T.I.-dings

from NewJUG/North

July & August 1989
Volume 7 # 7 & 8

P.O. Box 84
Dumont, NJ 07628

Your T.I.
'Monthly' Newsletter

Officers:

President: Walt Macieski...868-6593	Trez: Frank Filica...384-8797
Veep: Jerry Stockler...516-735-9517	Sec: James Ott......790-6052
Editor: Henry Hein.......607-988-7789	

Next Meeting: Aug. 15th at Bergenfield P.L. at 7 P.M.

"We are family enjoying the unspeakable peace and freedom of being orphans."
(Paraphrase of G.B. Shaw line in *Major Barbara*.)

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

BINGHAMTON, NY 139
PM
AUG 8
1989

July

Dallas TI Computer UG*
PO Box 29863
Dallas, TX 75229

User Groups: Please Reciprocate!

T.I.-dings
From NewJUG/North
PO Box 84
Dumont, NJ 07628
July and August 1989 Vol. 7, # 7 8

Editor: Henry Hein
Address: RD #1, Box 343 A
Otego, NY 13825
Phone: 607-988-7789

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

NEXT MEETING AUGUST 15th

at Bergenfield Public Library 7 PM

Correspondence Welcome!

Warning:

In order to cram as much news in these pages as I can I had to reduce the print contents of articles I found to be of interest to members. Our budget is low, and I feel the economy is necessary. Although the bulk of the cost goes to postage, we also have received requests from UGs we have not yet exchanged with. They have shared with us some original materials which may be included in these pages from time to time. Magnifying utensils may be necessary for some of us with feeble vision, like myself, but I do what I must to disseminate as much as possible. Please bear with me! Henry!

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

June's Meeting Minutes

By James Ott

The June meeting started on time with 15 members and 3 guests. Future meetings for the summer months was first mentioned by Frank Filice. He has booked us for July 18th and August 15th. The discussion then led to Funnelweb. It was acclaimed as a great program when utilized with a RAMdisk.

Herman Follick discussed his experience with reinking the ribbon on his Star NX1000. He says he mixed stamp pad ink with alcohol to ink the ribbon. The result was a bit messy at first, but cleared up after printing a few sheets.

(Ed. Note! don't mix the alc with ink! Add a few drops later and let ribbon sit a few days. Do not overink it either! The alcohol helps disperse the ink. Too much ink saturation of the ribbon takes a long time and causes the printer to draw unwanted lines above and below text. The ribbon cannot absorb too much ink, so be careful in not adding too much. A little goes a long way! If an eyedropper is used, just one dose should be enough as the ribbon winds with a hand drill tool using a screwdriver bit, or with the same tool and the ink roller running on the exposed ribbon surface itself. OK Herman, go to it! ED.)

A console was donated to the club and I will be bringing the computer to the club for demos.

Our tresury reported a balance of $234.58.

Following the meeting, I gave a dmeo of FORTRAN 99 using a simple program that displays a message on the screen 100 times to show the speed over BASIC. It was about 2 or three times faster. If I make more progress, I'll give another demo of Fortran soon.

John Bonito used TI-Artist and answered many questions on using the program.

July's Meeting

By James Ott

The meeting started a little late with an attendance of 12 members. John Dyer showed us his technique of reinking ribbons with his homemade device. The device has a place for the ribbon and a motor to turn the ribbon as it is reinked.

Henry Hein attended the meeting and gave a talk on his recent trip to Canada. He also brought in some new programs for the TI. One of them is supposedly a faster version of the TI-Writer - Formatter. Henry hasn't tried it yet but gave them to Bill Stpedeli to try. Perhaps Bill might give us a demo next month. Henry acknowledged that he got it from someone in the Ottawa UG and raved about the activity in that great group of TI enthusiasts.

Our treasurer reported no change in the the club's balance from last month.

There wasn't a demo but plenty of equipement was brought in. Walt Macieski brought in his Winchester drive to be looked at. Jerry Stockler and Steve Marino, Jr. made a quick repair and the drive was put in action.

