TRSTimes

Volume 6. No. 1. - Jan/Feb 1993 - $4.00

TRSTimes welcomes our new subscribers

LITTLE ORPHAN EIGHTY

1993! Can you believe it? It is hard for me to understand that time is moving that rapidly. It seems like I celebrated my 20th birthday just the other day; but now when people see me with my TRS-80 and I hear them whisper 'obsolete', I have that sinking feeling that they are talking about me!

While much has changed in the nation and in the world, our TRS-80's remain the same. They work, which is more than I can say for some of my fancy PC's. A couple of weeks ago, my wife's IBM clone blew up. It was just luck that she had finished her screenplay, so nothing of immediate value was lost. My own 386 is at times temperamental and refuses to allocate my 8 megs of memory correctly, causing Steven (my son) endless frustration when trying to play Ultima or Wing Commander. I probably need to replace the motherboard, but that will have to wait awhile.

My Model 4's, on the other hand, boot-up and get me to the 'LS-DOS Ready' prompt each and every time I turn the machines on. It is so refreshing to sit down to a working unit after having a bout with a PC. Maybe, that's why Tandy dropped the TRS-80 line: they didn't make enough money on repairs!

So, what's happening in Model I/III & 4 land? The big news (at least to me) is that Roy Beck and I finally have managed to get our BBS on-line and WORKING. For awhile we did have the BBS on-line but, unfortunately, nothing seemed to work. Boy, it was some kind of experience, and Roy eloquently tells the tale of our woes in his article 'Setting up the TRSuretrove BBS' on page 5.

We invite all readers to call this BBS and participate in the fun and excitement. Do note that there is absolutely no charge to log on and use this board. Your telephone company will undoubtably bill you long distance rates if you are calling outside the 213 area code, but the BBS is FREE. You may call as many times per day as you wish, stay on as long as you wish, and download as many files as you wish. We have implemented this (ugh! I hate to say the word) LIBERAL policy to make it as easy and pleasurable for TRS-80 owners to make contact with other TRS-80 people, and to allow you to really look around and get the files you need. The only thing we ask is that you participate in whatever discussions that may be going on, and if you know the answer to a question being asked that you post a message telling all how it is done. We would also appreciate uploads, but we absolutely do NOT want copyrighted commercial material. In other words, we would sure like your favorite public-domain programs, but don't send us a copies of Scripsit, LeScript, PFS-file, Little Brother, or any of the other programs that are the property of others.

Also, we do insist on common-sense behavior while on the board. Do not upload any pornographic or other questionable material, and do keep messages in tasteful English. Fair enough?

Now that I have covered the few rules of TRSuretrove etiquette, I will try to explain a feature of the board that seems to be a little confusing to new callers:

The TRSuretrove BBS is actually 3 boards in 1. That is, it has 3 sub-boards. When you first log on, after having answered the name and password prompts, you are automatically placed in **Sub-Board A**. This is called the **Public Board**. This is where most of the general messages are placed. As a rule, there will not be files for downloading here. You might think of this sub-board as the stepping stone to the other sub-boards.

Now, if you wish to access the Model 4 or Model I/III sub-boards, be aware that they are named **B** and **C** respectively. Simply choose the **S** option from the top menu (<S>elect sub-board) and type the desired sub-board letter when prompted.

The B sub-board is used for Model 4 discussion and file transfer. This is where you leave questions/answers for anything pertaining to the Mod 4. Also, an extensive download section is available here (with more to come).

The C sub-board is reserved for Model I/III discussion and file transfer. If you have questions (or an answer) or just general comments about the Mod I/III, this is where to put them. This section also sports a large number of files for you to download.

Keep in mind that the menus for each sub-board look exactly alike, so you must be aware of where you are. Confusing at times? Yes, I know, but we'll all get used to it shortly. Be sure to call:

TRSuretrove BBS

(Los Angeles)

(213) 664-5056

8 N 1

Where the TRS-80 crowd meets

Finally, a big thank you to Roy Beck, Chris Fara, Mike Banks, Ed Gracely, Bruce Tonkin, H. L. King, Mark Colen, Bruce Robbins, Henry Herrdegen, and Michael W. Ecker for their enthusiasm and contributions to this issue of TRSTimes. Also, thank you to all the readers for staying with us in this our 6th year. And now.....

Welcome to TRSTimes 6.1

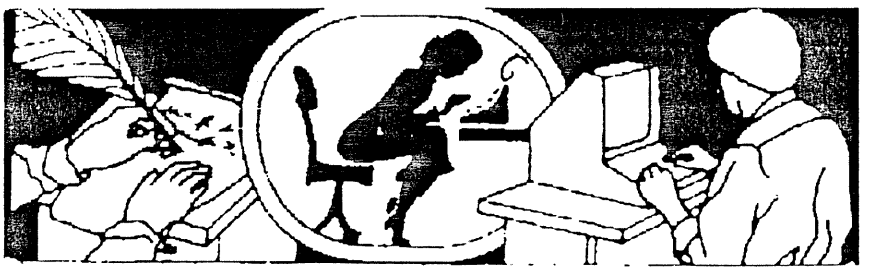
TRSTimes magazine

Volume 6. No. 1 - Jan/Feb 1993

PUBLISHER-EDITOR
Lance Wolstrup
CONTRIBUTING EDITORS
Roy Beck
Dr. Allen Jacobs
Dr. Michael W. Ecker
TECHNICAL ASSISTANCE
San Gabriel Tandy Users Group
Valley TRS-80 Users Group
Valley Hackers' TRS-80 Users Group

TRSTimes magazine is published bi-monthly by TRSTimes Publications. 5721 Topanga Canyon Blvd, Suite 4. Woodland Hills, CA. 91364. (818) 716-7154.
Publication months are January, March, May, July, September and November.

1993 subscription rates (6 issues):
UNITED STATES & CANADA: $20.00 (U.S. currency)
EUROPE, CENTRAL & SOUTH AMERICA: $24.00 for surface mail or $31.00 for air mail.
(U.S. currency only)
ASIA, AUSTRALIA & NEW ZEALAND: $26.00 for surface mail or $34.00 for air mail.
(U.S. currency only)
Article submissions from our readers are welcomed and encouraged. Anything pertaining to the TRS-80 will be evaluated for possible publication. Please send hardcopy and, if at all possible, a disk with the material saved in ASCII format. Any disk format is acceptable, but please note on label which format is used.



THE MAIL ROOM

TRSCAL

Yesterday I entered the code for TRSCAL/BAS, and have an unsolvable (to me) problem with it. When I run it, I put in the month and year prompts requested and the calendar outline appears on the screen. Next, the figure 1-9 are printed in a VERTICAL column down the Sunday row. At the bottom, the error message "Illegal function call in 23" appears and everything comes to a sudden stop. I checked and rechecked, thrice, all the code against the magazine pages 7 and 8. I do that by entering each line in the EDIT mode and tapping out the line with the spacebar, checking each character against the copy. It is a rather long method, but it does pick up the typos.

Since the figures appear in a vertical column instead of a horizontal row, it would appear that there is a semi-colon missing somewhere. I have looked all through the code listing, but can find no place to put it.

Line 23 - PRINT@SW*V+H,A$;:RETURN - has a semi-colon after A$, where I think it should be for a horizontal row, but it doesn't do the job.

Since you wrote the program I'm sure you can tell me if I have goofed someplace, or that perhaps there is an omission or typo in the magazine.

D. Cobaugh
Port Angeles, WA

The program, as listed in the Nov/Dec 1992 issue, is correct, which means that you have an error. Unfortunately, you did not include a listing of your typed code, so I am forced to guess at what is wrong. From your description, however, it appears that the vertical (V) and horizontal (H) cursor position values passed to the print routine in line 23 computes a PRINT@ position that is out of range. This means that the calling routine (in this case - lines 200-220) is passing improper values in variable V and, also possible, in variable H. The only other thing that I can think of (without seeing your code) is that variable D in line 200 has a value of other than 1-7. D is the numeric value of the day of the week (Sunday = 1,Monday = 1, etc) and is computed in the subroutine in line 43. If you still can't find it, send me a printout of the code, and I'll do what I can to track down the error..

Ed.

GRAPHICS-90

I got Larry Payne's GRAPHICS-90 from you when it first became available, but quite honestly haven't taken the time to do much with it. Purely my own fault. Maybe not enough imagination.

I have 13 Ghosts, so I know the tremendous potential the program has in both speed and versatility. I don't know how many copies may have gone out to subscribers, it truly is a masterpiece and a bargain.

Is there a possibility you might print another article or two, or some program or subroutines for GRAPHICS-90 in TRSTimes?

A couple of articles might generate more interest and get a few more copies of the program out. I would certainly be interested. Just a thought.

Richard M. Gilfillan
Wellington, KS

I quite agree that GRAPHICS-90 is a very special program, and to be sure, TRSTimes will be happy to publish articles about it. The challenge has been issued, folks. How about it? Can we get the TRSTimes readers to come up with articles on GRAPHICS-90?

Ed.

MYSTERY

The magazine (issue 5.6) arrived yesterday, and I thought McDowell's MYSTERY program could be probably some Christmas picture (I have a few of those programs), and I typed it in. Correct! But one thing puzzles me, and I have had no time to try to find out, other than triple-checking my typing: me thinks it should keep on "twinkling", but does not. It sets a block on top of the little package on the left, and after about 15 seconds and resetting 28 blocks, it stops for about 5, then resets 2 more inside the tree, and cuts off the tree tip. And that's the end of it? Maybe it should also separate the left-most package from the tree-bottom? I'll slip in a printout of a twinkling tree program I re-wrote for the Model III from a program printed in the Dec. 82 MicroComputer News (remember them?) for the Model II, which was a rewrite by R. P. Graham of a previous one I've never seen.

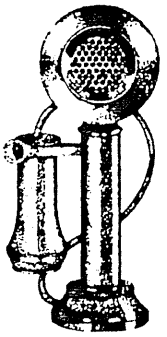
Henry Herrdegen
LaSalle, Ontario

Yes, the left-most package could stand being separated from the bottom of the tree, and a light does seem to have been dropped on the little package on the left. But the program does continue to run, setting and resetting the lights. You must have missed some piece of code somewhere.. If you ever get it running to your satisfaction, let me know, and we'll publish the changes. Though a program, such as MYSTERY/BAS, is useless, it is, nevertheless, fun to tinker with. Thanks for your XMAS program. The listing can be found in the HINTS & TIPS section.

Ed.

SETTING UP THE TRSuretrove BBS

By Roy T. Beck



Introduction

Recently the topic of running a Computer Bulletin Board (BBS) came up for discussion in a restaurant session after a club meeting. The parties involved were Lance Wolstrup and myself. Lance suggested the need for a TRS BBS in the Los Angeles area, as there appeared to be none which were local and oriented to the TRS group. The discussion revolved around hardware and facilities, as the software was known to be available (Mel Patrick's FASTPLUS).

I admitted to having a second phone line in my computer/office space in my home which was available, and of course I had a couple of extra computers and hard drives laying around. (Doesn't every one?) Lance soon convinced me I could set up the necessary hardware and do the physical maintenance, and he could supply suitable software for downloading to users.

I agreed to go along with this proposal (my wife, Barbara, sometimes questions my sanity at times like this...) and proceeded to try to order the FASTPLUS program from Mel Patrick. From here on, my story will be of the trials and tribulations involved in getting "up and running". Actually, I am still not through with this phase, but Lance asked me to set down on paper the events as they occurred. It wasn't as easy as A-B-C, to be sure.

Obtaining FASTPLUS

I had a copy of Mel Patrick's FASTTERM, an excellent shareware terminal program, and so I looked up his phone number in the sign-on screen. I then called Mel via FASTTERM to request a copy of FASTPLUS. The phone number (in British Columbia, Canada, was no longer in use. GRR. I then attempted to locate him via directory service, but no soap. I actually had an old copy of FASTPLUS on hand, but I wanted the latest version to start up the board. In leafing through the manual for the old program, I found an auxiliary program by Karl Mohr listed and described, complete with a phone number. I then called Karl who was still at the listed number. He in turn graciously gave me Mel's new number for his BBS, and I was back on track.

I called Mel's new BBS number, signed on, and left a message requesting information, costs, etc for a current copy of FASTPLUS. I eagerly called back the next day for an answer to my message. None... I then called the next day, and the next, etc, still none. I also left a few chiding messages about SYSOPS not answering their mail. About two weeks later, I found an answering message, chiding me in turn for complaining because he had been on vacation. How was I to know? Anyway, we were finally communicating. I then ordered a copy of the program ($30). However, he had warned me in his message that he was thinking about taking some more time off! To complicate this, I too, was planning a week off, although on business in my case. Anyway, a week or two later, I finally had the package in my sweaty little hands.

