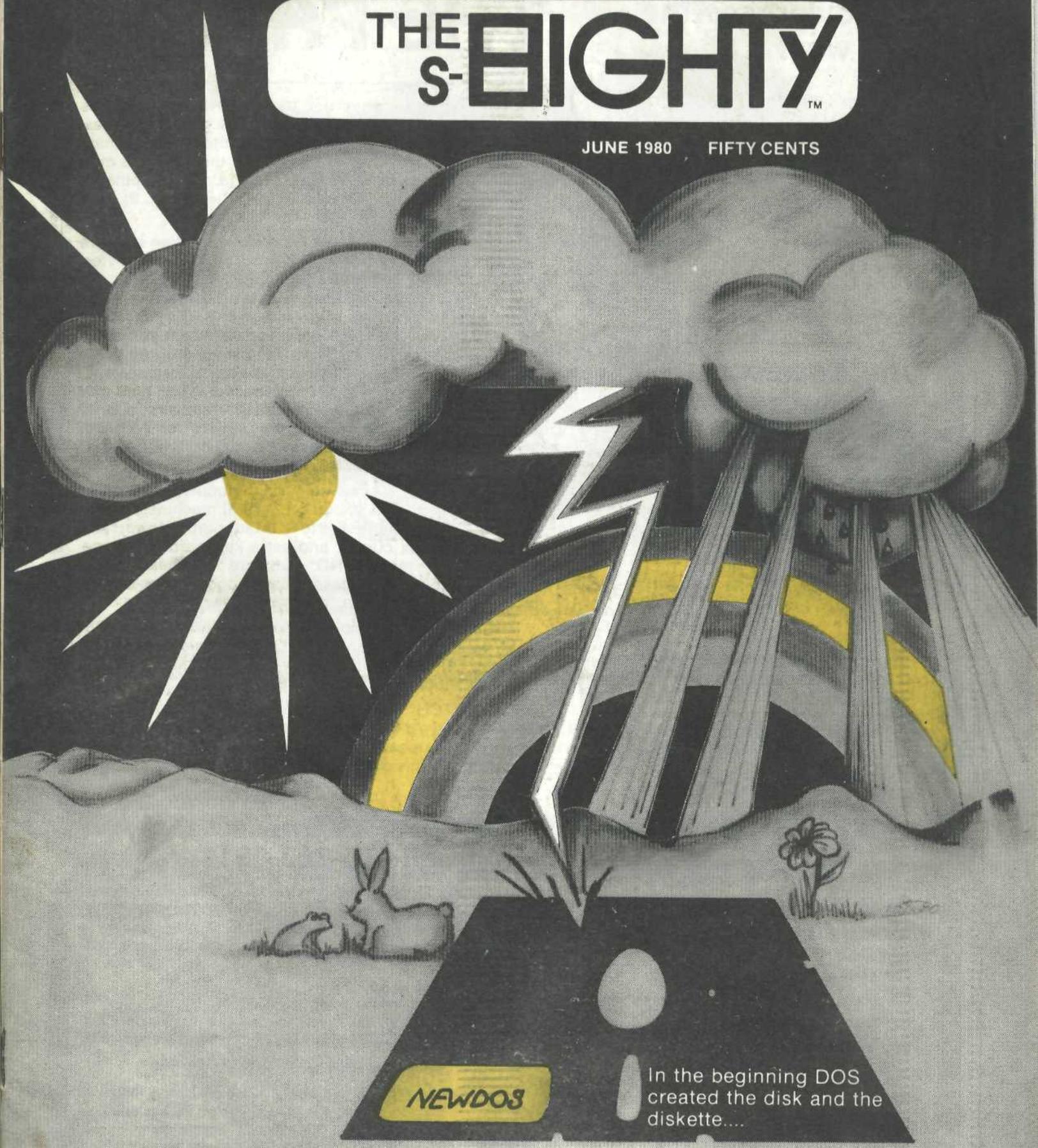


THE S-EIGHTY™

JUNE 1980 FIFTY CENTS



NEWDOS

In the beginning DOS created the disk and the diskette....

