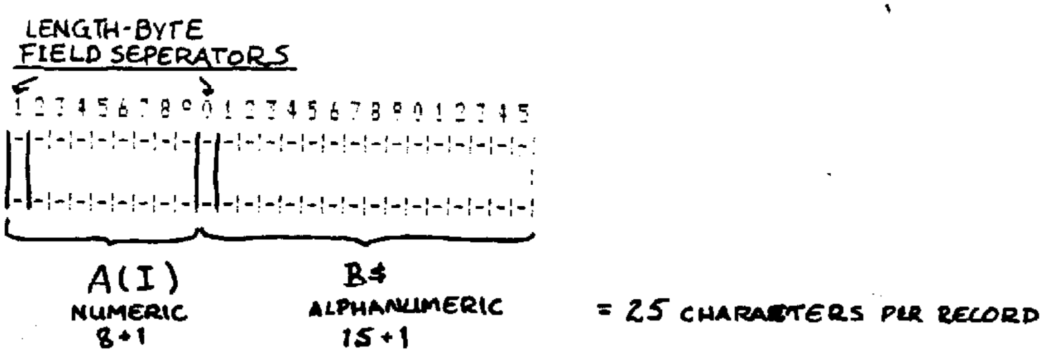
BOTTOM-TESTED (DO-UNTIL)

EXAMPLE #3:

OUTER LOOP

INNER LOOP

```
1010 PASS=0
1020 FLAG=0                      ← SET/RESET FLAG
1030 PASS=PASS+1
1040 FOR I=1 TO SIZE-PASS
1050 IF A(I) > A(I+1) THEN 1060 ELSE 1100 ← COMPARE ADJACENT PAIRS OF DATA
1060 TEMP = A(I)
1070 A(I) = A(I+1)
1090 A(I+1) = TEMP                 EXCHANGE & INCREMENT FLAG
1090 FLAG = FLAG + 1
1100 NEXT I
1110 IF FLAG = 0 THEN 1130 ← OUTER LOOP TEST
1120 GOTO 1020
1130 RETURN
```



Now that you have a 'picture' of the *record structure*, you are ready to specify the *file characteristics*. The OPEN statement must be the first file-related statement performed by the computer. This will set up a linkage in the computer between your program and the file. In the OPEN statement you can specify a number of attributes. A file must be assigned a number, because it will be referred to by number (OPEN #7:). You will also specify, by name, which file you are referring to (OPEN #7:"DSK1.EXAMPLE"). Now, whenever your program refers to file #7, the computer will know you mean DSK1.EXAMPLE.

You may now specify the *file organization* (SEQUENTIAL or RELATIVE), the *file type* (INTERNAL or DISPLAY), the *open-mode* (INPUT, OUTPUT, UPDATE, or APPEND), and the *record type* (FIXED or VARIABLE).

```
OPEN #7:"DSK1.EXAMPLE",SEQUENTIAL,INTERNAL,OUTPUT,FIXED 25
```

File organization refers to whether the records will be accessed sequentially, one after another, or randomly (record 15, then 37, then etc.) Cassette recorders can only read and write records sequentially, so there would be no need to specify SEQUENTIAL -- if not specified the computer would 'default' to sequential organization. Printers can only be written to in a sequential fashion, so a printer file would also be SEQUENTIAL. Disk drives can access records randomly, so organization could be either SEQUENTIAL or RELATIVE. However, you should learn sequential access methods, then move on to random methods.

File type actually refers to the way the data is stored, in INTERNAL or DISPLAY format. For the time being, use INTERNAL for cassette and disk, display for a printer.

Open mode specifies whether you will write to a file (OUTPUT) or read from a file (INPUT). These are the only two modes for use with cassette recorders. Disk drives can also be accessed in UPDATE mode which will read or write, as well as APPEND mode which will write additional records onto the end of a previously created file.

Record type actually refers to the length of the records. Cassette records can only be written/read in FIXED lengths of 64, 128 or 192 characters. If the data doesn't add up to one of these values, the computer will 'pad' each record with blanks, so don't worry if your total characters add up to less than 64, 128 or 192. Disk records can be FIXED length or VARIABLE length. Use fixed length records to begin with.

Once your program has OPENed a file, you can read data from the file with the INPUT statement (if the file was OPENed in INPUT or UPDATE mode). The INPUT statement includes a variable-list. The variables and their order correspond to the fields of the record. For instance, if the first field is a numeric data item, and the second field in a character string, the first variable in the variable-list will be a numeric variable, and the second varialble will be a alphanumeric variable (ends with a $).

```
INPUT #7:A(I),B$
```

Likewise, the PRINT statement is used to write to a file. It also has a variable-list that corresponds to the fields in the record.

```
PRINT #7:A(I),B$
```

When you are through using a file, you must close it.

```
CLOSE #7
```

This is an offer you can't refuse! The savings are anywhere from 30-40% compared to the list price. If you are interested, please see me after the meeting. I do have a dealer price list you can peruse.

Congratulations to Bob Viering for winning the raffle last month. Despite his love of the GRAPHX program, he naturally chose the PAINT 'N' PRINT program over the other items offered. We still have not given away the Multiplan program yet, so it will be offered again this month. Remember, the raffle box will be at the sign-up table, so be sure to put your entry, or entries, into the box before the meeting begins. Each entry consists of a $1.00 donation, placed into an envelope with your name written on the outside. Below is the complete list of items offered, from which the winner may choose one.

- MULTIPLAN spreadsheet program (requires full system)

- Return to Pirate's Island (can be used with console only)

- Scott Adams ADVENTURE module (cassette version)

The next Executive Board meeting will be October 26. For those of you on the Board, please make a note of it on your calendar, and plan to be there. Many things are discussed, and you miss out on the all-important decision making process when you're not there. Time is 1:00 pm, at Louis Guion's house.

I hope to see everyone at the meeting on October 18. As always, we will try to get started at 7:00 pm sharp. BE THERE!

Getting Down to Basics: Robert Lee Hoffpauer

Here are the references you should read, and have handy, as you learn to use files and records:

User's Reference Guide (TI) -- beginning on page II-115, covers the OPEN, and CLOSE statements, the INPUT and PRINT statements, the End-Of-File function, and the RESTORE statement; this is the primary reference for both disk and cassette users.

Disk Memory System Manual & RS232 Interface Card Manual (TI) -- provides additional information for disk systems and for RS232/PIO (printer,modem) applications. These manuals came with your expansion cards.

On the 99/4A, every device is a file. The printer is a file that can only be written to, a cassette recorder is a file that can be both written to and read from, a disk drive is a file that has 'sub-'files that can be written to and read from with greater flexibility than with cassettes. The singular fact is that the computer looks at all of these devices in the same way. All it needs is some help from YOU -- the programmer.

A file is simply a collection of records. In the case of the printer each line of print is a record. A printer is an output only file. With cassette recorders and disk drives you can write or read records into/outfrom memory. Each record can be composed of one or more fields, each containing a piece of data. YOU -- the programmer -- have to determine in advance what the *file characteristics* and what the *record structure* will be. YOU will tell the computer what the *file characteristics* are when you describe the file in the OPEN statement, you will tell the computer what the *record structure* is when you PRINT a record to, or INPUT a record from the file.

First, draw a picture of the *record structure*, identify each field by length (number of characters plus one for character strings, eight plus one for numeric values) and assign a variable name to each field. Count up the total number of characters to find out how long your record will be. Disk records cannot exceed 256 characters, cassettes cannot exceed 192 characters.