Configuring FASTPLUS

Next step was to find time to study the 80 page manual, figure out how to configure my setup, and finally find physical space to locate the beast. Since the phone line was only available in my office, the BBS reasonably had to be there, although I could jury rig an extension to some other room. However, Barbara hinted it had better stay in the office; too much of my junk has already overflowed to other locations in the house. Hence, the BBS had to be in the office. There is only one bench available for the purpose, and it already is occupied by an EPSON MX-80, a model 4P, and a 486-50. Where to put another 4P and hard drive? Well, I decided I would initially run the BBS on the existing 4P. and on floppies at that so I would not have to give up my existing hard drive. I had a 1200 baud modem which worked fine with FASTTERM, so of course that was going into the BBS setup. The space problem would be solved by "double-decking" my desk. I would attach some shelf brackets to the wall above the desk, lay some boards on the brackets, and thus have a space for a Model 4 and a hard drive, with space for a dedicated printer if I chose.

I then made an initial effort to run the BBS on the floppies on my 4P, which seemed to function OK. But with only two floppies available, there was really no room for uploads. So, I then grabbed the nearest spare hard drive which I had just acquired from a fellow in Iowa, a 5 meg

unit, and plugged that in. It had the usual 4 equal-sized partitions. I sort of randomly (ignorantly) stuffed the various files into the 5 meg drive. We then made the announcement in TRS-Times that we would be "on the air" by Nov 1. Well, we tried!

TELCO Trouble

The first problem was inability of any user to sign on to the BBS. Since I had another 4P available, and another modem, and parttime use of the other phone line, I set up a "user" computer right next to the BBS computer. After a bit of checking, I decided the BBS dedicated phone line was bad. (High noise level). I then called the phone company, and you know the reaction you always get when you suggest there could be something wrong with THEIR equipment! Especially since they wanted someone to be home all day the following day so they could get in to check the inside lines, equipment, etc. I insisted they come out anyway, and check out from our house back to their central office. They agreed to do that, although if the trouble was found to be inside my house, I would be stuck with a bill for about $45 to cover their costs. Since I was confident the trouble was on their side, I told them to have at it. Well, they came out the next afternoon and lo and behold, the trouble WAS on their line! Since it had just rained, I think rainwater got into a cracked cable sheath somewhere and caused the trouble. Anyway, they solved the problem, and I was back in business. I thought.

Hardware Trouble

Now that the phone line worked, and I could call in as a user and get an auto-answer, I expected we would be up and running. Not so. The next problem was losing messages, losing files, and locking up of the computer. After much burning of the midnight oil, (by both Lance and me), we began a wholesale replacement of components. We tried a different computer, a different modem (I had a 2400 baud Hayes available), even tried the older version of the software which I mentioned above. Finally, things degenerated to the point where a user could not even sign on, as the auto-answer function had stopped working. What were we doing wrong?

Since we had swapped everything, I was really frustrated. In a sort of desperation mode, I sent a copy of the software to Lance via FASTTERM and my 1200 baud modem, (which would still dial out, although it wouldn't auto-answer), and he proceeded to set up the BBS on his desk at his house. Glory be, it all worked as advertised! Back to my problem; What was I doing wrong? Good Ol' Lance at this point insisted it had to be my Modems, and he brought over a 1200 baud unit which had worked at his end. (That is a 50 mile round trip he made). Finally, everything began to click at my end. BOTH of my modems were defective in some way! Lance then went to a computer swap meet and bought a new 2400 baud modem for me, and I installed that. It worked!

Swapping Hard Drives

Now we could proceed to load files onto the 5 meg hard drive. But, disaster lurked again. Since the partitioning of the 5 meg drive allowed only 1.25 partitions, this had to be upgraded to something more roomy. I then went to my shop and picked up a complete, working 15 meg unit which was on the shelf, ready for sale to a customer. But the partitioning had to be tailored to the BBS, not just four equal partitions. So I reformatted the hard drive to 3 partitions, a 400K drive :0, and drives :1 and :2, each with about 7.3 meg. This would allow the DOS and the Public Board (board A) to be on drive :0, the Model 4 board (board B) on drive :1, and the Model I/III board (board C) to be on drive :2. I also had floppies :4 and :5.

Lost Bulletins

I read the BBS manual some more, and decided where to put all the files. I then told the BBS where it could expect to find the various bits and pieces. Time to smoke test everything. Yes, the board booted up, and I could sign on. But what happened to the sign-on bulletins for each board? Only the public board bulletin came up. The other two were missing. Back to the manual. I then realized I had misinterpreted a statement there, and instead of the bulletins being located on the drive corresponding to the boards, ALL the bulletins had to be on one drive, my choice. So I moved the bulletins for boards B and C back to drive :0, and now all my bulletins were working.

Lost Messages

Next problem; the message numbers were all messed up. If I left a message, it was numbered with too low a number, and it disappeared, never to return. Oh, woe is me. Back to the manual. Fortunately, Mel had indicated the structure of some of the files, and using PMOD6 by Kim Watt, part of TOOLBELT, I dissected the message file and the SYSINFO file which contained the pointers to the message file. With about a half hour of analysis and about 20 minutes zapping, I corrected all the message number-

ing and voila, the messages reappeared, and in the correct order.

Things are really looking up, now, but there are still a few error flags in the Model 4 catalog. Got to clean those up next.

Lost Files

We began loading files to the board, but soon ran into a new problem. When a user would call and download a file, it promptly disappeared off the board and off the disk as well. "FILE DELETED" was shown. What the heck? At first this was reported to me by a user in a message to the SYSOP, and then I actually saw it happen! Why was a file being deleted after downloading?

David Graves, a user in Arizona had it happen to him and we discussed it a bit via messages. He kindly offered to talk it over with Steve Groce a knowledgeable SYSOP also using FASTPLUS in Arizona. The answer came back a day later. I had set a flag incorrectly when I uploaded each file. The question at Upload was, "Single D/L, Yes/No". I didn't know what it meant, and mistakenly said "Yes". This had the effect of allowing only one download, and then deleting the file from the Download catalog AND FROM THE DISKS!

By this time we had a large number of files on the board, and I didn't want to go back and upload each one again just to answer the question with a "NO", so I looked for an easier way to solve the problem. But, I had a small problem; it was a weekend and I had left the BBS manual on my desk at work, and didn't want to drive there to get it. I reasoned the download catalog had to have a flag for each entry to show the YES/NO answer, and since space is precious, it probably would be bitmapped. In examining the download catalog, which is the file DLn/FPF, where n is the board letter, I found a few bytes which looked to be the bit mapping I was seeking. Sure enough, by repetitively uploading the same small file with various parameters changed each time, I was able to find the flag bit for the Single D/L question, and several others at the same time. Others include Visible/Invisible, and Protected status. These are all in byte 02 of each download entry. (Later, after bringing the manual home, I reviewed the section covering these bit assignments, and found the purpose of the other bits). The list below shows all 8 bits. It is interesting to note that bit 6, the one I needed, was listed in the manual as "unassigned"! The manual is a bit out of date. Even if I had had the manual, it would not have shown me the bit I needed.

Bit	Purpose
7	Unassigned
6	Single D/L, 0 = No, 1 = Yes
5	0 = normal access, 1 = SYSOP access only
4	0 = file OK, 1 = Error in file
3	0 = Use ASCII mode, 1 = Use XMODEM mode
2	Unassigned
1	0 = Visible, 1 = Invisible
0	0 = normal upload, 1 = SYSOP only

Now that I knew which bit of which byte to ZAP, I simply went in with PMOD6 of TOOLBELT and fixed all the erroneous entries in about 20 minutes. That problem is now solved; no more files disappearing! But next we will have to inventory and see which files disappeared and then upload these again! My sincere thanks to both David Graves and Steve Groce for so quickly helping me to solve "The Mystery of the Disappearing Files" as Perry Mason might have entitled it!

More Hardware Trouble

Still more recently, I decided to try to solve the problem of the screen of my 4P becoming rather dim. Since I had another 4P available, (both gate arrays), I simply swapped one for the other. Fortunately, I had backed up the the files the night before. Soon after swapping machines, I discovered the message files on Board A (the Public Board) were farkled. (I love that word!) In the process of trying to recover, I managed to write over my backups, without solving the problem. Final result was I lost all the messages on Board A. After much cursing and blue smoke, I finally associated the beginning of the trouble with the change in machines. (The replacement machine had been trouble free until then). After I swapped back, the message file appeared to function without further garbling. Now I have two problems, both unsolved at the moment. One is the dim screen on the BBS 4P, the other is apparently memory trouble on the second 4P, and that only showed up on BBS operation. I will have to load and run a continuous memory test to try to solve that one, I guess.

I also tried, briefly, using a Laser printer as the logging printer on the BBS. This was unsuccessful, due to the nature of the LaserJet. It is really a glorified Xerox machine in the way it functions, which is that it accumulates data in internal memory until it has a full page; at that point it prints one page and begins a new one. But logging on a BBS is a rather sporadic activity, and it is desirable to be able to look at the printer for recent sign-ons, messages to SYSOP, etc. This is not compatible with a laser printer; only an impact printer displays current activity. I immedi-

ately put the old MX80 back in service, a task for which it is well suited.

Problems yet to be Solved

One problem not yet solved is how the (Y)ours function works. This is supposed to select messages addressed to me. So far, this hasn't worked for me as SYSOP. Perhaps it works for users, but not SYSOP; I don't know.

Another curiousity is that (C)hat mode conversations back and forth between the SYSOP and a user cannot be routed to the printer. According to some comments by Mel, he had to remove this feature due to lack of space. I found the work-around is to screen print regularly, but this either gives an overlapping type of print-out, or if I forget to do it, a print-out with gaps in the text. Screen print is not the best answer, but it appears to be all that is available. Messages to SYSOP go directly to the printer if it is online, and to a disk file if it is not. All user messages go to disk files, and the catalog of available downloads is itself another disk file.

Odds and Ends

Chat mode can be requested by a user only when he is in the top menu. Since the SYSOP's keyboard is always hot, he also can request Chat mode by hitting "C" when the user is at the top menu. Regardless of who requests Chat mode, the SYSOP must initiate it with the F1 key. When they decide to terminate Chat, the SYSOP must do so by hitting the F2 key; the user cannot do so.

The SYSOP can also sign-on to the board at any time by simply hitting the F3 key. The BBS immediately recognizes him and gives him full access. The SYSOP can also sign on from a remote location, and do many things which he would otherwise do from the keyboard. The program will also recognize a "remote SYSOP". This allows some other person to maintain the board while the SYSOP is unavailable. (Out of town on business, for example). The privileges of the remote SYSOP may be less than the full access which the Master SYSOP has.

The SYSOP can upload files by simply inserting a floppy containing the files into a drive and using Copy or Backup to transfer them to the hard drive. The files are made available on the board by the SYSOP executing the Upload command. In this case, the procedure is quick, because the files are already on the BBS drive, and no time is spent sending them over the modem. The BBS software understands the SYSOP has direct access, and locks out the modem during SYSOP maintenance times.

The SYSOP can determine how many rings to allow on the phone before auto-answering the user's calls. Default is two rings. The BBS software allows installation of special initiation codes for the modem. The BBS is written around the Hayes modem, but can accommodate many others by implementing the correct initiation string.

By turning off the FASTPLUS BBS, and executing FASTTERM, the same hardware setup becomes a terminal facility for the SYSOP to use to sign-on to other BBS's. Since FASTTERM can download from the other BBS, the SYSOP can readily obtain files from other BBS for use on his own, and can of course engage in conversations with other BBS'. Naturally, the BBS's users cannot access the BBS when the SYSOP is running some other program. Since the pattern of callers-in soon becomes apparent, the SYSOP can use his machine at times for other purposes without seriously disadvantaging the users. Ideally, of course, the SYSOP should have a dedicated setup for the BBS, with separate computer, hard drives, printer, etc for his own use. I hope to achieve this separation very soon.

ARCing Files

As an addendum to this epistle, several people have wondered why we bother to ARC (compress) files when we have a 15 meg drive with much space available. There are several reasons. First, and foremost, ARCing a group of files insures that all the related pieces of a package stay together, and the user won't be frustrated by trying to run a new program, only to discover a data file, or somesuch is missing. The second reason is that some of the packages are physically quite large, and ARCing them affects a significant reduction in size, which means a corresponding reduction in download time and reduced long distance.

New Feature

Mike Baldwin of San Diego is going to conduct a forum on another sub-board of our BBS. He proposes to hold Q and A sessions via messages, and of course he can receive and send messages. This will probably be Sub Board D. The BBS software can accommodate up to 16 sub-boards, so there is no danger of running out of room for sometime to come. This should provide a nice soapbox for any among us with an urge to offer advice and information. Come give it a try!

Conclusion

The more I work with this package, the more impressed I am with the scope and detail of Mel Patrick's programming efforts. Come participate with me in this extension of your computer's capability. All you need is a terminal program and a good modem. 2400 baud, 8-N-1 is the best, but slower modems, (300 and 1200 baud) also work well.