RIGHT JUSTIFIED DISK FILES OR BEING LEFT TO YOUR OWN DEVICES

As a follow-up to Lance Micklus' article on DEVICES, I would like to tell you about one very nice advantage to using an operating system such as VTOS 3.0 which is "device independent". Being device independent means that the output stays the same and the user tells the operating system what to route the output to. This is a very powerful function and can be used in a variety of ways. For example, if you do not own a printer you can still run programs designed to output to one. Using the command `ROUTE *PR TO *DO` will take all LPRINT commands and route them to the video. Simple; yes? Reversing the command (`ROUTE *DO TO *PR`) does the exact opposite, thereby causing everything that is usually displayed to be printed on your printer. The command `LINK *DO TO *PR` gives you both devices at the same time. Things start to get interesting when you consider that disk drives are also devices. Any program that uses the operating system's printer driver can have the printed output written to a disk file instead.

Enter ELECTRIC PENCIL. The VTOS 3.0 version of The ELECTRIC PENCIL fits perfectly into this situation. What follows is a description of a technique I use for generating perfectly formatted files for transmission to time sharing systems. You will soon see that the technique can have myriad other uses and, besides that, it's an interesting concept to play around with.

The only necessary tools for this experiment are VTOS 3.0, ELECTRIC PENCIL and, of course, a disk drive. From DOS READY, type `ROUTE *PR TO FILENAME`. FILENAME is any name you want it to be. You have now opened a file and you have told VTOS to route all printer output to that file. Now type PENCIL (or whatever you call it on your disk) and either load in a PENCIL file or type in a new one.

Once you have a text file resident in PENCIL, type control "K" and enter your print formatting commands. For this experiment let's use J1, L60, and SX. Hit your Right Arrow and return to your text. Type Control "B" to position the cursor at the beginning of your text and then type Control "P" to print it. If you have followed the above instructions correctly your disk should now be spinning. Once the flashing cursor has returned and the disk has stopped spinning, type Control "O" to return to DOS READY. The last is very important; do not hit your RESET BUTTON to exit PENCIL. You do not want to re-boot! You must return to DOS to close the file you have opened. This is easily accomplished by typing RESET from DOS READY. Once DOS READY reappears, type LIST FILENAME. Again, FILENAME is the name you gave when you typed `ROUTE *PR TO FILENAME`. There it is. Your text file right justified and with a 60 character line length. If your file started with a \$HEADER you will also have page numbers and if your file was longer than one page, you will see the blank lines separating each page where PENCIL put in a FORM FEED to advance to the next page. Anytime you want a copy of the document you have only to type PRINT FILENAME from DOS READY and it will be printed exactly as if you had run PENCIL and put in all the formatting commands.

Since time sharing systems have various line length limits, I use this technique to format large text files so that I am able to transmit them in the correct format without having to put carriage returns at the end of every line (a very tedious job).

Please note that whenever you use a filename that already exists on the disk, VTOS will not write over the existing file. Instead, it will APPEND the new output to the old file. This can be an advantage and also a disadvantage. If you made a mistake the first time around and are re-doing the experiment, be sure to KILL the file before trying the experiment again.

The SX command is needed to prevent a double line feed which will cause the file to be double spaced.

SCRIPSIT, the new word processor from Radio Shack, cannot be used for this experiment. SCRIPSIT uses its own printer driver and no amount of LINK or ROUTE will change the output from ultimately going to the printer.

NOTICE TO S-8IGHTY READERS

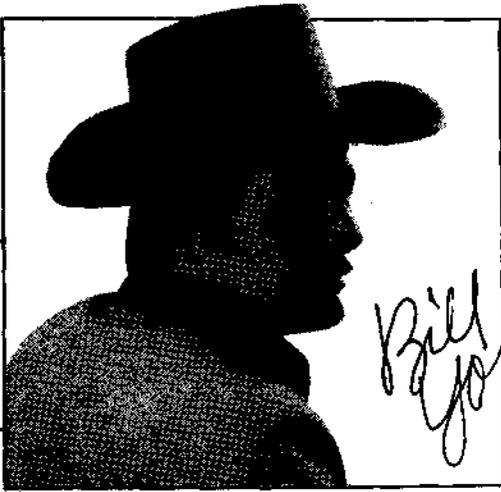
With the May issue, the S-8ighty Magazine will arrive free of charge to readers who subscribe to either SoftSide S-80 Edition, or Prog/80, and recent customers of The Software Exchange. To all others, a \$5.00 subscription fee for 12 issues will be required.