Bruce Mosher had some trouble with a RAMdisk and the problem wasn't cleared up when he shipped it back to the manufacturer and was returned with the maker's assurances. (Keep trying, Bruce! Maybe the software is at fault, try a copy of Frank's! Ed.)

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

Ed Comment:

It's too bad that the summer meetings are limited in time to one hour and forty-five minutes. We should strive to get there on time. Frank Filice assured me that the members are active and really devoted to their beasties giving them much love and care. I was pleased to hear it and hope this attitude continues. Sorry to have missed John Bonito, Jim Lambert, and the others not present. I hope to make an effort to visit you again soon. Trying to find lodgings may be difficult since some old neighborhood friends will be away on their vacations. Hundred+ dollars at motel rooms don't appeal to me. I like to be with friends and enjoy their company, and treat 'em to dinner instead!

Still waiting for guests from the club up here in Leatherstocking country!

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

RAMblings

By Henry

It was a great tour! Unguided! My wife and I travelled with as few plans and reservations as possible. We visited friends, friends of friends, and relatives of friends. Hardly a better way to go! Getting to meet and know people, their feelings, and way of life was our main goal. And we succeeded. Sorry to have monopolized the time at the last meeting about this trip. There were some contacts made with those very creative Canadians of the Ottawa TI UG.

This group is headed by a very energetic lady, Jane Laflamme. She is the sole Canadian distributor of MYARC equipment. She uses the Geneve in her office AND home. She shares an office with her husband's business, which is also prospering with the help of the TI 'stepup'. She has contacts with almost all the UG's in Canada, and runs a BBS, too! How does she find the time?! She lives up to her name, too. A ball of fire! Lots of fun to talk to, besides. Though their businesses are young they are optimistic and energetic, both husband and wife.

Bill Sponchia, who works in the government, comparable to an auditor with our own GAO (Government Accounting Office), has a full range of MYARC equipment, demonstrated (as Jane did), the merits of his equipment as opposed to the IBM. He also demonstrated the speed of data management with some of his own home made programs I passed on to the club at the last meeting. He also described to me the 'fat' in TIW's

FORMATTER. He said the latter had so much immaterial stuff in it which had no useful purpose and gave me a copy of it to pass on. Haven't tried it yet, but maybe someone in the club could tell me how it, the shorter version, goes!

While in Ottawa both gave us a exciting tour of this beautiful city. Bill entertained us the first night, an evening adventure along the canals in the late dusk and moonlight, and Jane the next day, a different part of town filled with parks and waterways. She took us to her office to demo her Geneve. I never saw anything work so fast! Let 'em keep their APPLEs, IBMs, etc.! This Geneve, along with the proper programs easily available to TI users, can do wonders, and at last look I can say that Lou Phillips really made a contribution to TI users who take advantage of his product.

In Toronto I looked up TI user, Wes Shunk. He too got the Geneve and some of the latest database programs. His huge inventory of antiques, mailing lists, clients, accounts receivable, etc., flashed across the screen with a speed IBMs and MACs would envy. Calling up files and records were instantaneous! He uses several of the popular databases made for TI. I forgot which one he likes best, but I think its the recent TI-BASE. Hard Drive, MYARC RAMcard, Quad disk controller, etc., he had it all, and puts it all to WORK! What an array!

I had a long list of names to contact in other Canadian cities I stopped in or passed through. Winnepeg, was in our path but our train was three hours late to bother to get off so late in the evening without reservations. We could have stopped in Regina, Saskatchewan, but the train pulled in about 4 AM, not worth making a hotel stop to catch the next train 24 hours later. We did stop at Calgary for 4 days to see some friends, but had neither time or opportunity even call someone in the TI community there. This was, to me, the most fascinating place in all of Canada. The site of Canada's adventurous pioneering past. Here is where energy, both human and nature's, got together. Both had a hand in building this sprawling prairie metropolis of 3/4 million. Just about every building gleamed like gems in the sunlight, and the dawn illuminating the purple mountains with white peaks to the west about 70 miles away.

The Vancouver area also had its natural landmarks, and there, too, we couldn't look up members of two or three UG's in the area, namely Victoria, Nanaimo, and Vancouver UGs. We had so much else to do.