PROGRAMMING TIDBITS

BASIC SOUNDEX

Some 120 years ago, as the nation was inching up to the 50-million population mark, the Bureau of Census was sinking down under the weight of the Great American Melting Pot. The plethora of personal and geographical name spellings (and mis-spellings) was growing and indexing of census files was getting out of control. In preparation for the 1880 census a new system was developed, based on a phonetic similarity of words. This indexing system, known as SOUNDEX, has been in widespread use ever since, mainly because of its practical simplicity.

Here's how it works. Any word, regardless of its length, is translated into a 4-character code. The first character is always the first letter of the original word. The remaining 3 characters are digits taken from a table of similarly voiced consonants:

0	none
1	B, P, F, V
2	C, S, K, G, J, Q, X, Z
3	D, T
4	L
5	M, N
6	R

Phonetic vowels (a, e, i, o, u, y, w, h) are ignored. For example:

Smith = S530

The first consonant after S is M which translates into 5. The next T = 3. There are no more consonants after that, so the final digit is 0. Double consonants which evaluate into the same digit are treated as single characters. Thus Smith, Smythe, Smitty, Schmidt, Szmyd, Smead, etc, have the same code S530. If there are no consonants after the first character then all digits are 0 as in L000 for Lee, Leah, Leo, Lew, Louie.

Variants of Soundex include special cases (e.g. the Polish digraph RZ which is treated as single R), or translate into a digit also the first letter, if it's a consonant.

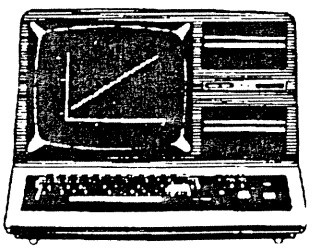
The standard Soundex code can be generated with a simple BASIC routine. Try this:

```
10 defint a-z
11 x$ = "BPFV1/CSKGJQXZ2/DT3/L4/MN5/R6/"

20 line input "Enter word: "; a$
21 if a$ = "" then END
22 b$ = "": if len(a$) < 2 then goto 40
23 k$ = left$(a$, 1): gosub 90

30 FOR x = 2 to len(a$): j$ = k$
31 k$ = mid$(a$, x, 1): gosub 90
32 if k$ < > j$ then b$ = b$ + k$
33 NEXT

40 b$ = left$(a$, 1) + left$(b$ + "000", 3)
41 print "Soundex is: "; b$: goto 20

90 k$ = chr$(asc(k$) + 32*(k$ > = "a" and k$ < = "z"))
91 k = instr(x$, k$)
92 if k = 0 then k$ = "": return
93 k = instr(k, x$, "/") - 1
94 k$ = mid$(x$, k, 1): return
```

The Soundex code is often used in data base 'find' routines and in spelling checkers of word processors. It is more flexible than the 'wildcard' method. With the wildcards the spelled-out portion of a 'search string' still has to match exactly, and characters 'masked out' by the wildcards don't provide any clue for the search. With Soundex a reasonable match can be always located, if it exists at all.

Another application is in the historic and linguistic research. Many words and names that have evolved apart over the centuries, both in spelling and in pronunciation, still maintain their basic phonetics. For example the English 'school', German 'Schule', Polish 'szkola', Italian 'scuola', Latin 'schola', mean the same thing and produce the same Soundex code S400. (It's amusing to reflect that all these words are derived from the ancient Greek root 'sxole' whose original meaning was... 'leisure'. Tell that to the Nintendo generation).

Soundex is fairly efficient primarily with languages that belong to the Indo-European group. As America's population inches up to the 300-million (?) mark and information is getting global, other linguistic families increasingly affect our nomenclature. After a century of good service, Soundex may have to be eventually replaced with some other, more international invention.



Tips on Modem Use

by Michael A. Banks

Modems aren't particularly difficult to set up and use. On the other hand, it's relatively easy to foul up your modem's operation through carelessness or lack of knowledge. This article will serve as a guide in setting up and using your modem. If you observe the procedures and precautions mentioned herein, you'll save yourself a lot of trouble later on.

Marking Connections

By the Book

As with any electronic device, it is important that you connect your modem properly and use it under practical operating conditions. Study the manual that comes with your modem to assure proper connection, and consult with your computer store and the modem manufacturer's customer support department if necessary.

Connectors and Cables

The cables (and the connectors used with those cables) that connect a computer's serial port with a modem are obviously very important elements in the data transfer chain. like serial ports, connectors and cables used with serial ports must conform to the RS-232C standard.

Connectors

There are two types of RS-232 connectors--nine- and 25-pin--and these may be male or female. Nine- and 25-pin connectors are known as DB9 and DB25 connectors, respectively. Each type has numbered pins (very important if you intend to make your own cables--saves a lot of messing around with a continuity tester).

DB connectors can be found on the serial port of your computer and modem, and at either end of the connecting cable. (A connecting cable is typically a "ribbon cable"--a flat cable with multiple connectors.)

Mixing Connectors

DB-25 and DB-9 connectors can be used at opposite ends of a cable if necessary (as when a computer's serial port has a DB-9 connector and its modem has a DB-25 connector). All that's required for the connection to be successful is that the pins on each connector be properly wired (i.e., the wire on each numbered pin on the DB-9 connector should be connected to the correspondingly-numbered pin on the DB-25 connector).

Connector and Cable "Gender."

Incidentally, there's a standard that dictates that the female version of a DB connector should be used only on modems, while the male version should be used on computer serial ports. Thus, a "standard" RS-232C cable will have a male connector on one end (to connect with the modem), and a female connector on the other end (to connect with the computer).

Unfortunately, not all manufacturers follow this standard regarding the gender of their serial ports. So, with some equipment you may find that you have what is called a "gender problem." (No sex-change jokes, please--this is serious stuff!) when this is the case, you'll have to buy or make an appropriate cable with both female or both male connectors. Or, you can obtain what are called "gender changes" to change the "sex" of one end of the cable.

Telephone Plug/Jack Types

Before plugging a telephone line's modular plug into a modem, make sure of the plug's type. Most modems are designed with modular jacks, but modular (also designated "RJ") jacks and plugs come in more than one variety. Some are cross-compatible, and some aren't.

Generally, a home or single-line business telephone system uses RJ-11 plugs, and these present no problem--even if the plugs are set up to provide dial light power to a "Trimline" phone or other lighted dial telephone set. (The only danger in using an RJ-11 plug that provides power for a lighted dial with a modem is if the modem is set up to operate with RJ-12 or RJ-14 plugs. See below for more information on these plugs.)

RJ-41 and RJ-45S plugs are also "safe" to use with most all modems; the exceptions may be modems which have RJ-12 or RJ-14 plugs. See your modem's documentation for details.

If your telephone system is a multiple-line or "key" telephone system, you must have a modem that is capable of interfacing with RJ-12 or RJ-13 plugs (such as a Hayes Smartmodem 2400). The modem you use must also be software-switchable to RJ- 12/RJ-14 operation.

Getting "Set"

DIP Switch Settings

Most modems have user-accessible DIP switches (although the relative accessibility varies from modem to

modem). DIP switches are used to set various attributes of a modem, such as whether it waits for a carrier detect before going online, etc.

Some software packages require that certain modem attributes be set to a specific state. If your software has decent documentation, it will tell you which states must be set, in which case all you have to do is refer to your modem's documentation to find out which DIP switches are used to set the attributes in question. (Some software manuals will even tell you how to set each DIP switch on the more popular modem brands. Too, some modem manuals provide specific instructions on DIP switch settings for certain software packages.)

Rules and Regulations

Telephone Company Regulations

Local telephone company regulations may vary, but in general the following rules are in effect:

- Your telephone company should be notified that you will be connecting an FCC-registered device to your telephone line before you connect it, and that you will be disconnecting the modem when you disconnect it permanently.

- You cannot connect a direct-connect modem to a pay telephone, nor to a party line.

Ventilation and Heat

Don't use a modem as a bookshelf or repository for other materials. While some external modems are designed to serve as a resting-place for a telephone set, they aren't designed to be smothered by papers, disks, etc.

A modem's electronic components generate heat, which must be dissipated; too much heat build-up can interfere with proper operation of the modem. Therefore, heat vents--as well as most of the top of the modem--should be not be covered.

Use the Switch

If your modem is equipped with a power switch, use it to turn the modem off and on. Leaving the switch in the "ON" position and just plugging and unplugging the modem's power supply is not a good idea; this can occasionally create power surges or current overload. When changing the battery in a battery-powered modem, the power switch should be in the "OFF" position, for the same reasons.

Overloading Circuits

Don't plug your modem into an overloaded or faulty circuit. Aside from the fire hazard this creates, overloaded circuits often have low voltage, and low voltage can cause excess heat and poor performance in your modem. (Overloaded circuits are typically those with too many electrical devices plugged into them.)

Surge Protectors

Surge protectors (also called "spike protectors") are an excellent investment. The purpose of a surge protector is to protect an electronic device from surges in a power or telephone line. Such surges are common during thunderstorms and during periods when electrical power consumption is particularly heavy.

Power line surge protectors come in a variety of styles, but all operate in the same manner. Placed in the circuit between your computer and/or modem and the wall outlet, they contain capacitors which absorb and then bleed off excess power.

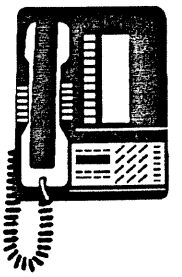
Note that power line surge protectors come several configurations. Some are simply small cylinders or cubes and offer only one receptacle. Others are large rectangular boxes which mount on the wall in place of the wall receptacle's cover. These usually offer more than one receptacle.

Some of the better surge protectors not only provide protection against power surges, but also filter "line noise," and provide a circuit breaker for protection against current overload.

Telephone line surge protectors operate on the same principle as power line surge protectors. Installed between a modem and its telephone line, a telephone line surge protector absorbs then slowly discharges potentially damaging voltage spikes.

Weather Conditions

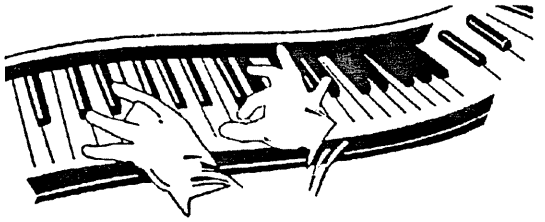
Never use your modem during a severe thunderstorm, nor at any time you observe lightning. Lightning is a guaranteed source of power surges in both AC power lines and telephone lines--and, unless you have a surge protector on both your modem's telephone line and power line, there's an excellent chance that your modem and computer will be "zapped" by a current surge. (Even with surge protectors, there's no guarantee that lightning won't damage your equipment.)



TUNEMEM

Model 4 - Basic

By Ed Gracely



In February, 1986, an article appeared in the old 80-Micro magazine ("Sounds Incredible" by Robert Anderson) which described how to enhance the tone generator qualities of the TRS-80. Many people may never have seen it, to their unknowing loss. I used it to create a number of simple programs, including the one I call "TUNEMEM/BAS", which will serve to also provide the core routine for anyone who wishes to create their own approach to tone-generation.

What this program does is to turn the keys from 1 through 0, across the top of the keyboard, plus the colon key, into a crude scale. As you hit these keys, it plays the tune for you, staccato. I'm not tone-deaf, but I'm no musician either, so my crude scale may be off a bit. I'm not sure that can be helped, because of the limitations of the TRS-80's tone generating capacity, but perhaps a slight change in the tones used would improve it a bit.

The program also records the notes you've hit (up to 500 of them) plus a crude loop-based estimate of the length of pause between them. To exit from entering the tune, you simply type X. It will then play back the tune, upon request.

Anyone interested in the tone-generator itself can fairly easily modify this program to produce tones other than mine. The ITON variable will produce a variety of tones from low tones for large numbers (2000 max) to high tones for small numbers (10 and such).

The tones vary in loudness in somewhat irregular ways, including from machine to machine. For example, on my 4-D, 17 is almost inaudible, while 14 and 13 are fairly loud. On my 4-P, 17 is as loud as the best, while 14 and 13 are rather faint. I cannot hear notes higher than 8 (on the 4-P) or 9 (on the 4-D), but my hearing is not perfect so perhaps others can.

The program claims a bit of high memory in order to have a place to load the tone generating routine. I normally run it on the old TRSDOS 6.2 Basic, although in a brief test it worked fine on LS-DOS 6.3.1 Basic. I have a Screen-editor Basic that I use for everything else, and the program will not run from within that, at least without modifications I don't know how to make.

There may also be some problems if the user tries to add data lines or whatnot. I would advise against it unless you understand such routines better than I do.

The program is fairly self-explanatory. Key it in, run it, and try the results!