Please fill out the attached and return with your subscription fee to: **P.O. Box 68, Milford, New Hampshire 03055**

Name-----

Address-----

City-----State-----Zip-----



PURE & SIMPLE

I come to you not as a prophet.....

page. If you chat with a good ad salesman he will tell you that in many cases a ¼ page ad will outpull a full page if five categories are created and satisfied:

- 1) **Establish Contact.** Get their attention
- 2) **Create customer awareness.** Solid pertinent information.
- 3) **Arouse interest.** What it can do for me
- 4) **Build customer preference.** Product loyalty based on quality
- 5) **Keep the customers sold.** Good product support and follow-up

Most Important of All!!!!

Check the information on the magazines that you intend to use as advertising vehicles. Check with some of the companies that are marketing products similar to yours and see how they did, or are doing. You'll be very surprised at how cooperative they can be. There are, unfortunately, two or three computer publications that claim circulations far in excess of what they actually are...and charge rates to support the inflated figures. When you know that you are getting X dollars in sales from an expenditure of \$X, then you have achieved the maximum in measuring and evaluating your advertising.

Now that I drive back and forth everyday to my office in Boston, I have become hooked on a morning "BS" show where the announcer is constantly asking trivia questions...after a while you get to be pretty sharp. There's hope for us all yet. Keep smilin'.

Economics??

I have never even attempted to try to understand how the economic situation can be so bad that the prime lending rate soars to 20 plus percent, building starts drop by 60 percent, construction companies lay off hundreds of people at a time...and most computer companies are doing so well that they are having a hard time keeping up with orders. It just doesn't make sense...or does it? We are in a high-technology business where a machine is necessary to remain a competitive force. Without one, you fall behind your rivals and soon find yourself in dire straits. I'm not referring to all businesses, of course...but a business of any size falls easily within these boundaries.

Many companies were very surprised when at a time when tax refunds were coming in...no one was buying anything and some businesses experienced sharp declines in sales. A fast glance into one of the major financial journals provided the answer. They stated that many people had to borrow money to pay their bills as the winter progressed, and they were now paying back those loans with their tax refunds. It seems as though the only major flow of monies is in big business...and the little guy is taking it on the chin, as usual.

How do you compare ad results?

I am continuously amazed at how few companies keep accurate records of the effectiveness of their advertising. Many of the returns are never checked to see if they produced a sale. What the heck are you spending your money for??? Your primary interest should be how much money are you making versus how much you have spent..."Bang for the buck" as a wise man once said. Many people judge their campaigns by the amount of responses they receive...cow chips!...the fact that a label is only costing you \$.75 doesn't mean that you sold any of your product; you can't eat interest or curiosity. Cost per thousand is good sound economic reasoning...but, once again, it only speaks to how you invested your funds...and not what the investment produced for you.

When you receive a telephone order, ask the person how he heard about you. In your follow-up literature, provide a portion of the order sheet where the respondee can advise where he saw your ad. Run coupon type ads, and code the address so you will have a direct measurement of how well your ad is pulling for you. Your ad doesn't have to be a full

Announcing the

NAME-THE-GORILLA CONTEST WINNER

Congratulations to Mr. Alan Simons of Bergenfield, NJ for correctly naming SoftSide Editor/Programmer James Garon as the wearer of the gorilla suit on our April 1980 cover!



IN THE BEGINNING "DOS", AND OTHER NEAT STUFF

by Scott Snyder

In the beginning DOS created the diskette and the disk. And the diskette was without format and void. And darkness was upon the face of the disk. And the spirit of DOS moved upon the face of the keyboards.

And DOS said "Let there be light" and there was H.C. Pennington.

And DOS saw Pennington, that he was good. And DOS divided Pennington from the darkness.

And DOS called the light "TRS-80 Disk and Other Mysteries", and the darkness he called Tandy. And the evening and the morning were the first day.

And DOS said "Let there be a firmament in the midst of the keyboards, and let it divide the keyboards from the keyboards".

And DOS made the firmament, and divided the keyboards which were under the firmament from the keyboards which were above the firmament, and it was so.

And DOS called the firmament "NEWDOS". And the evening and the morning were the second day.