Our way back routed us to Edmonton, Alberta, Saskatoon, and Winnepeg (again), and Toronto. Again, train schedules and connections forbade us from dallying. We had been informed of some bad news from home and cut off some six days of our 30-day excursion. Nothing serious, but a matter of urgency which had to be handled with our presence. I regret not seeing some other TIers but I was relieved my wife got an education on the value of computers, at home, offices, etc., and no longer complains of being a computer 'widow.' Besides the geography, geology, history, of Canada, she got her education on computers as well.

She knows now this trip was an unexpected bonus.

During the trip she sported a T-Shirt (frequently washed, of course) with the inscription. I'm spending my kids' inheritance!" OK Ma! can I have a Geneve now?? And spend a little, too?

Members of the train crews wanted and asked to be adopted! Ha!

I'm in the process of writing a more detailed version of the journey for publication, possibly, but too lengthy for this NL and its intent. Anyone wishing a copy of it send a formatted disk SASE. Lots of fun, hints, observations and critiques, and a bit of a travelog in itself.

More on the trip *Re* computers. In almost all the major Canadian cities one can find free literature on computers and computer products. These publications are similar to stateside 'computer shoppers.' Free for the taking! They are geared, however, to IBMs and its clones, MACs, Amigas, etc. Canadians are into computers, and how! And TI has benefitted by TIers like Randy Boone, Clint Pulley, and many, many others.

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

NEW RELEASES

Believe it or not!
by Henry

ASGARD announced PAGE PRO99, a high speed assembled program for the expanded 99/4a or the GENEVE. It features paste up capability of 28 pics/page, one large and one small font per page, two linestyles to be used at once, importing/exporting TIW files, printing in single, double, and quad densities, two column texts, multidirectional typing, and a converter for TI-Artist fonts and instances (images). Included in the package - a collection of starter fonts and pics, and a tutorial "straight out of the box." Compared to The Printer's Apprentice (written in FORTH) it should run much faster and perhaps use less disk space for your files.

Though it works with a single drive system, I recommend a multiple DS drive system to avoid swapping and storing for editing recall.

Write to:
Asgard Software
P.O. Box 10306
Rockville, MD 20850
or call:703-255-3085

For this price, compared to the much more expensive TPA which came with very meager instructions one can outdo programs such as Spinnaker's *NEWSROOM* made for IBM and APPLE sold for about the same price!

At last month's meeting I passed a copy of Art Green's (Ottawa UG) TIW v.4.0. He is selling v.4.3 NOW for $10. It supposedly is faster in wordwrapping and disk handling of files. It includes utilities for varied interfacing with Geneves, RAMdisks, and Supercarts, etc.

Maybe worth it! You got his 4.0 from me, try it. If you like it, get his upgrade and give him his dues! Write to
Art Green
1032 Chantenay Drive
Gloucester, Ont.
Canada, KIC 2K9

According to Bill Gaskell in the LA Topics, this program allows you to install the program to your system, configure your own defaults for printer name, screen colors, tab settings, word wrap (on or off), line display (on or off) and defined character set. A few extra ctrl commands to go to top and bottom of texts instantaneously, sending control codes without upsetting line count (with no need of TLs), conditional page breaks, and chain file operations when using multiple drives. Also, as mentioned in *RAMblings*, above, the author took a lot of FAT out of the FORMATTER.

By now INSCEBOT is marketing an assembled TI-SORT. A high speed program for sorting the data from TI-BASE, PR-BASE, and maybe others. For $14.95 plus $2.50 S/H (kinda high S/H!) Well, wotever! Writ by the Faherty duo! Write to Inscebot Inc.
Box 291610
Port Orange, FL 32029

Still in the works! TI-Artist III! Ask about it from INSCEBOT. It may outdo PAGE PRO99 mentioned above.
ÇàÇ┌íÉ0ƒ┌|ª╫┌┌┌┌┌┌┌┌Ça

ONE GALLON INTO A ONE QUART BOTTLE
Manipulating files bigger than RAM
PART 1. Understanding Records

By Art Byers, CN 99'ers.