TUNEMEM/BAS

```
10 PRINT " TUNEMEM/BAS by Ed J. Gracely. Based on a routine by Robert Anderson
20 PRINT " in 80-Micro, Feb 1986
30 PRINT " "
40 '
50 ' Clearing memory then calling the routine that primes the tone generator
60 '
70 CLEAR,65000!:
GOSUB 660
80 '
90 ' Some initial settings
100 '
110 DEFINT I
120 DIM IATUNE(500),IADUR(500)
130 ISND = &HFF00
140 IDUR = 1    ' Staccato tone duration used throughout
150 ITON = 40   ' Initial tone pitch, soon changed
160 KOUNT = 0   ' Counter to reproduce duration of pauses.
170 '
180 ' Instructions
190 '
200 PRINT "Notes are on the row from 1 to : "
210 PRINT "Begin playing. X ends the tune. Then you can replay it. "
220 PRINT "Up to 500 notes accepted. "
230 KDUR = 1
240 '
250 ' Inputting notes and analyzing them
260 '
270 C$ = INKEY$:
IF C$ = "" THEN KDUR = KDUR + 1:
GOTO 270
280 IF C$ = "1" THEN ITON = 50
290 IF C$ = "2" THEN ITON = 45
300 IF C$ = "3" THEN ITON = 40
310 IF C$ = "4" THEN ITON = 38
```

```
320 IF C$="5" THEN ITON=33
330 IF C$="6" THEN ITON=30
340 IF C$="7" THEN ITON=27
350 IF C$="8" THEN ITON=25
360 IF C$="9" THEN ITON=22
370 IF C$="0" THEN ITON=20
380 IF C$=":" THEN ITON=19
390 IF C$="X" THEN 590
400 '
410 ' Calling the tone routine
420 '
430 CALL ISND(ITON,IDUR)
440 IATUNE(KOUNT+1)=ITON:
IADUR(KOUNT)=KDUR
450 KOUNT=KOUNT+1
460 GOTO 230
470 '
480 ' Replaying the tune
490 '
500 FOR LN=1 TO KOUNT
510 ITON=IATUNE(LN)
520 KDUR=IADUR(LN)
540 CALL ISND(ITON,IDUR)
550 FOR LWAIT= 1 TO KDUR*1.4:
X=LWAIT*2:
NEXT LWAIT
560 NEXT LN
570 '
580 ' Determining user's choices
581 '
590 INPUT "A-gain, N-ew song or E-nd";CH$
600 IF CH$="A" OR CH$="a" THEN 470
610 IF CH$="N" OR CH$="n" THEN 160
620 END
630 '
640 ' The routine that does the tones. Don't ask me how
it works
650 '
660 CK=&H2C82:
ML=&HFF00
670 ML=&HFF00:
FOR T = 1 TO 6
680 READ BY$
690 FOR R=1 TO 16:
BT=VAL("&H"+MID$(BY$,R*2-1,2))
700 POKE ML,BT:
ML=ML+1
710 CK=CK-BT
720 NEXT R,T
730 IF CK <> 0 THEN PRINT "Error":
END
740 RETURN
750 DATA f34e236669225effeb4e236669225cff
760 DATA 11d0072a5eff7cb5c8eb01000003ed52
770 DATA d21dff0b78b1c81100002a5cff7cb5c8
780 DATA eb190b78b1C231FF444D3E00D3902A5E
790 DATA FF2B7DB4C241FF3E01D3902A5EFF2B7D
800 DATA B4C24EFF0B78B1C23AFFFBC901005000
```

Tonkin's Computer Dictionary

Humor by Bruce Tonkin

Advanced (adj.) doesn't work yet, but it's pretty close. See: bug, glitch.

Analyst (n.) one who writes programs and doesn't trust them. A cynic.

Assembler (n.) a minor program of interest only to obsessed programmers.

BASIC (n.) a computer one-word oxymoron.

BBS (n.) a system for connecting computers and exchanging gossip, facts, and uniformed speculation under false names.

Benchmark (n.) a test written ostensibly to compare hardware or software, but actually used by manufacturers to misinterpret or quote out of context in advertisements.

Binary (n.) a two-valued logic especially susceptible to glitches and bugs. It originated as a way of counting on the thumbs, since programming managers usually find fingers far too confusing. See: Hexadecimal, Octal.

Bug (n.) any program feature not yet described to the marketing department.

Bus (n.) a connector you plug money into, something like a slot machine.

Byte (n.) eight bits, or one dollar (in 1950 terms). Presently worth about two-tenths of a cent and falling fast.

C (n.) the language following A and B. The world still awaits D and E. By Z, it may be acceptable for general use.

Chip (n.) a stylized picture of a logic diagram on refined and alloyed sand. See: glitch, bug.

COBOL (n.) an old computer language, designed to be read and not run. Unfortunately, it is often run anyway.

Code (n.) a means of concealing bugs favored by programmers. (v.) the process of concealing bugs by programming.

Cookie (n.) any recondite message displayed by a time-shared system. the message is not often seen, because it only appears when the system is operating properly. Common cookies include the timeless "Murphy was an optimist" and "When in danger or in doubt, run in circles, scream and shout."

Copy Protection (n.) a means of circumventing various rights granted by the Constitution so as to artificially inflate profits.

CPU (n.) acronym for Central Purging Unit. A device which discards or distorts data sent to it, sometimes returning more data and sometimes merely over-heating.

Crash (v.) to terminate a program in the usual fashion, i.e. by locking up the computer of setting a fire at the printer. (n.) the process of such termination.

Data (n.) raw information, esp. that supplied to the central purging unit for transformation and disposal.

Data Base Manager (n.) any fast filing system which gives misleading answers. Also see: menu, bug.

Diagnostic (n.) a test foolishly but often believed to determine the reason for a particular failure. Competent professionals prefer the I Ching or phrenology.

Digital (adj.) of or pertaining to the fingers, esp. to counting on them. See: Binary, Hexadecimal, Octal.

Documentation (n.) a novel sold with software, designed to entertain the operator during episodes of bugs or glitches.

DOS (n.) Acronym. a program which outputs questions given answers, putting users in jeopardy.

Emulate (v.) to simulate hardware glitches with software bugs.

Emulator (n.) a program which emulates. See: Virtual.

Engineer (v.) to build something with bugs (software) or glitches (hardware). (n.) One who engineers.

Format (v.) to erase irrevocably and unintentionally. (n.) The process of such erasure.

Forth (n.) a stack-oriented programming language written right to left and read from bottom to top. It runs efficiently on no common computers and is written effectively by no common programmers.

FORTRAN (n.) an ancient programming language which changed IF's to GOTO's by using a strange three-valued logic on binary computers.
Glitch (n.) an undocumented design feature, esp. of hardware.

GOTO (n.) an efficient and general way of controlling a program, much despised by academics and others whose brains have been ruined by over exposure to Pascal. See: Pascal.

Hard Disk (n.) a rapidly spinning platter divided into sectors. See: Sector, Glitch, Bug.

Hardware (n.) anything prone to physical failure.

Head (n.) the part of a disk drive which detects sectors and decides which of the two possible values to return: 'lose a turn' or 'bankrupt.'

Hexadecimal (adj.) of or referring to base-16 numbers - binary numbers grouped four digits at a time so as to quadruple the opportunity for glitches and bugs. Originated as a means of counting on the fingers of one hand, using the thumb for the 'carry.' Purists who don't like to use the thumb at all prefer 'octal.' See: Octal, Binary.

Icon (n.) a complex, blurry, and easily-misinterpreted pictorial representation of a single unambiguous word. Preferred by illiterates and semiliterates for these reasons, and by others because it slows most computers down so even a cretin with an IQ of 53 may justly feel superior.

Increment (v.) to increase by one, except when segments are used; then, the increase may be by sixteen unless word mode addressing is used in which case the increase is by one or two, depending on the processor and whether the address is on an even boundary or such increase causes an overflow exception processor fault, which may either cause the program to crash or decrease by a large number instead of increase, depending the register used and the operation being attempted.

Iterate (v.) to repeat an action for a potentially and often actually infinite number of times.

Joystick (n.) a device essential for performing business tasks and training exercises esp. favored by pilots, tank commanders, riverboat gamblers, and medieval warlords.

K (n., adj.) a binary thousand, which isn't a decimal thousand or even really a binary thousand (which is eight), but is the binary number closest to a decimal thousand. This has proven so completely confusing that is has become a standard.

Kernal (n.) a misspelling of 'kernel' used by beginning (functionally illiterate) programmers, especially those with some knowledge of C.

Kernel (n.) the core of a program, i.e. the source of all errors. Thus the common misspelling, 'kernal.'

Keyboard (n.) a device used by programmers to write software for a mouse or joystick and by operators for playing games such as 'word processing.'

Kludge (v., adj., or n.) to fix a program in the usual way.

Leading Edge (n., adj.) anything which uses advanced technology. See: Advanced.

License (n.) a covenant which tells the buyer that nothing has been purchased and that no refund, support, advice, or instruction may be anticipated and that no resale is permitted. A modern way of saying "Thanks for all your money and goodbye," far less crude than "Stick 'em up" but even more effective since the purchaser will often borrow the funds requested.

Logic (n.) a system of determining truth or falsity, implication or exclusion, by means of a sort of binary Oneiromancy.

Loop (n., v.) 1. a series of instructions to be iterated. 2. the process of iterating them. Most loops ar unintentional and can be quite droll.

Macro (n.) a series of keystrokes used to simulate a missing but essential command.

Megabyte (n.) more than you can comprehend and less than you'll need. See: UNIX.

Megahertz (n.) a way of measuring how well your computer matches the frequency of your local television channels. Most computers perform exceptionally well on this test, especially the higher-quality foreign-made ones.

Menu (n.) any list of choices, each of which is either unsatisfactory or in some fashion contradictory.

Micro- (prefix) anything both very small and very expensive.

Mode (n.) a way of forcing glitch or bug.

Modem (n., v.) a device used to connect computers (see: BBS) or the process of transmitting data between or among computers, esp. for those unable or unwilling to speak.

Monitor (n.) a sort of television with exceptionally poor picture quality and limited to a single very local station.

Motherboard (n.) the hardware version of the software 'kernel.'

Mouse (n.) an input device used by management to force computer users to keep at least a part of their desks clean.

Nano- (prefix) a thousandth of a thousandth, but not a binary thousandth in either case. Decimal is used for all very small measurements since no further confusion is necessary.

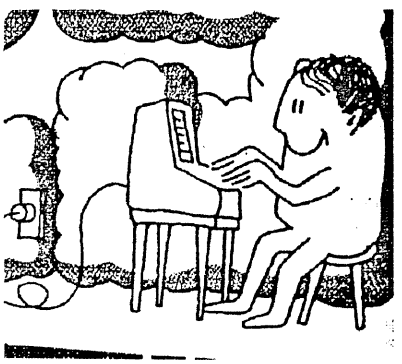
Octal (n.) a base-8 counting system designed so that one hand may count upon the fingers of the other. Thumbs are not used, and the index finger is reserved for the 'carry.'

Offset (n.) a method which permits access to any memory location in thousands of ways, each of which appears to be different, but is not. See: Segment

Operator (n.) 1. One who has no experience with computers. 2. Any beginner, esp. one part of whose salary is paid in soft drinks and processed salted food treated with dangerous and illegal drugs or preservatives. Differs from a programmer in that a programmer will often take the dangerous and illegal drugs or preservatives directly.

Pascal (n.) a classroom project which was released before it could be graded - probably a good idea, considering. One wishes the University had a better system of academic controls.

Patch (v.) to fix a program by changing bytes according to the rules of logic. (n.) Any repair of this form.



Pirate (v., n.) to steal software, or one who is such a thief. True pirates see nothing wrong with thievery, having successfully forgotten or repressed all moral values.

Pop (v.) to remove from an area of memory naively thought to be the stack in a futile attempt to keep a program running.

Portable (adj.) that which can be physically moved more than a hundred yards by an unaided olympic athlete without permanent damage to that individual more than 50% of the time.

Printer (n.) a small box attached to a computer and used to start fires in cold weather.

Procedure (n.) a method of performing a program subtask in an inefficient way by extensively using the stack instead of a GOTO. See: Pascal and C.

Processor (n.) a device for converting sense to nonsense at the speed of electricity, or (rarely) the reverse.

Program (n.) that which manipulates symbols rapidly with unforseen results. Also: a bug's way of perpetuating bugs.

Programmer (n.) 1. one who writes programs and trusts them. An optimist. 2. Any employee who needs neither food nor sleep but exists on large quantities of caffeine, nicotine, sucrose, and machine-vended preservatives thinly disguised as foodstuffs.

Programming Language (n.) a shorthand way of describing a series of bugs to a computer or a programmer.

Prompt (n.) a computer request for a random operator error. Also a game where the computer plays the part of Vanna White and the operator, a contestant. There are no prizes for winning.

Push (v.) to put into an area of memory believed to be the stack for the ostensible purpose of later retrieval. Tonkin's rule: In any program there are always more 'pushes' than 'pops.' See: Recursion.

Quantum leap (adj.) literally, to move by the smallest amount theoretically possible. In advertising, to move by the largest leap imaginable (in the mind of the advertiser). There is no contradiction.

Recursion (n.) a programming method which tests the limits of available memory in an iterative way by using the stack. When the program fails, all memory has been used. Memorize this definition, then see: Recursion.