By now you may have guessed that I am a confirmed believer in the almighty DOS. By his will alone, all my free time is spent pondering over the keyboard. My bible (some refer to it as Level II Disk BASIC Reference Manual) is always there by my side. When there seems to be no way out I pick up my bible and read, chapter by chapter for inspiration and guidance.

Yea, though I walk through the valley of the shadow of disk I will fear no evil, for thou art with me, thy ROM and thy RAM, they comfort me.

Although my life with DOS may not be filled with green pastures, and some times I feel that my best friends initials are R.O.M., there is always some light peeking out of the clouds to brighten up a dark

subject. Here then, my fellow keypunchers are some observations that I have found.

These short subroutines have been extracted from my self written payroll program and are used in many of my other programs.

I have a 48K disk system with 2 RS drives, the RS lower case mod, printer and RS232 board. (Just for your information.)

The subroutine at Line 11000 is to clear the screen from any point (ZZ) to the bottom of the screen.

In all of my programs the headings are always displayed (see Line 210).

The subroutine starting at Line 12000 is my combination INKEY\$, flashing cursor, check if numeric input, error trap and see if your shoe is untied subroutine.

We must first look to Line 110 for the set up of the screen display. In Line 110 you will see the prompting message (that can be changed) followed by a STRING\$ print. CHR\$(136) is a graphics character and CHR\$(24) is a back space cursor command. So what we have done is to print our message and set up our field length in one easy step. CL\$ is "clear to the end of the line".

Now the meat!!! In Line 120 you will see three variables.

These variables are: X1, Y1, and Z1.

X1 is the length of the field to be input. NOTE: If the [ENTER] key is to terminate input then the length of X1 must be increased by 1 to provide for the extra character.

(Y1 = 1) if the last character to be input is the [ENTER] key. If input is to be terminated by the last key stroke then set (Y1 = 0)

(Z1 = 1) if the input is to be numeric only. In which case only the characters from zero [0] to nine [9] and the point [.] will be accepted by the system.

The flashing dot routine is in Lines 12030 and 12040. In this loaded routine, the keyboard scan (INKEY\$) is placed inside the "FOR NEXT" loop of the cursor on, cursor off timer. In this way while the computer is counting to 25 (or whatever) the keyboard is being scanned for an input in which case control passes to the next statement. If no input is detected, the counting continues and the cycle is again performed.

Upon return from the subroutine, if XZ = 1, then an error has occurred so go back to the line and input it over again (see Line 130) else continue.

Also upon return, the variable (B\$) contains the values just input, so reassign B\$ to a numeric value if so desired.

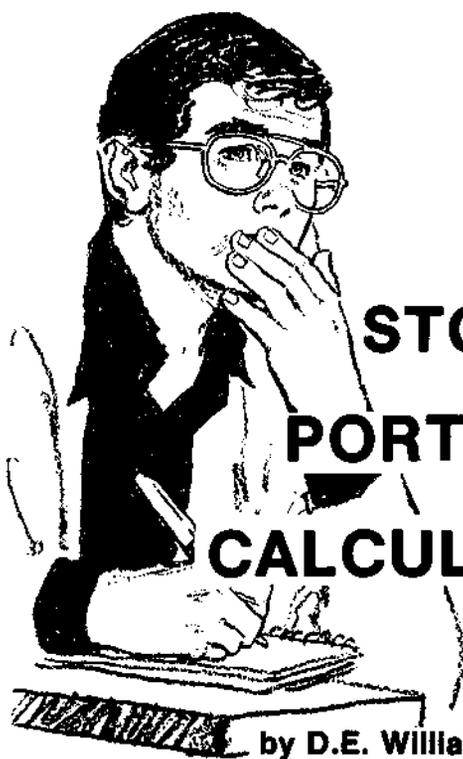
Everybody has their own way to approach a problem; these are some of the problems that I have had to deal with and these are my "solutions". If they can help solve some of yours, feel free to use them.

If you have some other ideas about these or any other little tips, please write them down and send them in. Let others benefit from your SN ERRORS.

By the way—if you're not a disk jockey, delete the POKES in Line 200. They set the date in the interface clock.