The advent of new data bases, such as FIRST BASE and TI-BASE, that promise scads of storage space is only possible because of some tried and true techniques of being able to sort and search disk files that are much too large to fit into RAM.

Although these articles refer to XB programming, the techniques can be and are being applied to other 99/4A languages. This is really a generic tutorial. You may want to refresh your memory on just how the 99/4A DOS handles file records on disk.

Helpful study references are:

TI Basic User's reference Guide, section II, pages 118 thru 136.

TI Extended Basic Owner's Reference Manual, pages 61,74,82,104,113;138, 144,150,156.

Disk Memory System Manual, pages 30-36.

Think of storage of information on a disk rather like storage of folders in a file cabinet. Each folder can hold so many sheets of paper, and each file drawer or cabinet a finite number of folders. So our disks can hold a finite number of pieces of information, so many to a sector, so many to a disk. This information and what we can do with it is really the heart of soul of one of a computer's most useful functions. It can sort and search. It can store and retrieve information very quickly. It can update files, etc.

It does not matter if the information retrieved ends up on the screen as a chart, drawing, a digitalized reproduction of a photograph, or becomes an invoice going out to the printer through the parallel port. The time and effort saved, in that we do not have to go to a file cabinet, open the drawer, search for a file folder, remove a sheet of paper that has the information we are seeking, copy it or modify it, put it back into the folder, put the folder back into the drawer etc., is really what computing is about. The 99/4A is pretty good at handling files and records. Let's take a look.

When the TI-99/4A first came out, it was, in many ways, superior to most of its contemporary home computers. Among these superiorities was its many flexible and varied methods of handling disk files. Even many years later, a few much newer well known computers have yet to match it. Let me give you just one example out of the many possible: My friends with Apples and IBM PC's, who program in Basic, must use error traps to find the end of a file. The 99/4A has the EOF(n) function instead.(a)

Among the best features of a good computers' file access is the ability to go directly to a disk file and read a single record of our choice, modify it if we want and rewrite it back to the same place on disk. Additionally, a record can be divided into segments or fields. Each part of a record can then be accessed and read and, if we choose, changed and rewritten.

If we want to obtain specific record numbers at will, we must set up our files as random access or RELATIVE files. If we have only one field in the record, it can be read with either INPUT or LINPUT. However, if we have divided the record into several fields, it must be read with INPUT. There is a small problem with INPUT and DISPLAY VARIABLE files in that it recognizes the comma as a field divider and will only read a string until it comes to a comma. Therefore, text records having commas are best read with LINPUT.

To avoid system errors, the INPUT statement must match the record format. If you have stored 4 numbers in four fields of a record, INPUTing 5 fields will cause a disrupting error. For example, if we have OPENed file #1, record 6, and PRINTed four fields with four numeric values - 10,15,20,25: trying to access the information with INPUT #1, REC 6: A,B,C,D,E would have one field too many and cause an error.

You can mix numeric and string data fields -that is format fields any way you want. ie PRINT #1, REC 8:A$,B,C, D$,E$,F. However, you must then INPUT or read them off the disk in the exact same form they were written. You cannot read a string record as a numeric field.

When we OPEN a disk file, we must specify the length such as DISPLAY, VARIABLE 80 or INTERNAL, FIXED 127. The TI manuals tell us that the most efficient method of file storage for the 99/4A is in binary or INTERNAL form. In order to use our available disk storage space to its maximum efficency, we need to know what size or length to set our records. If a record is only 10 bytes, we can fit 25 records to a sector. If it is 125 bytes, we will get only 2 records to a sector. How do we count Bytes?

Take a look at the PRINT list. Assume we are writing two fields - one numeric and one string of 18 characters. PRINT #12,REC 22:N,N$

When written to an INTERNAL format file, each numeric item will use 9 bytes. The first byte gives the length and is always followed by 8 bytes for the number. A String also has a length byte followed by one byte for each char, so the 18 char field would take 19 bytes. We also need one byte for each of our field separators, the comma, totaling that up means a minimum of 29 bytes for that record. I think there is also one byte of other disk overhead used so we would end up needing 30 bytes for each record. If we have made it a fixed length record and some strings are less than 18 characters, the empty spaces are padded out with binary zeros. If the strings are longer than 18 bytes, they will be truncated. We can get eight 30 byte records to a sector. It follows that because a SSSD disk has 358 useable sectors, the format described above would allow 2864 SUCH records on the disk.