Register (n.) a part of the central purging unit used to distort or destroy incoming data by arbitrary rules. See: Increment.

Relational (adj.) purchased from, or sold to, blood kin. See: True relational.

Sector (n.) a disk arc on which is inscribed 'lose a turn' or 'bankrupt.' See: Hard disk, Head, Glitch.

Segment (n.) a way of restricting or complicating access to memory in an attempt to break a programmer's will to live. Outlawed by both the A.S.P.C.A and the U.N. but still practiced in some backward areas of the world. See: Offset.

Software (n.) anything other than hardware. That which hardware manufacturers can blame can blame for physical failures.

Sort (v.) to order a list of data in such a way as to destroy all relationships between the items. (n.) The process which accomplishes this, esp. if it takes a very long time.

Source Code (n.) a record of a programmer's thought for a period of time. A stream-of-consciousness novel or short story.

Spreadsheet (n.) a way of forcing repeatable answers from insufficient data for superficial purposes. Also, a game played during office hours by bored or restless yuppies.

Stack (n.) any area of memory which grows and eventually destroys both code and data. (v.) To place in such an area.

Standard (n., adj.) a design target which manufacturers may embellish, improve upon, or ignore as they wish, so long as it can be used profitably in their advertising.

Transportable (adj.) said of software - that which can be put on a new machine in less time than it took to write in the first place. Said of hardware - that which can theoretically be moved more than ten feet in one minute by some combination of machinery or explosives. The meanings are equivalent.

Truly relational (adj.) relational, but where the paternity is indubitable.

TSR (n.) acronym for Terminate and Stay Resident. A way of turning a useless computer with plenty of memory into a computer with no memory at all.

Turbo- (prefix) computer software which uses air under pressure (supplied by a special fan) to achieve high performance.

User-friendly (adj.) trivialized, slow, incapable, and boring. See: Icon, Mouse.

UNIX (n., v.) a DOS which needs more memory than you have and run more slowly than you can bear. To UNIX: to grossly enlarge and slow down out of all proportion, esp. by using C.

User (n.) one who knows from experience that programs cannot be trusted. A realist.

Vendor (n.) a manufacturer's lackey.

Virtual (adj.) emulated. See: Emulate.

Warranty (n.) a list of vendor's promises with carefully-worded exceptions which cancel each of the promises in turn. See: License.

Windowing (n., adj.) a way of making a large and easily-read display into many small, cluttered, and confusing ones.

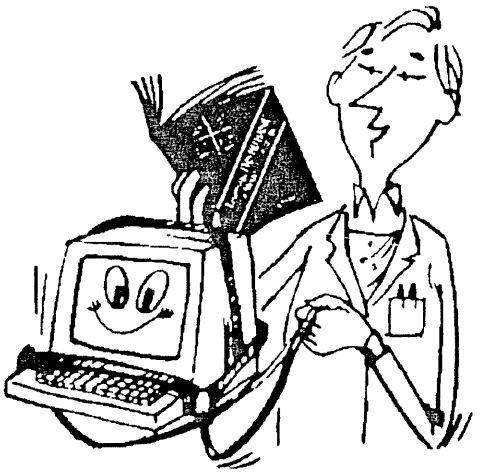
Word Processor (n.) A program which makes a $5,000 computer into a $250 typewriter. A computer game for beginning operators.

WORM (n.) acronym for Write Once, Read Mangled. Used to describe a normally-functioning computer disk of the very latest design.

XYZZY (n.) a common user prompt.

Yarrow (n.) kind of stalks - used by computer diagnosticians when performing the ritual of the I Ching. See: Diagnostics.

Zaxxon (n.) a sophisticated simulation and design program used by the brightest programmers to test the consistency of internal logic and memory. Management prefers to use games such as 'spreadsheet' for the same purpose.



HINTS & TIPS

For this issue, we have compiled a series of short articles, all pertaining to telecommunications. We start with an explanation of just what the lights on our modems mean.

THE MODEM LIGHTS
by H.L. King

The lights on the front panel of your modem tell you a lot if you know how to decode them. Is it sending or receiving data? Is it properly connected to your PC? When you first turn your modem on, the HS, TR, and MR lights should be illuminated, and under most circumstances, they should stay that way. Here's what the lights mean and how to troubleshoot.

- **HS -- HIGH SPEED**

Your modem is operating at its highest transmission rate -- 1200, 2400, 0r 9600 bps -- unless this light (and none of the others) goes off, which means you're transmitting at a slower speed.

- **TR -- TEMINAL READY**

This light indicates that your modem and TRS-80 are connected and ready to swap instructions. The TRS-80 sends a Data Terminal Ready signal from the communications software to the modem, which tells the modem that the software is loaded. If it's not on, turn off the power, check connections, and try again.

- **MR -- MODEM READY**

If this light is not on when you turn on your modem, check the power adapter connection. In the first few seconds after you turn the power on, your modem runs a check routine, during which the MR light may blink for a while, then glow steadily. If the MR light is not lit after this routine, turn off your modem and power up again.

- **AA -- AUTO ANSWER**

If you intend to host an incoming modem call, enable Auto-Answer. Not all Communications programs illuminate the AA light, even though they enable the modem's first S register, S0, which controls the Auto-Answer facility. To be sure your modem will pick up an incoming call after one ring, enter terminal mode, type in ATS0 = 1, and press Enter. This should turn on the light.

- **CD -- CARRIER DETECT**

This light shines when your modem connects with another system. This usually takes about 15 seconds after first dialing out. Use this light to check that you're still connected if there's a pause in your online activity.

- **OH -- OFF HOOK**

This light is on whenever the modem takes control of the phone line -- the equivalent of taking a handset of the hook of a telephone.

- **RD - RECEIVE DATA**

This light blinks when the modem is receiving data. When you're downloading a file, this light is often the only sign that something is happening.

- **SD - SEND DATA**

The SD light blinks when the modem is receiving commands from its host computer or transmitting data to another modem.

SAVE SOME MONEY
by Mark Colen

Modems run on DC power and require adapters to convert electricity from AC power lines. When your modem is not in use, those adapters continue to draw power -- as much as $25 to $40 worth per year. Regardless of whether the device it powers is turned on, a power adapter plugged into a live socket draws between 20 and 30 watts of power, which over a year adds up to more kilowatt-hours than you'd think. So unplug the adapter when you're not using it -- or to make thinks more convenient, plug the adapter into a power strip. When you've finished computing for the day, turn everything off from the power strip's switch.

CALL WAITING
by H.L. King

The signal generated by the telephone company's Call Waiting feature is a modem user's nightmare.: It usually breaks the modem connection, leaving you with a 'No Carrier' message and no other clues. However, it is possible to turn off Call Waiting for the duration of any call, including your on-line session.

Check your telephone book for a feature called Tone Block. If you punch in either ***70** or **1170** before placing a call, you won't hear the Call Waiting signal. Anyone who calls you will simply get a busy signal. If you can't find Tone Block in your phone book, call your phone company business office and ask if this option is available in your area.

To call the number 555-5555 and take advantage of Tone Block during the call, your Hayes-compatiblemodem would need the dialing command

ATDT 1170, 555-5555 (for a tone dialing system) or
ATDP 1170, 555-5555 (for pulse dialing).

The 1170 activates Tone Block, and the comma tells the modem to pause briefly while Tone Block takes effect. You can disable Call Waiting for each number in your communications program's directory by prefacing each entry with the characters 1170, (don't forget the comma).

AW, SHUT UP
by Bruce Robbins

Don't give yourself a headache with frequent dialing on a noisy modem. Soften the volume of a Hayes-compatible modem -- or silence it completely -- with a simple four-digit command. Replace the default medium speaker volume setting (ATL2) by entering one of the following commands in your communications software's terminal mode.

ATL0 or ATL1 sets the speaker low; on most machines, there is no audible difference between the two settings. To turn off the speaker completely, use ATM2. If you want to increase the speaker volumeto high, use ATL3. Press <ENTER> to send the command to the modem, then dial out as usual.

FAST DIALING
by H.L. King

Hayes-compatible modems can be real dialing slouches on your touchtone phone line, but they don't have to be. You can double the speed of dialing a BBS or another modem by changing the setting of register S11, which controls the dialing delay. Get into terminal mode on your communications program and type the command

ATS11=50

and press <ENTER> to send the command to the modem. Now call up a BBS in the usual way. The Modem should whiz through the number. If this works well, make the command part of your communications program's setup string.

TEST THE SERIAL PORT
by Bruce Robbins

If you're having problems with a modem, it's not always easy to diagnose. The problem might concern software, the modem, or the serial port or cable. If you're not intimidated by a little soldering, it's fairly easy to make a serial loopback test to plug to test the serial port.

Buy a female RS-232 connector (RS 276-1548) and a hood (RS 276-1549) from Radio Shack. The cost is only a few dollars. Connect a jumper wire from pin 2 to pin 3. Connect another wire from pin 6 to pin 20. Connect a third jumper wire that connects pins 4, 5, and 8. Put the hood on the connector to protect the wires and allow for easy insertion and removal of the plug, and mark the plug so you can identify it later.

Now insert the plug into the serial port and fire up your favorite terminal program. Set the communications to any speed you like and start typing. When the port is set for half duplex, you should see two of everything you type. When it's set for full duplex you should see just one of each character. If you don't see any characters -- or just one in half duplex -- then the problem most likely has to do with the serial port. Since the loopback plug you built may also have a problem, double-check its wiring and test it on a serial port you know is in good order before relying on it.

XMASTREE/BAS
By Henry Herrdegen

```
0 'XMASTREE-twinkling Xmas tree, Mod II, MCN Dec 82
1 'changed for Mod III, H.H.H., 86 02 26.
10 CLS:
DIM S(225):
P$=CHR$(43):
S$=CHR$(42):
A=-3
20 FOR L=1 TO 11
30 A=A+2
40 FOR W=32+(64*L)-((A/2)+.5) TO 32+(64*L)+
((A/2)+.5)
50 PRINT@W,S$:
Y=Y+1:
S(Y)=W
60 NEXT W
70 NEXT L
80 PRINT@736,CHR$(182)@800,CHR$(191)
90 PRINT@906,"* * *  M E R R Y  C H R I S T M A S  *
* *";
100 Y=RND(120)+1:
Z=RND(120)+1
110 IF Y=111 THEN Y=Y-RND(109)
120 IF Z=111 THEN Z=Z-RND(109)
130 FOR I=1 TO 3
140 X=RND(120)+1:
IF X=111 THEN X=X-RND(109)
150 PRINT@906,P$;@908,P$;@910,P$;@945,P$;
@947,P$;@949,P$;
160 PRINT@S(1),P$;@S(X),P$;@S(Y),P$;@S(Z),P$;:
GOSUB 210
170 PRINT@906,S$;@908,S$;@910,S$;@945,S$;
@947,S$;@949,S$;
180 PRINT@S(1),S$;@S(X),S$;@S(Y),S$;@S(Z),S$;:
GOSUB 210
190 NEXT I
200 K$=INKEY$:IF K$="Q" THEN CLS:
END
ELSE 100
210 FOR T=1 TO 30
NEXT
RETURN
```

HUNTING FOR BURIED TREASURE
PEEKING & POKING MODEL 4
LS-DOS 6.x.x - Basic
By Lance Wolstrup

In the last issue of TRSTimes, I ended up writing two articles. The first was about a utility I had created for my wife, TRSLIST/BAS, while the second was a compilation of most of the PEEKS and POKES we have discovered during the life of this column.

As it turned out, both articles were very well received, generating a lot of response. While I cannot respond to every comment or question I received in the mail, I will use this installment to satisfy (hopefully) the many readers who wrote in and asked something like this: 'the PEEKS and POKES are interesting - BUT - how do we REALLY use them?

This brings me to TRSLIST/BAS. The mail brought several requests to publish a Model 4 translation. Not a bad idea. So, after thinking about it for a while, I decided to combine the rewrite of TRSLIST/BAS with a demonstration of how to implement some of the more exotic Model 4 PEEKS and POKES.

The new program is called TRSLIST4/BAS. It does exactly the same thing the Model III version does; that is, it prints lists created on any ASCII text editor.

TRSLIST4/BAS begins by prompting for the name of the file to be loaded. Pressing ENTER without a filename, or pressing ESC (SHIFT UP-ARROW) exits the program to Basic.

Entering a filename will cause the program to search the drives for a file by that name. If the file is not found, an error message will be displayed and the program flows back to the filename prompt.

If the file is found, TRSLIST4/BAS will attempt to load it into memory. If the file is a TRSLIST file, it will load properly and a menu of available options will be displayed. If the file is not a TRSLIST file, the program will, without a doubt, crash.

Assuming the file is in the agreeable format, the very top line of the screen will display the name of the file, as well as the number of records, and the number of tabs used to display each record.

The menu allows the user to press S to list the records to the screen, P to list the records to the printer, N to load a new file, or Q to quit the program.