So there you have it, everything I know about the TRS-80, and then some, (HA, HA)

HAPPY COMPUTING



STOCK PORTFOLIO CALCULATIONS

by D.E. Williamson

This program is designed for the TRS-80 Level II 16K with, or without, Disk Drive and Printer. To use without Disk or Printer, it should be abbreviated as will be described later. The program is not intended to show you how to make money in the stock market. Should I discover such a program, I will share it with you after I have made my first million. Instead, it is based on the philosophy that by paying attention to a subject you will see problems that might otherwise go unnoticed. The Portfolio calculations can, of course, be done with a pencil and paper given suitable time, determination, and interest.

Note that the basic (fixed) stock data is contained in subroutines. In the illustration, the first subroutine (line 1000) concerns a fictitious stock which is abbreviated "Gennut". The subroutines are intentionally arranged so that all tens digits of the line numbers are alike. This is a convenience in making stock additions, as all you have to do is copy the format of the others, with the appropriate thousands and hundreds digits. The --90 END is considered good practice to keep from accidentally running into the next subroutine.

The subroutines are called in Line 130 in any convenient order (such as the order in which they occur in the stock market reports). If a stock is added or subtracted, Line 100 will also have to be changed to correspond to the number of subroutines called in Line 130. If a stock is deleted, its subroutine need not be deleted unless you need the space.

With TOTALS (Line 570) included, there is room on the screen for only 13 stocks, including the heading and the "prompt". (Another plug for the printer.)

Since there are too many columns to get on the screen at one time, the program prints two "pages". After the first page has filled the screen, the program stops at Line 490. Pressing ENTER prints the second page. The pages can be repeated ad infinitum. With more than 14 stocks, you will have to be quick with the "shift @" to prevent scrolling

20

information off the top.

Now—about the use without a printer and disk drive. Without either, merely delete Lines 600 and 620 to 830 inclusive. In Line 590 delete "OR L(ENTER) TO PRINT". The parts of the program thus deleted will omit PROFIT(LOSS) since data from the previous entry must be stored on disk. If there is a disk drive but no printer, delete only Lines 710 through 800 plus 820 and 830. In Line 590, replace "PRINT" with "FINISH".

Now for the display. The first question asked will be for the date of the closing prices on the exchange. The year must be entered with all four digits. Since the number of days in the calculation is not needed to any great accuracy, it was assumed that one month contained 30.4 days (see Lines 160 and 170).

the program will then display the name of the stock and request its latest quotation, which is entered in the usual way. This will occur for all the stocks listed in the subroutines in Line 130. If you have purchased the same stock at different times and prices, merely omit the line calling for INPUT in the subsequent subroutines. The program will then use the last value of P2 that it has, until there is another input. When all data has been entered, the first page will be displayed. The START DATE in the subroutine may be any date from which you wish to start your comparison of how that stock is doing. If that stock was left to you by your aunt Emma on July 10, 1970, that may not be a good starting time to see how that stock is doing now.

YIELD is the percent yield based on current price (not the BASE price, which is, after all, past history). %GAIN/YR is probably the most controversial quantity, and is also the hardest to explain. Lines 280 and 290 show the percent appreciation of the stock price during the first whole year. After a whole year, the fractions of a year are added (as calculated in Line 220) and are included in the percent appreciation calculation. To find out if the total percent profit is doing better than the yield alone, the two are added together in Line 320. If this number is less than YIELD, that stock should be examined for possible replacement unless it has a satisfactorily high yield.

Page two is obtained by pressing ENTER. No entries need be made. The column BASE is the value of P1 in the subroutine. CURRENT is the value of P2 which was entered as the current price. VALUE is the product of the current price and number of shares (Line 230). \$/SH GAIN is self-explanatory. PROFIT is the profit on that stock to date.