The 99/4A disk system starts numbering records with zero as the first record number. Record #1 is actually the second record of the file. I have found it useful to store pertinent file information in record zero and begin data storage with record #1. When we begin reading data, we either RESTORE the pointer to record one, or specify the record number in the INPUT statement. In record zero I usually put the title of the file, number of records, date of the last update, a password if required by the software, etc. This is a carry-over from my early days of using Cassette for data file storage. The OS for cassette does not create or recognize end of file markers and it was customary to either load all the needed info as to number of records etc into the very first record or place a dummy marker file as the last file record and have the program look for the dummy file and take appropriate action when it was found.(b)

When programming for cassette, the first record would be read and the number of records obtained so that the balance of the records could be read with a do loop. (FOR/NEXT in Basic). The title and last update printed on the screen helped us to know we had the correct file, etc. I see no reason to abandon this just because data is now on disk.

Since access to cassette was so slow, when we first got our 99's, we tended to make our data files small enough so that everything could be read into RAM and searched, sorted, changed, in RAM, as that was very fast. Then everything was rewritten out to tape, even the records we might never have touched.

Although disk access speeded up data retrieval, its most important time saving factor was the ability, mentioned above, not to have to read a whole file into memory, but rather to go like an arrow to a target bull's eye and pick out one record or even a segment of a record.

However, there is another side to this - in short order we soon had more information, more records in storage on disk than would fit in RAM. This is not unique to our computer. Even main frames have massive amounts of data stored on tape or disks that they cannot fit into RAM. For home computers, the need to speed up data access has resulted in RAM-disks and Hard disks a/w/a floppies. That has just made some problems worse. There is no way to fit 20 meg of hard disk info into a PC's RAM. If we can't fit all this into RAM, how can we work with data without innumerable time consuming accesses to disk, no matter how fast the access may be. How can we sort a file too big to fit into ram? How can we search huge files without it taking hours and hours to go through hundreds or thousands of records??

Before we go on to the techniques of solving these seemingly impossible problems, you must fully understand how the 99/4A stores data in record form because it is the individual records we will be reading and manipulating as we search, sort, change, and rewrite back out to disk.

The next part will explain how we can sort and search large disk-based files.

Footnotes:

(a) There is an evident contradiction in the TI manuals listed above. TI BASIC warns on page II-129 that EOF() cannot be used with Relative files. This caveat is not mentioned in either the XB or Disk System manuals, and in fact the other two say specifically "The EOF() function always assumes that the next record is going to be read sequentially even if you are using a RELATIVE file." This implies that EOF() will work with RELATIVE files in XB, and in fact. that is true.

(b) A side comment: Cassette files have one safety feature lacking in disk files. If you have an electric power lapse, or computer failure while saving a disk file and it is not properly closed, you will have real difficulty retrieving your data and will need considerable expertise to do so. With a cassette file, all you will lose is the single record that was being processed when the failure took place. You will be able to read and retrieve all the other data on the tape with a few lines of simple Basic programming. This is true only for Data File storage. A program being stored on tape will be busted by a failure during the storage process.

$$$$$$$$$$$$$$$$$$$$EoF$Part$$$$$$$

ONE GALLON INTO A ONE QUART BOTTLE
Manipulating files bigger than RAM
PART 2: Sorting and Searching

By Art Byers, CW 99'ers.

This article is going to be a simple presentation of what can be some very complicated ideas. If you want a detailed text, try your best to find a book called "Data and File Management for the TI-99/4A by J P Grillo, J D Robertson and H M Zbyszynski, published by Wm C Brown Publishing of Dubuque, IA in 1984. It is out of print, but copies do show up at TI Faires, Public Libraries, and TI Clubs' collections. The whole book is very worth while, but chapter 6 specifically covers disk searches and sorts.