At the very bottom of the screen you will see something slightly unusual. It is a trick I used when writing DR. PATCH; the real-time clock is displayed and running correctly at the bottom line of the screen, instead of its normal position on the top line.

Another trick is that the bottom three lines of the screen are no longer recognized. Basic (and LS-DOS) thinks that the screen is only 21 lines long - thus the bottom three lines are not overwritten and scrolled when data is displayed.

A third trick is employed to scroll protect the top three lines, keeping the information about the file from scrolling off the screen. Go ahead, load a TRSLIST file, press <S> and see how only screen lines 3 through 20 are active and available for scrolling data.

A fourth trick is used for the screen display and printer routines. Normally there would be a routine handling the screen display and one for sending the data to the printer. In TRSLIST4/BAS, there is only one routine. Everything pertaining to the data is handled with the LPRINT command. Yes, even displaying the data on the screen is done with LPRINT.

A fifth trick keeps track of when the program is altered; that is, the time and the date of the last revision. This is for programmer convenience and has nothing whatsoever to do with the program itself. I just threw it in there to demonstrate that it works.

Finally, I have included two /LST files just for fun. The first, ACADEMY/LST, should be of interest to anyone who enjoys movies. It displays the year, name of movie, and the studio of the academy award winner for each year since the very first in 1927. Very 'in' -- very 'Hollywood'. The second, ODDS/LST, is a list of odds for all you TRS-80 gamblers who enjoy Blackjack and Craps.

Before I begin the technical discussion of the program, I better explain how to create a TRSLIST file.

A TRSLIST file (/LST) is an unformatted ASCII file. It can be created by any text-editor or word-processor, and need only follow a few rules. The very first piece of data must be a number indicating the number of records in the file. The second piece of data must be a number indicating the number of columns you wish to print; the next pieces of data are also numbers, indicating the tab positions for each column to print; folowing these numbers is the data for the actual list.

Now we come to the line-by-line, technical discussion of TRSLIST4/BAS.

Lines 10 through 13 contain the routine to update the date and time. As the program goes through this code each time it is RUN, the date and time is automatically updated. When the program is changed and then subsequently saved back to disk, the changes to date and time is saved also. The routine works as follows: Line 10 initializes variable UD$="Last update = 11/28/92 01:58:38". The date and time in UD$ could initially be any date and time, as long as the format of UD$ is correct; that is the length of UD$ must be 32 characters, "Last update = " +17 characters that will eventually be changed to the current date and time.

Line 11 stores the current date and time in variable DT$.

Line 12 finds the actual address of variable UD$ and stores it in variable X. Since we know know the address of variable UD$, we know that the date and time characters are stored beginning at X+14. Thus, we set up a loop that extracts the date and time characters from DT$ and then POKE them into UD$.

Line 15 stores the screen width in variable SW, and then jumps over the subroutines to line 100, which is the beginning of the program.

Line 20 through 61 contain the subroutines used in TRSLIST4/BAS.

Lines 20 through 23 hold the program video display routines. Variables V and H respectively contain the vertical and horizontal position of the cursor.

Line 20 setting H=0 positions the cursor at the extreme left side of the screen when the routine jumps to line 23.

Line 21 computes variable H to contain the cursor position for centering the text in A$.

Line 23 displays the text stored in variable A$ at the cursor position determined by V and H. The routine then returns to the caller.

Lines 30-37 contain the multiple keystroke INKEY$ routine. This routine is entered with three parameters passed from the caller; the vertical and horizontal position of the cursor is passed in V and H, and the number of the maximum allowable user keystrokes is stored in ML.

Line 30 displays the maximum allowable keystrokes as periods beginning at the cursor position; the cursor is then made visible; the flag (FL) is set to 0, as is the length of the current user keystrokes (L); A$, which is used to build the user input, is nilled out.

Line 31 forces upper case by setting bit 6 of memory location &H74 (K flags). It then polls for keystroke. If none is found, the program goes back to line 31 and repeats the procedure. All user input using this routine will be in upper case, eliminating the necessity for the program to check for lower case keystrokes.

Line 32 tests if ENTER or ESC has been pressed. If ENTER was pressed then the cursor is made invisible and the program flow is returned to the caller. If ESC was pressed, the flag (FL) is set to 1; the cursor is made invisible and program flow is returned to the caller. Note that line 32 is the only way to exit the routine; the calling routine will know how the subroutine was terminated: if FL=0 the routine was terminated with ENTER; if FL=1 then the subroutine was terminated with ESC.

Line 33 checks if the LEFT ARROW key (chrs$(8)) was pressed at a time when there is no input (L=0). This, of course, is not allowed, and program flow is directed back to line 31.

Line 34 - we again check if the LEFT ARROW key (chr$(8)) was pressed. If so, we delete the character to the left. The cursor is turned off (chr$(15)); the horizontal position of the cursor is decremented; the character to be deleted is overwritten by a period; the cursor (invisible) is positioned on top of the newly-written period; the length of the user input is decremented (L=L-1); the user buffer (A$) is decremented; the cursor is made visible (chr$(14)), and the program flow is returned to line 31.

Line 35 checks if a character with a value less than the SPACEBAR (32) was pressed. As these are mostly control characters, they are neither needed nor wanted in this program; thus, if one was pressed, the program flow is returned to line 31.

Line 36 makes sure that we don't exceed the maximum length of user keystrokes. If we have reached the max (L=ML), then program flow is returned to line 31.

Line 37 displays the keystroke (I$); the horizontal position of the cursor is incremented (H=H+1); the keystroke is added to the input buffer (A$=A$+I$); the length of the total input is incremented (L=L+1), and program flow is directed back to line 31.

Lines 40 through 43 contain the routine to move the real-time clock and then display it.

Line 40 sets variable S with the beginning screen position to display the clock. Note that the vertical position of the clock is determined by variable R, which is passed by the calling routine. The DOS routine that handles the position of the clock demands that the position be offset by &HF800; thus, If R=24 then the algorhitm S=SW*R-8+&HF800 can be visualized as S=80*24-8+&HF800.

Subtracting 8 is necessary since the length of the time is eight characters long. Variable S now holds the address of the starting position of the clock. As we will be doing division to split the address into MSB (most significant byte) and LSB (least significant byte), we must make sure that it is a positive number; therefore we add 65536 if S happens to be negative.

Line 41 stores the MSB of S in variable S2 and the LSB of S in variable S1.

Line 42 POKES the new address of the starting position of the clock in memory locations &H78B and &H78C.

Line 43 returns program flow to the caller.

Lines 50 through 56 contain the routine to bottom scroll protect 24-R number of lines. The value of R is passed from the calling routine.

Line 50 sets variable S = the length of the desired screen (R*80); Since the DOS routines that handle the screen length demand to know the end address of the screen (offset by &HF800), we set variable T equal to that value. Again, to make sure that T is a positive number, we add 65536 if T is found to be negative.

Line 51 splits the value of S into MSB and LSB and stores these values in S2 and S1 respectively. It also splits the value of T into MSB and LSB and stores these values in T2 and T1 respectively.

Line 52 POKES the new screen length into memory locations &HCD7 and &HCD8; also &HDA8 and &HDA9.

Line 53 POKES the new screen ending address into memory locations &HCC5 and &HCC6, as well as &HD20 and &HD21.

Line 54 - since the DOS routine to move the cursor down one line checks on End of Screen + 1, we must make sure that our End of Screen + 1 will be in the correct MSB and LSB format.

Line 55 POKES the new End of Screen + 1 address into memory locations &HD0B and &HD0C.

Line 56 returns the program flow to the calling routine.

Lines 60 and 61 contain the printer routine.

Line 60 erases the screen from line 4 to the end of the display, and then displays the 'Ready printer' message; the horizontal position of the cursor is updated in variable H to be located at the and of the message.

Line 61 sets the maximum user keystokes to 1 (ML = 1) and calls the multiple keystroke INKEY$ subroutine in line 30. Since the prompt called for pressing the ENTER key, only the ENTER and ESC keys are allowed. Any other keys will cause a jump back to line 60.

Line 100 begins the actual program. The entire screen is erased; bit 4 of memory location &H7C is set (S flags), which disables the BREAK key.

Line 101 sets bit 3 of memory location &HB94, forcing the special characters set active.

Line 110 turns off the cursor and then displays the line of chr$(140)'s on screen line 22.

Line 111 moves the cursor down to line 23 and then displays the 'TRSTimes presents' message left justified; it displays the program name centered, and displays the current date at horizontal position 61.

Line 112 moves the real-time clock to the bottom line (R = 24) by using the subroutine in line 40. Bit 4 of memory location &H7F (V flags) is set; this displays the clock.

Line 113 - since we now have setup the bottom of the screen, we can bottom scroll protect the last three lines. We do this by setting variable R = 21 and then calling the subroutine in line 50.

Line 114 displays the 'TRSLIST file in memory' message at the top of the screen, left justified..

Line 115 displays the graphic line of chr$(140)'s on the second screen line.

Line 116 sets bits 0 and 1 in memory location &HB94; this scroll protects the top three lines.

Line 120 stores the value of memory location &H74 (K flags) in variable CA; this is done to determine the original setting of the Caps Lock key. The unprotected portion of the screen is then erased, and any message that may exist on line 0, at tab 25 and higher, is erased. Then at tab 25, the message 'NONE' is displayed.

Line 125 prompts for 'the name of the file to list' on line 4. The maximum user keystrokes allowed to respond to this prompt is 12 (ML = 12) and the program calls the multiple keystrokes inkey$ routine in line 30.

Line 130 first determines how the routine in line 30 exited. If ENTER was pressed without a filename, or ESC was pressed then we must exit the program. But before we exit, we must reset all the memory locations we have played tricks with to their normal values. First, POKE the startup Caps Lock value (CA) into &H74; re-enable the BREAK key by turning off bit 4 of &H7C (AND 239); turn off the clock display by resetting bit 4 of &H7F (AND 239); turn off bits 0,1,2, and 3 of &HB94 (AND 240); this turns off the top scroll protect, as well as turning off the

special character set. Finally, turn off all bottom scroll protection by setting R=24 and calling the routine in line 50. Erase screen and end program. If ESC was not pressed or data was typed before ENTER was pressed, program copies the filename from A$ to F$.

Line 135 sets flag (variable F) to 0.

Line 140 - the error routine will keep the program from crashing if we attemp to open a file that does not exit. Should that happen, the routine branches to line 1000.

Line 150 opens the selected file.

Line 160 - if we get this far the file obviously existed and we close the error trap. As I mentioned before, the only thing that can crash the program at this point, is if the selected file is not a TRSLIST file.

Line 170 - erase screen from line 4 to end of the short screen, and then display the 'reading' message.

Line 180 - read number of records into variable R, and read number of tab-columns into variable C.

Line 190 - dimension the variables.

Line 200 - read the tab settings into variables TB(X).

Line 210 - read records into R$(X,Y).

Line 220 closes the file.

Line 225 displays on line 0 the number of records and the number of tab settings found in the file.

Line 230 moves the cursor to line 3 and erases to the end of the screen.

Line 240 displays the program options - 'List to Screen or Printer, New file or Quit'.

Line 250 accepts the keyboard press and then determines which action to take.

Line 260 is reached if either List to Screen or Printer is chosen. If variable F=0 (list to screen) then we POKE the address of the screen DCB into the printer DCB. This allows us to LPRINT to the screen. If F=1 (print to printer) we use the subroutine in line 60 to display the 'Ready printer' message. The linecount (LC) is set to 0, and the page number (PG) is set to 1.

Line 265 erases from line 4 to the end of the screen.

Line 270 sets up the loop to step through each record.

Line 275 if F (output to printer) then send carriage return and line feed to the printer and increment the linecount by one. If the linecount is '1' then print the filename centered on the printer, as well as the page number at tab(70); increment the linecount - print a blank line - and increment the linecount again.

Line 277 if F (output to printer) and the linecounter is larger than 60 then set up loop to perform top of form; reset linecounter (LC) to 0 and increment the page number (PG) by 1. Go back to line 275 and repeat.

Line 280 sets up loop to display or print one record at the appropriate tab settings.

Line 290 if F=0 (output to screen) we check if we have put 16 records on the screen. If so (X/16=INT(X/16)) the use the subroutine in line 70 to pause for a keystroke.

Line 300 sends the outer loop (X) back for more.

Line 305 performs a carriage return and a line feed. The linecounter is incremented by 1. The address of the real printer routine is POKEd back into the printer DCB and if output was going to the printer, we jump to line 330.

Line 310 display the 'Press <ENTER>' message.

Line 320 set the max allowable keystroke to 1 and uses the subroutine in line 30 to get the user response. If ESC or ENTER was pressed we go back to line 230 otherwise we keep prompting to press ENTER.

Line 330 sets up a loop to perform a top of form and then returns program flow to line 230.

Line 1000 begins the error trap routine by erasing the screen from line 4 to the end of the screen.

Line 1010 displays the 'Unable' message. Note that F will always be 0 in this routine. It is possible to set up a whole series of error messages based on the value of F.