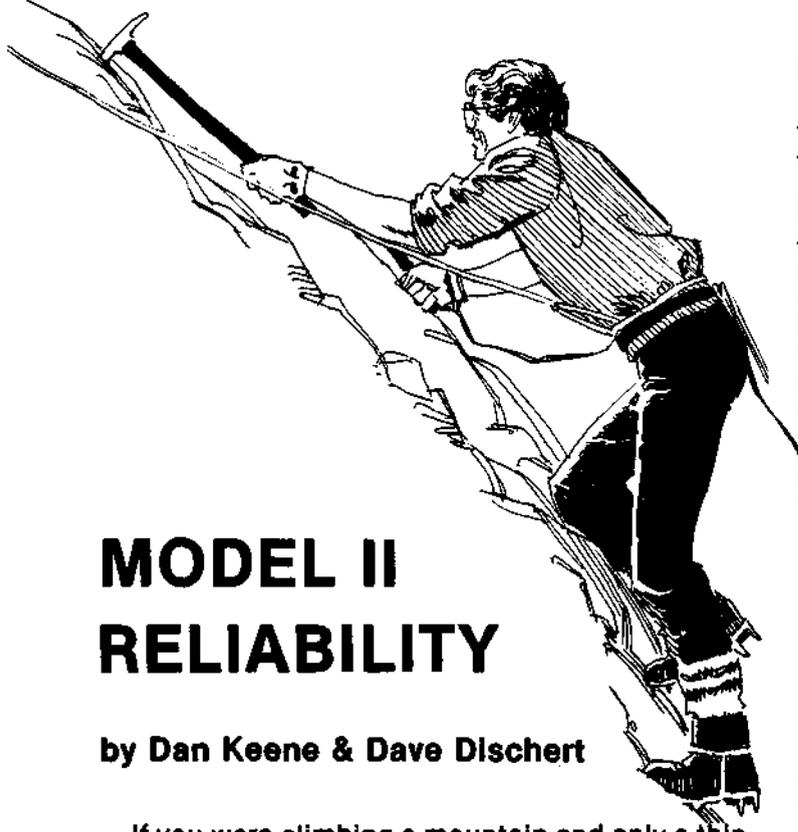
TOTALS shows the value of the portfolio and how much the profit on the portfolio is worth on paper. PROFIT(LOSS) SINCE MM/DD/YYYY is only available by the use of a disk drive which will store up the previous PROFIT. This is subtracted from the current PROFIT to show if there was an increase or decrease since the last entry. Note that "profit" in the program is called PG (PORTFOLIO GAIN(LOSS)).

There is room on each page for more columns, particularly with suitable rounding or truncation. It is left to the ingenuity of the programmer to make such alterations. This author would be appreciative of suggestions.

```

10 REM      STOCK PORTFOLIO CALCULATIONS
20 REM      D. E. WILLIAMSON
30 REM      13001 OLD CUTLER ROAD
40 REM      MIAMI, FLA. 33156
50 REM      NOVEMBER 16, 1979
60 REM
70 CLS
80 REM      ** TO CHANGE PORTFOLIO CHANGE LINES 100 AND 130 **
90 INPUT "CURRENT MONTH, DAY, 4-PLACE YEAR"; M2, D2, Y2
100 N=3: REM      ** N= NUMBER OF SUBROUTINES IN 130 **
110 DIM A$(N), V(N), G(N), P(N), CAP(N), P1(N), P2(N), YD(N), M1(N),
D1(N), Y1(N), NO(N)
120 FOR I=1 TO N
130 ON I GOSUB 1100 , 1000 , 1200
140 REM      ** CALCULATE YEARS OWNED **
150 DS=0: DE=0: WY=0
160 K1=30.4*(M1-1)+D1: REM      **DAY # START**
170 K2=30.4*(M2-1)+D2: REM      **CURRENT DAY # **
180 IF(Y2-Y1)=0 THEN DS=K2-K1: REM      ** DAYS IN SAME YEAR **
190 IF(Y2-Y1)<>0 THEN DS=365-K1: REM      **REMAINDER OF DAYS AFTER K1**
200 IF(Y2-Y1)>0 THEN DE=K2: REM      ** DAYS IN CURRENT YEAR **
210 IF(Y2-Y1)>1 THEN WY=(Y2-Y1-1)*365: REM      ** WHOLE YEARS **
220 Y=(DS+WY+DE)/365: REM      ** FRACTIONS OF YEARS OF ANALYSIS **
230 V=SHARES*P2: REM      ** CURRENT VALUE OF STOCK **
240 PROFIT=P2-P1: REM      ** DOLLAR APPRECIATION PER SHARE **
250 CAP=PROFIT*SHARES: REM      ** PROFIT ON THAT STOCK **
260 PG=PG+CAP: REM      ** PORTFOLIO GAIN (LOSS) **
270 REM      ** % APPRECIATION PER YEAR **
280 IF Y<1 THEN Y=1: REM      ** % BASED ON ONE YEAR UNTIL Y>1 **
290 PCT=((P2-P1)/P2)*100/Y: REM      % GAIN BASED ON CURRENT PRICE **
300 REM      ** % INCREASE PER YEAR + % YIELD **
310 YIELD=INT((100*DIV/P2)*100)/100: REM      ** YIELD % CALC FROM P2 **
320 GAIN=INT((PCT+YIELD)*100)/100: REM      ** INCREASE + YIELD **
330 PRINT"CALCULATION STARTING DATE "; M1; D1; Y1
340 REM      ** CAP IS GAIN OR LOSS ON THAT STOCK **
350 A$(I)=A$: V(I)=V: G(I)=GAIN: P(I)=PROFIT: CAP(I)=CAP
360 P1(I)=P1: P2(I)=P2: YD(I)=YIELD: M1(I)=M1: D1(I)=D1: Y1(I)=Y1
370 NO(I)=SHARES
380 PRINT"STOCK", "VALUE", "%GAIN/YR", "$/SH GAIN"
390 PRINT A$, V, GAIN, PROFIT
400 PRINT
410 T=T+V
420 NEXT I
430 CLS
440 Y$="%      %      #####      ##/##/####      ###.##      ###.##
450 PRINT" STOCK      SHARES      START DATE      YIELD      %GAIN
/YR"
460 FOR I=1 TO N
470 PRINT USING Y$: A$(I), NO(I), M1(I), D1(I), Y1(I), YD(I), G(I)
480 NEXT I
490 INPUT"PRESS ENTER TO CONTINUE"; A$
500 CLS
510 Z$="%      %      ###.###      ###.###      ####.##      ####.##
#####"
520 PRINT" STOCK      BASE      CURRENT      VALUE      $/SH GAIN
PROFIT"
530 FOR I=1 TO N
540 PRINT USING Z$: A$(I), P1(I), P2(I), V(I), P(I), CAP(I)
550 NEXT I
560 Q$="%      %      #####.##      #