Those of us who started with just a 99/4A console and a cassette player did all file manipulations in RAM. We sorted and searched at a pretty fast speed. Some of this was because our data bases were small (oh say a 50 name list) but most was due to the fact that RAM sorts and searches are quick.

With the advent of Disk based records, we soon began to build files that were too big to go into RAM to be Sorted. How then can we manage?

Why not forget about RAM sorts and just use any good sort that compares two disk records and rewrites them in proper order right on the disk. Only a thousand bytes of memory would be needed for that?? Frankly, it could take HOURS and HOURS to sort a large data base on-disk because each disk access is more than a hundred times slower than a RAM access, not to mention that the wear and tear on your drives would be tremendous as many thousands of disk accesses would be needed.

METHOD ONE:

The answer lies in the DOS ability to seek out a specific record and then a program's ability to use just a small part (or field) of that record. Let us call that small part the "key" part. We then load just the keys into memory along with their original (disk file) record numbers and do our sort. After sorting, each key has two record numbers tied to it: the original record number and the new record number that will be used to copy the file to a new disk in its sorted form.

Here is a short example: We'll read records 1 through 5, (remember from part one, we use record zero for other purposes, so our first record is #1), picking off the key and the record number where it was found. For simplicity we will use a single letter as a key but the key would normally be from 4 to 7 digits or letters.

Z-1 C-2 X-3 F-4 M-5

We sort on the keys and assign the new set of record numbers:

C-2-1 F-4-2 M-5-3 X-3-4 Z-1-5

Now a simple program can go to record #2 on DSK1 and read the whole record (not just the key, C) and then write it out to record #1 on DSK2. The program will then go to record #4 on DSK1 and read the whole record (not just the key, F) and write it out to record #2 on DSK2, get record #5 on DSK1 and write is as record #3 on DSK2, etc. This method is also a safety feature as we keep the original disk intact until after we have verified that our new disk has the file, sorted, all in good condition. There are some other excellent methods that will rewrite to the same disk, but you sure had better have a backup copy just in case!!

Why do we need this system? Assume we have 14,000 bytes of free Ram but our disk file has 1436 records and uses about 90k. There is NO WAY we can fit that into Ram for a sort. BUT, using the example you have just seen, We use only a small part, the first five letters of a last name or a Zip Code or the last four digits of a phone number and using that as our "key" load it into memory along with it's record number.

Obviously, the first five letters of a last name multiplied by 1436 records is 7180 byts, a little over half of our available memory. Some additional memory is needed for the original record number and the new record number, stored in string form, but we still fit easily into our available RAM.

Next we put them into Alpha order. We can use some pretty good RAM sorts such as Quick Sort or Shell Metzner. (This newsletter has published several routines for sorting on any segment of a string and they should be in your libraries.)

Now there are pitfalls. Because we are only sorting on five letters, the name Brownwell can end up ahead of Brown or Brownley. Battaglione might be ahead of Battaglia. We might need a "second pass", ie: If we so chose, we can then run a secondary sort of only the "B's" using ten letters and properly reorder those records.

There can be no pretense that this will be as fast as a RAM sort even if your storage is on RAM-disk. Also true that this type of record manipulation on Floppy disk will be much slower than if done on a Hard Disk or RAM Disk. However, there is not too much choice with the limited RAM of the 99/4A, and the majority of 99/4A owners do NOT have hard disks or even RAM disks.

A BETTER METHOD:

There is still another method related to the above. That will speed up the process even more: Instead of our rewriting the entire disk of records, as re-ordered, to DSK2. We simply write a file to the same disk, DSK1, listing the sequence of records as it needs to be accessed to find the keys in proper alpha order. This leaves DSK1 intact and only requires the writing of a new file, which we will identify as our "sequence file". This is much faster than making a second disk. The increase in speed applies to Hard disk as well as Ram disk.

Using the same example as method one, above, we can omit the keys and just write the numbers. We will reverse the position of the digits and list the order first and the record number (on DSK1) second.

ie: 1-2 2-4 3-5 4-3 5-1. Compare this to the last example and be sure you understand what is happening.