Line 1100 displays the 'Press <ENTER>' message.

Lines 1110-1130 ends the error trap and flow returns to line 120

TRSLIST4/BAS

```
1 'TRSLIST4/BAS for Model 4
2 'Copyright 1992 TRSTimes & Lance Wolstrup
3 'All rights reserved
4 '
10 UD$="Last update = 11/29/92 00:08:22"
11 DT$=DATE$+" "+TIME$
12 X=PEEK(VARPTR(UD$)+1)+PEEK(VARPTR(UD$)
+2)*256
13 FOR Y=0 TO 16:
POKE X+Y+14, ASC(MID$(DT$,Y+1,1)):NEXT
```

```
14 '
15 SW=80:GOTO 100
20 H=0:GOTO 23
21 H=INT((80-LEN(A$))/2)
23 PRINT@V*80+H,A$;:RETURN
24 '
30 A$=STRING$(ML,46):GOSUB 23:
A$=CHR$(14):GOSUB 23:FL=0:L=0:A$=""
31 POKE &H74,PEEK(&H74) OR 32:
I$=INKEY$:IF I$="" THEN 31
32 IF I$=CHR$(13) THEN PRINT CHR$(15);:RETURN
ELSE IF I$=CHR$(27) THEN PRINT CHR$(15);:
FL=1:RETURN
33 IF I$=CHR$(8) AND L=0 THEN 31
34 IF I$=CHR$(8) THEN PRINT CHR$(15);:
H=H-1:PRINT@V*SW+H,CHR$(46);:
PRINT@V*SW+H,"";:L=L-1:A$=LEFT$(A$,L):
PRINT CHR$(14);:GOTO 31
35 IF I$<CHR$(32) THEN 31
36 IF L=ML THEN 31
37 PRINT I$;:H=H+1:A$=A$+I$:L=L+1:GOTO 31
38 '
40 S=R*SW-8+&HF800:IF S<0 THEN S=S+65536!
41 S2=INT(S/256):S1=S-S2*256
42 POKE &H78B,S1:POKE &H78C,S2
43 RETURN
44 '
50 S=R*80:
T=&HF800+S-1:IF T<0 THEN T=T+65536!
51 S2=INT(S/256):S1=S-S2*256:
T2=INT(T/256):T1=T-T2*256
52 POKE &HCD7,S1:POKE &HCD8,S2:
POKE &HDA8,S1:POKE &HDA9,S2
53 POKE &HCC5,T1:POKE &HCC6,T2:
POKE &HD20,T1:POKE &HD21,T2
54 IF T1+1>255 THEN T1=0:T2=T2+1
ELSE T1=T1+1
55 POKE &HD0B,T1:POKE &HD0C,T2
56 RETURN
57 '
60 V=4:A$=CHR$(31):GOSUB 20:
A$="Ready printer - then press <ENTER> ":
GOSUB 21:H=H+LEN(A$)
61 ML=1:GOSUB 30:IF FL OR A$="" THEN RETURN
ELSE 60
70 I$=INKEY$:IF I$="" THEN 70 ELSE RETURN
100 CLS:
POKE &H7C,PEEK(&H7C) OR 16 'disable break key
101 POKE &HB94,PEEK(&HB94) OR 8 'special chrs
110 PRINT CHR$(15):
V=22:A$=STRING$(80,131):GOSUB 20
111 V=V+1:
A$="TRSTimes presents: "+CHR$(143)+CHR$(244)+
CHR$(245)+ CHR$(246):GOSUB 20:
A$="T R S L I S T 4":GOSUB 21:
H=61:A$=DATE$:GOSUB 23
112 R=24:GOSUB 40:
POKE &H7F,PEEK(&H7F) OR 16 'move clock & display
113 R=21:GOSUB 50 'scroll protect bottom 3 lines
114 V=0:A$="TRSLIST file in memory =":GOSUB 20
115 V=V+1:A$=STRING$(80,140):GOSUB 20
116 POKE &HB94,PEEK(&HB94) OR 3
120 CA=PEEK(&H74):V=3:A$=CHR$(31):GOSUB 20:
V=0:H=25:A$=CHR$(30)+"NONE":GOSUB 23
125 V=4:A$="Enter name of file to list: ":GOSUB 20:
H=29:ML=12:GOSUB 30
130 IF FL OR A$="" THEN POKE &H74,CA:
POKE &H7C,PEEK(&H7C) AND 239:
POKE &H7F,PEEK(&H7F) AND 239:
POKE &HB94,PEEK(&HB94) AND 244:
R=24:GOSUB 50:CLS:END ELSE F$=A$
135 F=0
140 ON ERROR GOTO 1000
150 OPEN"I",1,F$
160 ON ERROR GOTO 0
170 V=4:A$=CHR$(31):GOSUB 20:
A$="Reading "+F$:GOSUB 20
180 INPUT#1,R:INPUT#1,C
190 DIM R$(R,C),TB(C)
200 FOR X=1 TO C:INPUT#1,TB(X):NEXT
210 FOR X=1 TO R:FOR Y=1 TO C:
INPUT#1,R$(X,Y):NEXT:NEXT
220 CLOSE
225 V=0:H=25:A$=CHR$(30)+F$:GOSUB 23:
H=45:A$="Records ="+STR$(R):GOSUB 23:
H=65:A$="Tabs ="+STR$(C):GOSUB 23
230 V=3:A$=CHR$(31):GOSUB 20
240 V=4:F=0:
A$="List to (S)creen or (P)rinter, (N)ew file or (Q)uit ":
GOSUB 21:H=H+LEN(A$)
250 ML=1:GOSUB 30:
IF FL OR A$="N" THEN RUN 120
ELSE IF A$="Q" THEN A$="":GOTO 130
ELSE IF A$="P" THEN F=1
ELSE IF A$<>"S" THEN 240
260 IF F=0 THEN POKE &H219,&H88:
POKE &H21A,&HB ELSE GOSUB 60:LC=0:PG=1
265 V=4:A$=CHR$(31):GOSUB 20
270 FOR X=1 TO R
275 IF F THEN LPRINT:LC=LC+1:
IF LC=1 THEN LPRINT TAB((80-LEN(F$))/2)F$;TAB(70)
"Page";PG:LC=LC+1:LPRINT:LC=LC+1
277 IF F AND LC>60 THEN FOR Z=LC+1 TO 66:
LPRINT:NEXT:LC=0:PG=PG+1:GOTO 275
280 FOR Y=1 TO C:LPRINT TAB(TB(Y))R$(X,Y);:NEXT
290 IF F=0 AND X/16=INT(X/16) THEN GOSUB 70
300 NEXT
305 LPRINT:LC=LC+1:
```

```
POKE &H219,&H1:POKE &H21A,&HE:IF F GOTO 330
310 V=20:A$="Press <ENTER> to continue":
GOSUB 21:H=H+LEN(A$)
320 ML=1:GOSUB 30:IF FL OR A$="" THEN 230
ELSE 310
330 FOR Z=LC+1 TO 66:LPRINT:NEXT:GOTO 230
1000 V=4:A$=CHR$(31):GOSUB 20
1010 IF F=0 THEN A$="Unable to find "+F$:
GOSUB 21
1100 V=6:A$="Press <ENTER> to continue":
GOSUB 21:H=H+LEN(A$)
1110 ML=1:GOSUB 30:
IF FL OR A$="" THEN RESUME 1120 ELSE 1100
1120 ON ERROR GOTO 0
1130 GOTO 120
```

ACADEMY/LST

```
62
3,0,15,60
1927/28,Wings,Paramount
1928/29,Broadway Melody,MGM
1929/30,All quiet on the Western Front,Universal
1930/31,Cimarron,RKO
1931/32,Grand Hotel,MGM
1932/33,Cavalcade,Fox
1934,It Happened One Night,Columbia
1935,Mutiny on the Bounty,MGM
1936,The Great Ziegfeld,MGM
1937,Life of Emile Zola,Warner
1938,You Can't Take It With You,Columbia
1939,Gone With the Wind,Selznick Int'l
1940,Rebecca,Selznick Int'l
1941,How Green Was My Valley,20th Century-Fox
1942,Mrs. Miniver,MGM
1943,Casablanca,Warner
1944,Going My Way,Paramount
1945,The Lost Weekend,Paramount
1946,The Best Years of Our Lives,Goldwyn-RKO
1947,Gentleman's Agreement,20th Century-Fox
1948,Hamlet,Universal Int'l
1949,All the King's Men,Columbia
1950,All About Eve,20th Century-Fox
1951,An American in Paris,MGM
1952,Greatest Show on Earth,Paramount
1953,From Here to Eternity,Columbia
1954,On the Waterfront,Columbia
1955,Marty,United Artists
1956,Around the World in 80 Days,United Artists
1957,The Bridge on the River Kwai,Columbia
1958,Gigi,MGM
1959,Ben-Hur,MGM
1960,The Apartment,United Artists
1961,West Side Story,United Artists
1962,Lawrence of Arabia,Columbia
1963,Tom Jones,United Artists
1964,My Fair Lady,Warner Bros.
1965,The Sound of Music,20th Century-Fox
1966,A Man for All Seasons,Columbia
1967,In the Heat of the Night,
1968,Oliver,
1969,Midnight Cowboy,
1970,Patton,
1971,The French Connection,
1972,The Godfather,
1973,The Sting,
1974,The Godfather-Part II,
1975,One Flew Over the Cuckoo's Nest,
1976,Rocky,
1977,Annie Hall,
1978,The Deer Hunter,
1979,Kramer vs. Kramer,
1980,Ordinary People,
1981,Chariots of Fire,
1982,Gandhi,
1983,Terms of Endearment,
1984,Amadeus,
1985,Out of Africa,
1986,Platoon,
1987,The Last Emperor,
1988,Rain Man,
1989,Driving Miss Daisy,
```

ODDS/LST

```
27
3,0,30,55
POKER,,
Hand,Number posibble,Odds against
Royal Flush,4,649739 to 1
Other Straight Flush,36,72192 to 1
Four of a Kind,624,4164 to 1
Full House,3744,693 to 1
Flush,5108,508 to 1
Straight,10200,254 to 1
Three of a Kind,54912,46 to 1
Two Pairs,123552,20 to 1
One Pair,1098240,4 to 3 (1.37 to 1)
Nothing,1302540,1 to 1
Total,2598960,
,,
DICE,,
Total,Odds against (Single toss),
2,35 to 1,
3,17 to 1,
4,11 to 1,
5,8 to 1,
6,31 to 5,
7,5 to 1,
8,31 to 5,
9,8 to 1,
10,11 to 1,
11,17 to 1,
12,35 to 1,
```

GRAPHICS-90

NOW for Model I or III

The definitive graphics program that Radio Shack SHOULD have released is now, through the courtesy of author Larry Payne and the hard work of Gary Shanafelt, available from TRSTimes. Our agreement is to sell this wonderful package at COST. Neither Larry Payne, Gary Shanafelt nor TRSTimes will make a dime on this venture. The price charged will cover the cost of reproducing the manual, disks, and mailing only. The only one profiting is YOU!!

GRAPHICS-90 Model III $ 11.50
Model I $ 13.00

Shipping: Europe: add $4.50 air mail or $2.75 surface mail
Asia, Austrailia & New Zealand: add $5.25 air mail or $3.50 surface mail
Please specify Model I or III and DOS preference

TRSTimes magazine - dept. GR90
5721 Topanga Canyon Blvd. #4
Woodland Hills, CA 91367

RECREATIONAL & EDUCATIONAL COMPUTING

REC is the only publication devoted to the playful interaction of computers and 'mathemagic' - from digital delights to strange attractors, from special number classes to computer graphics and fractals. Edited and published by computer columnist and math professor Dr. Michael W. Ecker, REC features programs, challenges, puzzles, program teasers, art, editorial, humor, and much more, all laser printed. REC supports many computer brands as it has done since inception Jan. 1986. Back issues are available.

To subscribe for one year of 8 issues, send $27 US or $36 outside North America to REC, Att: Dr. M. Ecker, 909 Violet Terrace, Clarks Summit, PA 18411, USA or send $10 ($13 non-US) for 3 sample issues, creditable.

TRSTimes on DISK #10

Issue #10 of TRSTimes on DISK is now available, featuring the programs from the Jul/Aug,Sep/Oct & Nov/Dec 1992 issues:

TRSTimes on DISK is reasonably priced:

U.S. & Canada: $5.00 (U.S.)
Other countries: $7.00 (U.S.)
Send check or money order to:

TRSTimes on DISK
5721 Topanga Canyon Blvd. #4
Woodland Hills, CA 91367

TRSTimes on DISK #1, 2, 3, 4, 5, 6, 7, 8, & 9 are still available at the above prices

Recreational and Educational Computing

Finding "The Next Number" in Any Sequence

In the premier issue of REC (Vol. 1, No. 1, Jan. 1986), and later repeated in this magazine, I included a program for "finding the next number" in a sequence. I use the quote-marks as a disclaimer, as there is no such thing as "the next number". This discussion followed up on my earlier "Recreational Computing" column in the Dec. 1985 Creative Computing, which was the last of that magazine (and one of the reasons I started REC). A few years later I reprinted Prof. Jim Householder's "Note to a Psychologist" (first printed around 1953). It brilliantly proved that no matter how many terms you give of a sequence, any next number you offer is a "correct" continuation or next number, in this sense: Given a sequence a1, a2, ..., an, and given any number M, there is a "natural" function f that satisfies f(1) = a1, f(2) = a2, ..., f(n) = an, and yet f(n+1) = M.