```



MODEL II RELIABILITY

by Dan Keene & Dave Dischert

If you were climbing a mountain and only a thin piece of rope existed between you and the ground far below, would you put all your faith in that rope, or would you feel better if there were a backup rope too? If you owned a corporation and 100 million dollars worth of transactions were stored in a computer and there were no other records of these figures anywhere, could you sleep nights knowing that the life and death of the corporation was in the hands of a computer?

The point we're trying to make is that no computer system is perfectly reliable. Anyone who has worked with even a home computer knows that any program he values must be stored in at least two different places to insure that gremlins don't creep in.

How does the business world feel about computer reliability? Judging from the large number of systems used in every area of business, there is a lot of confidence in them. The New Jersey State Board of Pharmacy now approves of patient profile records being stored on computers (as do a few other states). Certain information on the patient is required by law to be kept, such as prescription numbers, pharmacist's initials, drug allergies of the customer and the like. The actual prescriptions written by the doctor must physically remain in files for five years. But they do allow the storage of other information on a computer medium. Many pharmacies, especially chain operations, already have systems in full swing.

TRW electronics has systems in a lot of chain stores where transactions from the floor terminals (electronic cash registers) are recorded simultaneously on both disk and tape. If the disk drive crashes (it happens!), after repairs are made, the disk file can be recreated from the tape. Also, to further insure against losing data, this computer is polled every six seconds by an IBM 370 located downtown in the main office. They trust computers enough to use them extensively but

not enough to eliminate backup systems.

How does the Model II compare? What can be done to insure that data can always be retrieved and is never lost?

The Model II is a good solid system. We have been transporting one all around the state on demonstrations with no trouble. The only hardware problem has been a loose-fitting AC cord. The plug can fall out of the back of the computer if it is bumped, but this is easily remedied by taping it to the cabinet and locating the machine in such a way as to avoid its being touched. Of course, even accidental power down would not cause any data stored on the disk to be lost.

The main unit has a built-in fan and is always cool even after 12 hours of use.

We have not experienced any of the problems that first plagued the Model I such as sporadic rebooting. The Radio Shack fixes, such as buffered cables, adding an extra line between the CPU and the interface, updating the interface printed circuit board, and improving the DOS system software solved that Model I problem.

We have had the Model II lock-up where the only button that will do anything is RESET. Typing SYSTEM"LIST" will bomb, causing an endless display of scrolling data to appear. Of course, this kind of trouble was caused by our failure to give the machine a valid command. Under a normally well-written program, the opportunity to make this kind of error would not exist. Needless to say, the first step toward creating and maintaining a data file is a thoroughly debugged program.