You must leave room on your file disk for this sequence file. Remember that for our imaginary file of 1436 records, we will have two numbers for each record. String form will be used to take up less disk space, but a good rule of thumb is to always leave about 10% unused on any disk of files. You never know when you will need some extra room.

Now when we want to search the disk, we simply load this special file into a memory array, search it with a combination of RAM search and disk search as will be explained below.

There are some much more complicated methods. We can use Binary sorts or Tree sorts that leave the files in place on disk but use "pointers" to locate alphabetical or numeric order. They require leaving unused buffer or over-flow space on the disk and need much much more sophisticated programming. Read the book mentioned above if you want further insights into these advanced techniques.

Why sort at all??

The main reason is that it makes searches much easier and faster.

HOW DO WE SEARCH??

One logical way would be to start at record #1 and proceed towards the last record, and at each record stop to compare the field we are searching with the string or number we want to match, but that would take much too long when we have large files, especially if it turns out our "match" was near the last record of the file.

We start our search in the MIDDLE!! Assume we have 26 records. #1 holds A, #2 holds B, #3 holds C, etc. to record #26 which holds Z. Let's search for "Q". Starting in the MIDDLE we first compare the unknown with record 13 and find M.

We now ask, by reading the record off disk, is the data in record 13 equal to, greater than or less than Q. If it turned out that the data was equal to Q the search would be over. In this case the answer is "less than".

Next we go to the halfway mark, between record 13 and record 26, record 19. There we find S and ask the same questions. The answer is "greater than" so we look at the halfway between 19 and 13 which will be 16. We discover Q is not there either. So, now we have only two places to look, location 17 and 18. The next place our search algorithm will take us is to record #17 and there is our match in only four reads of the disk at the least, five at the most.

If we had started at record 1 and simply searched them all it would have taken 17 reads. Do you see the principle involved?

Let's take our ficticious 1436 record file mentioned above. Assume we have sorted on a five letter key and written the special sequence file. We would start at the middle number and access the attached record number (on DSK1). Number 718, the middle, might lead us to record #45. If our next look was at number 354, that could take us to record #1400. Then we look at number 177 which might lead us to record #976. Etc and etc until we find our match. The maximum number of disk accesses, to find any key (or to establish that the match is NOT in the file), would be 11. Now that's pretty snazzy for a file with 1436 records!! Without having first sorted the file records, however, this method could not work.

Please understand there are many ways to skin the proverbial cat. You may have a different way of locating records. If your favorite way of searching is better, come to the next club meeting and share it with us by explaining how you do it!!

$$$$$$$$$$$$$$$$EoF$$$$$$$$$$$$$$$$$$

ÇAÇ┌iéb⁄-│ª╫┌┌┌┌┌┌┌┌┌ÇÂ

Great tutorial, Art! Thanx! Henry.

FREEWARE REVIEW: 99-CALC

99-CALC is a spreadsheet written for the 4A. What makes this program valuable is that it will run on a bare bones system (no memory expansion and no disk drive). It requires only a cassette player and Extended Basic. A printer, disk drive, and memory expansion are optional.

Given the small size of the requirements, 99-CALC compares well with full featured programs. You can do arithmetical functions (add, divide, subtract , multiply and per-cent) and you can total and average columns and rows.

You have full screen editing, can move from one cell to an adjacent one or jump to any active cell. The spreadsheet can be printed on a 80 column printer. You can opt to print formulas.

Phil Barnes, who wrote 99-CALC has truly come up with an ingenious program. He uses every bit of memory possible. For example, you can hide numbers under column and row titles. The program is filled with nice touches. 99-CALC comes with on disk documentation (that prints 7 pages) and a sample file.

RECOMMENDATION: If you do not have memory expansion and have need of a spread sheet program, 99-CALC would be a valuable addition to your library.

For those of you with 32K and a disk drive, Multiplan has more features. 99-CALC is worth getting if for no other reason than to see what can be done with the 4A's 16K of VDP RAM. 99-CALC is also much easier to master.

HOW TO ORDER: Send a disk and postage paid, self addressed return mailer to:

Phil Barnes
24631 Via San Fernando
Mission Viejo, CA 92692