Nevertheless, these puzzles persist every year. The problem is that, as I just outlined, any answer is correct, though we humans subjectively deem one answer to be a "natural" one or even a "best" one. But, of course, if we took away the ambiguity and said what we meant (i.e., what the pattern was, or what the function really is), we would be giving away the store.

So, to retrieve something of value, I defined the issue, in most of these problems, as one of getting the pattern via a common first difference, second difference, etc.

For example, consider a sequence begun by 1, 2, 4, 7, 11, 16, ... I am thinking of adding 1, then 2, then 3, then 4, then 5, etc. If this is understood, then there is only one such sequence, and the next number should be 16+6 = 22.

The program that follows will work for any such scenario. In the theory of difference equations, this would evidently be equivalent to the requirement that the function (the sequence) be generated by a polynomial function f. An example of one that would not fit is, say, the powers of 2: 1, 2, 4, ... Indeed, the program would provide the number 7, as it would be looking for a constant n-fold difference or (n+1)-fold zero difference. But the program will recognize patterns such as odd numbers (give it 1, 3, 5), even numbers (give it 2, 4, 6), perfect squares (give it 1, 4, 9, 16, 25 - or try giving it even fewer of these), perfect cubes, etc.

The only new feature in this program over the previous version is that it will generate as many terms as you want beyond the provided ones. Bear in mind that you must provide a sufficient number of initial terms of the sequence.

"Next Number in Sequence" Program

```
10 CLS:PRINT "Dr. M. Ecker's 'Finding A Next Number',
(c) 1985, 1987, 1992."
20 PRINT "This program will allow you to input any number N of numbers and it
30 PRINT "then determines the next numbers. These numbers, of course, are based"
40 PRINT "on eventual N-fold difference of zero -> constant n-1st difference."
50 PRINT: 'A(J,K) = entry in Jth row, Kth across
60 INPUT "<Enter> to begin...", XX$: PRINT: PRINT
70 INPUT "How many numbers are given"; N
80 INPUT "How many more numbers are needed"; NN
90 DIM A(N,N+NN): PRINT
100 FOR J=1 TO N
110 PRINT "Element number "; J;
120 INPUT A(1,J)
130 NEXT
140 FOR K=2 TO N
150 FOR J=1 TO N+1-K
160 A(K,J)=A(K-1,J+1)-A(K-1,J)
170 NEXT J
180 NEXT K
190 FOR M=2 TO NN+1
200 A(N,M)=A(N,1)
210 NEXT M
220 FOR M=1 TO NN
230 FOR L=N-1 TO 1 STEP -1
240  A(L,N+M-L+1)=A(L,N+M-L)+A(L+1,N+M-L)
250 NEXT L
260 NEXT M
270 PRINT "The next numbers are: ";
280 FOR M=1 TO NN
290 PRINT A(1,N+M);
300 NEXT M
```

Holiday Computer Shopping Glossary of Computer Terminology

Now that the holiday shopping season is over (by the time you read this, that is), you can find out what the computer hype really meant. Here's a partial lexicon somebody shared with me.

New: Has different color from previous design.
All-new: Has no interchangeable parts or compatibility with previous model.
Foolproof operation: No provision is made for any adjustments at all.
Advanced design: The advertising agency hasn't a clue what it's pushing.
Field-tested: Has to be; the manufacturer lacks testing equipment.
Years of development: They finally have one that works.
Revolutionary: Different.
Maintenance-free: Impossible to fix.
Direct sales only: The manufacturer had a big fight with the distributor.
Easy to use: Not.

Create Your Own Magic Squares!

A magic square is an n x n square in which each space contains a number such that the total in any row, in any column, or on either diagonal is the same. If you have never tried to construct one, you might find it much harder to do than it first appears. For instance, try to place the nine numbers 1 through 9 in the nine squares of a 3 x 3 square. Many people report either failure or taking a very long time.

Now, I must confess that I am not that big a magic squares fan. But, when a reader sent me this one (the first one, namely the one on the left), I was intrigued.

You see, with it, you can create your own personal magic square, and I will show you how. For instance, using the supplementary digits 4 and 9, I obtained the second square (shown on right).

11	8	23
26	14	2
5	20	17

Take any two numbers (e.g., 4 and 9 - or pick your own digits). Multiply the number of tens in the tens digit of each element of the square on the left by your first choice and the units digit of each element by your second choice. Add the results and write this in the corresponding location in the square on the right.

49	72	107
134	76	18
45	80	103

Amazingly, regardless of choice, the new square shown will be magic. For instance, the original upper left entry is 11. Let's make our first choice 4 and our second 9. Multiply the tens digit 10 by 4 to get 40 and the ones digit 1 by 9 to get 9 for a total of 40+9 = 49; this is the upper left entry of the square on the right.

That's all for this time. If you like the sort of recreation, programming, and challenges I've just hinted at here, I close with four special offers.

1) Take the double volume of the menu-driven Magic Math Plus (ordinarily $40 for TRS-80) AND a one-year subscription to Recreational & Educational Computing (ordinarily $27 in USA, $28 Canada, $36 abroad) both for over 40% off: Just $39.95. That's like subscribing and getting Magic Math Plus as a post-holiday present for 68% off!

For those who don't know, Magic Math Plus contains number tricks, mind-reading, and more, including the Game of N, 31 (like Nim), other games of logic, and much more - about 40 programs in all. Everything comes on one self-booting disk compatible with TRSDOS 1.3. All you do is select programs from a menu. At the conclusion of any, just press M to return to the menu.

That's all there is to it. But things are really easy, and you will feel like a math genius with all the great tricks and explanations.

2) If you prefer, take REC with Casino 21, a terrific game of blackjack ordinarily sold for $17 alone. Your cost is just $4.95 for Casino 21 when ordered with REC for one year ($27 USA only), total $31.95.

Casino 21 includes hitting and sticking, play with up to seven players, insurance, splitting, etc.

3) The Conversational Computer will have your friends thinking your computer is omniscient. How does it work? Only you will know. Source code is included for easy study of the effect. This $10 value is just $3.33 with your REC subscription ($27 USA) for one year, total $30.33.

4) TRS-80 Logic Games Disk (self-booting), ordinarily $12 alone, may be had for just $3.50 along with REC ($27 annually in USA) for a combined total of $30.50.

Sorry, but only one package may be selected per REC subscription. All others may try three sample issues or a trial subscription for just $10.

Otherwise, I welcome any correspondence, your favorite puzzles or challenges, etc., plus any relevant questions. Please write me at:

Dr. Michael Ecker
909 Violet Terrace
Clarks Summit, PA 18411

MODEL I FINAL SOFTWARE CLOSEOUT

Many hard-to-find items...when they're gone - they're gone

Visicalc	disk & manual + the Power of Visicalc Vol I & II	$28.00
Advanced Statistical Analysis	cassettes & manual	$25.00
Disk Mailing List - RS 26-1551	disks & manual	$18.00
Zaxxon	disk only	$10.00
Newbasic	from the Alternate Source - disk only	$10.00
The Adventure System	Alternate Source - disk & manual	$15.00
SuperScripsit & Proofread Program	2 disks, no manual	$15.00
DFBLOAD	from Lewis Rosenfelder	$10.00
Trakcess	from the Altenate Source - disk only	$10.00
Isar	from the Alternate Source - disk only	$10.00
Isar Expanded	from the Alternate Source - disk only	$10.00
Snapp-Ware Trial Package	disk only	$10.00
Series-I Editor/Assembler	disk only	$10.00
Series-I Editor/Assembler	cassettes & manual	$15.00
Scripsit	cassette version with manual	$15.00
Scripsit 1.0	disk only	$10.00
Scripsit RS-26-1563	disk, cassettes & manual	$22.00
Superdirectory	from Computer Shack - disk only	$10.00
Editor/Assembler	RS-2202 - 2 disks, no manual	$15.00
TRSDOS 2.3	disk only	$ 8.00
TRSDOS 2.3	disk & manual	$12.00
SOLE	from Misosys - disk only	$12.00
I/O Monitor	from LSI - disk only	$10.00
Hayes Master	from George Fischer - disk only	$12.00
Disk Instruction Course	4 disks, no manual	$15.00
Lescript	from Anitek - disk only	$15.00
Inventory Control	RS 26-1553 - disk only	$10.00
Omniterm Intelligent Terminal	disk only	$ 8.00
Powertool	RS26-2022 - disk & Manual	$20.00
Budget Management	RS26-1603 - 3 cassettes & manual	$15.00
Ehnbas	from Cornsoft Group - disk & manual	$20.00
Z-80 Machine Language Techniques	book	$10.00
Voyage of the Valkyrie	disk & manual	$12.00
Newdos/80 v2.0	disk & manual	$20.00
Videotex	RS26-2220 - cassette & manual	$ 8.00
Profile	RS 26-1562 - disk & manual	$20.00
Newscript	from ProSoft - disks & manual	$22.00
Qwerty	disk & manual	$15.00
Tax Saver	disks & manual	$10.00
In-Memory Information System	cassettes & manual	$15.00
Alpha-Key	RS 26-1718 - cassette & manual	$15.00

TRSTimes magazine, 5721 Topanga Canyon Blvd. #4, Woodland Hills, CA 91367

*** DR. PATCH ***

BRAND-NEW UTILITY FOR TRS-80 MODEL 4 AND LS-DOS 6.3.1

A 'MUST HAVE' FOR ALL LS-DOS 6.3.1 OWNERS.

DR. PATCH MODIFIES LS-DOS 6.3.1 TO DO THINGS THAT WERE NEVER BEFORE POSSIBLE.
COMPLETELY SELF-CONTAINED - MENU-DRIVEN FOR MAXIMUM USER CONVENIENCE.
FAST & SAFE - EACH MODIFICATION IS EASILY REVERSED TO NORMAL DOS OPERATION.

DISABLE PASSWORD CHECK IN FORMAT/CMD
FORMAT DOUBLE-SIDED AS DEFAULT
FORMAT 80 TRACKS AS DEFAULT
DISABLE VERIFY AFTER FORMAT
CHANGE 'DIR' TO 'D'
CHANGE 'CAT' TO 'C'
VIEW DIR/CAT WITH (I) PARAMETER AS DEFAULT
VIEW DIR/CAT WITH (S,I) PARAMETERS AS DEFAULT
CHANGE 'REMOVE' TO 'DEL'
CHANGE 'RENAME' TO 'REN'
CHANGE 'MEMORY' TO 'MEM'
CHANGE 'DEVICE' TO 'DEV'
DISABLE THE BOOT 'DATE' PROMPT
DISABLE THE BOOT 'TIME' PROMPT
DISABLE FILE PASSWORD PROTECTION
ENABLE EXTENDED ERROR MESSAGES

DISABLE PASSWORD CHECK IN BACKUP/CMD
BACKUP WITH (I) PARAMETER AS DEFAULT
BACKUP WITH VERIFY DISABLED
DISABLE BACKUP 'LIMIT' PROTECTION
DISABLE PASSWORD CHECK IN PURGE
PURGE WITH (I) PARAMETER AS DEFAULT
PURGE WITH (S,I) PARAMETERS AS DEFAULT
PURGE WITH (Q=N) PARAMETER AS DEFAULT
IMPLEMENT THE DOS 'KILL' COMMAND
CHANGE DOS PROMPT TO CUSTOM PROMPT
TURN 'AUTO BREAK DISABLE' OFF
TURN 'SYSGEN' MESSAGE OFF
BOOT WITH NON-BLINKING CURSOR
BOOT WITH CUSTOM CURSOR
BOOT WITH CLOCK ON
BOOT WITH FAST KEY-REPEAT

DR. PATCH IS THE ONLY PROGRAM OF ITS TYPE EVER WRITTEN FOR THE TRS-80 MODEL 4 AND LS-DOS 6.3.1. IT IS DISTRIBUTED EXCLUSIVELY BY TRSTIMES MAGAZINE ON A STANDARD LS-DOS 6.3.1 DATA DISKETTE, ALONG WITH WRITTEN DOCUMENTATION.

*** DR. PATCH *** $14.95

SHIPPING & HANDLING: U.S & CANADA - NONE
ELSEWHERE - ADD $4.00
(U.S CURRENCY ONLY, PLEASE)

TRSTimes magazine - dept. DP
5721 Topanga Canyon Blvd. #4
Woodland Hills, CA 91367

DON'T LET YOUR LS-DOS 6.3.1 BE WITHOUT IT!