The safest way to store data is to put it on two separate diskettes. There are two ways to do this. Written into the program could be a routine for creating two files, on two separate disks (requiring two drives); everytime data is PUT to the one disk, it could then be PUT to the other. This keeps two data files constantly up-to-date.

Another method would be to make a copy of the data file onto another diskette at the end of the day. However, the COPY utility supplied with Model II DOS does not copy Random Access data files (called Direct Access in the Model II manual). A short program must be written to do this operation. We wrote one as an example and it is shown in Figure One. Only one drive is needed to use it. You must enter the file name and the logical record length. If you don't know the LRL, you can examine the OPEN statement in the program that created it or get a DIRectory and it will be displayed.

Nothing is 100% reliable when it comes to storage. Someone could carry a pile of magnets in his pocket and pass by your diskettes, automatically reducing your mailing list by 1000 names. Storing the diskettes in a metal file cabinet should prevent that from occurring, as will keeping unauthorized persons out of the computer room.

Even the traditionally "safe" medium of storage, paper, could be caught in a fire and destroyed.

The Model II is more than adequate for safe and reliable storage of data. But precautions need to be taken as with any type of filing system for important records.

FIGURE ONE

```
0 CLEAR0:A=MEM-6000:CLEARRA:CLS:DIMDA$(1000)
10
    * COPY ROUTINE FOR DIRECT ACCESS DATA FILES /MODEL II *
20 INPUT"ENTER FILENAME":F$
30 INPUT"ENTER LOGICAL RECORD LENGTH":LR$
40 OPEN"D":L,F$,VAL(LR$)
50 FIELD1,VAL(LR$)AS R$
60 FORX=1TOLOF(1):GETL,X:DA$(X)=R$:NEXT
70 CLOSE:INPUT"INSERT DESTINATION DISKETTE "(X)X$
80 SYSTEM"I":INPUT"HIT ENTER TO CONTINUE ":XX$
90 OPEN"D":L,F$,VAL(LR$)
100 FIELD1,VAL(LR$)AS R$
110 FORY=1TOX:LSETR$=DA$(Y):PUTL,Y:NEXT
120 CLOSE:PRINT"COMPLETED":END
```



THE GREAT BASIC DISASSEMBLER FASTEST FOOT RACE

RULES:

1. The program **MUST** be written in BASIC. No sneaky machine language READ-DATA or fake string PEEKs or POKEs are allowed, whatsoever. Robert M. Richardson will be the referee on questionable rule breaking. Fastest time wins.
2. The foot race is open to any and all comers. Those who wish to have their programs timed may submit the program on disk to Richcraft Engineering Ltd. with a \$10 entry fee to cover prize \$ and return postage. All entries must be postmarked prior to September 30, 1980.
3. Run time measurement may be made on any or all parts in MEM from decimal zero to decimal 12288 using the TRS-80 standard clock. Minimum MEM locations disassembled will be 6000. Every disassembler error will ADD 30 seconds to the timed run. At the very minimum, as many Z-80 object codes as utilized by Level II ROM/RAM and NEWDOS+ must be accommodated. Output may be in decimal or hex and **MUST** include at least: MEM address, 1st byte of object code and Z-80 instruction.
4. The winner and losers will be notified by AIRMAIL not later than October 15, 1980.
5. First prize will be greater than \$100, second prize greater than \$50, and third prize greater than \$25. **All entry fees above actual postage and printing costs will be deposited in the winners' purse and paid out as prize money.** Richcraft Engineering Ltd. is guaranteeing minimum prize money.
6. The winner's name will be announced in 80 Microcomputing magazine, and if the program is published, the writer will be compensated at the publisher's standard rates.

ENTRY BLANK

RICHCRAFT ENGINEERING LTD.
Drawer 1065, #1 Wahmeda Industrial Park
Chautauqua, New York 14722

I wish to shoot crap with destiny. Enclosed is my \$10 entry fee and program on mini-disk (35 or 40 track). I understand that you will return it to me within 15 days after receipt.

NAME _____

STREET _____

CITY _____ STATE _____ ZIP _____

Richcraft agrees to respect the implied copyright of all contestants' contest entries. None will be published without the author's written permission.