

NOTE FROM THE EDITOR

This is the first issue of Compass for INSUA's third year of existence. Because the copy was put together before the annual meeting at the San Francisco Computer Faire on March 19, 1983, INSUA is not able to list the official board of directors for the 1983 INSUA year. This is an excellent opportunity, however, to acknowledge the Board of Directors which served from March 1982 to March 1983:

Clyde Steiner, Chairman
Bob Beaver, Treasurer
Sarah Wasserman, Secretary
George Riddle, Membership
Pavel Breder, Software
Mark Sheppard, Disk Librarian
Ed Coudal, Clubs
Alan Nelson, Compass Editor

As a member of the board, and as Compass editor, I am in an especially good position to appreciate the work and responsibility of each of the Board members. At the risk of embarrassing one of our own, I am particularly pleased to declare the Board's pleasure at the magnificent job which George Riddle and his wife Pat have done on membership, postal box, correspondence, and disk orders. During this year, INSUA has experienced a significant increase in membership. The burden of registering the new members, mailing out back issues of Compass, and passing mail on to other members of the Board, fell directly on George and Pat. This part of INSUA was handled splendidly!

It is particularly important to understand that if any inquiries to INSUA went unanswered, the fault was not with the mail box operation, since letters were distributed efficiently from the mail box to other

members of the Board. The Board has been aware of its shortcomings in dealing with mail inquiries. On the one hand we hope the members of INSUA will appreciate that the members of the Board are all volunteers who give four or five full weeks each out of the fifty-two available in the year just to keep INSUA running, the Compass rolling off the press, the auditors happy, and so forth.

On the other hand, we feel that INSUA should be able to deal with individual inquiries from its members; and we hope that our arrangements with experts such as Ron LaPedis and Bob Cowart, along with the Compass letters to the editor column, will help to satisfy the demand for detailed advice.

As Compass editor I particularly beg the indulgence of anyone who has sent in a note, article, or a program which has not been published. During the year, Compass has included four issues of about forty pages each, plus a Special Issue of twelve full-size, typeset tabloid pages: I hope all members will feel that their \$20.00 membership fee was well spent, judged on the quality of Compass alone.

We are always in need of good copy on topics of current interest, and are particularly on the lookout for material on NorthStar DOS and on the Advantage, as well as on the well-established Horizon. Any material more than two paragraphs in length should be sent on a **single-sided** disk (DOS or CP/M) together with a printout if at all possible.

Thanks for joining INSUA and for reading and supporting Compass!

--Alan H. Nelson, Editor



Here are a few math operators:

COMMAND:	
+	PLUS
-	MINUS
*	TIMES
/	DIVIDE
MOD	LEAVES ONLY REMAINDER
/MOD	DIVIDEND AND REMAINDER
*/	A*B/C USED FOR SCALING
1+	1 PLUS
1-	1 MINUS
2*	2 TIMES
2/	2 DIVIDE

FORTH's text interpreter is very simple. All words and numbers are separated by one or more spaces. Any number of commands may be typed that will fit on a line. After the carriage return is hit FORTH starts processing the line. The first word is processed first. FORTH first searches its dictionary for a match. If a match is found, the word is either compiled or executed depending on whether FORTH is in the compile or execution state. If the word is not found, FORTH will try to convert it to a number. If this fails, an error condition exists. In Marx FORTH the response will be "Say What?". After processing the first word, the rest of the text is processed until it is exhausted. FORTH will then respond with "OK" and expect you to type something else. The input text stream does not have to come from the keyboard. It may also come from a text file on disk, thus allowing extensions and/or control sequences to be created with a word processor and executed later by FORTH.

Logic operators use the data stack just like math operators. Thus when = executes, it takes the top two arguments off the data stack and leaves a 1 if they are equal, and a 0 if not. A 1 is output as a TRUE flag and a 0 as a FALSE flag. These flags are used by FORTH's condition structures such as IF-ELSE-THEN, BEGIN-WHILE-REPEAT, BEGIN-UNTIL etc. In FORTH, IF takes one argument from the data stack. If it's a true flag (non zero), it executes all words between IF and ELSE; otherwise it executes all words between ELSE and THEN. Execution from either path continues after THEN (See EXAMPLE 1).

FORTH CONSTANTS AND VARIABLES

Unlike BASIC, FORTH constants and variables have meaningful names. Thus you can pick a name that describes what is stored in the variable. A constant is created as follows:

```
60 CONSTANT MINUTES/HOUR
```

In this example, CONSTANT creates the word MINUTES/HOUR to behave as a constant and stores 60 into it. When MINUTES/HOUR executes, it leaves a 60 on the data stack.

A variable is created as follows:

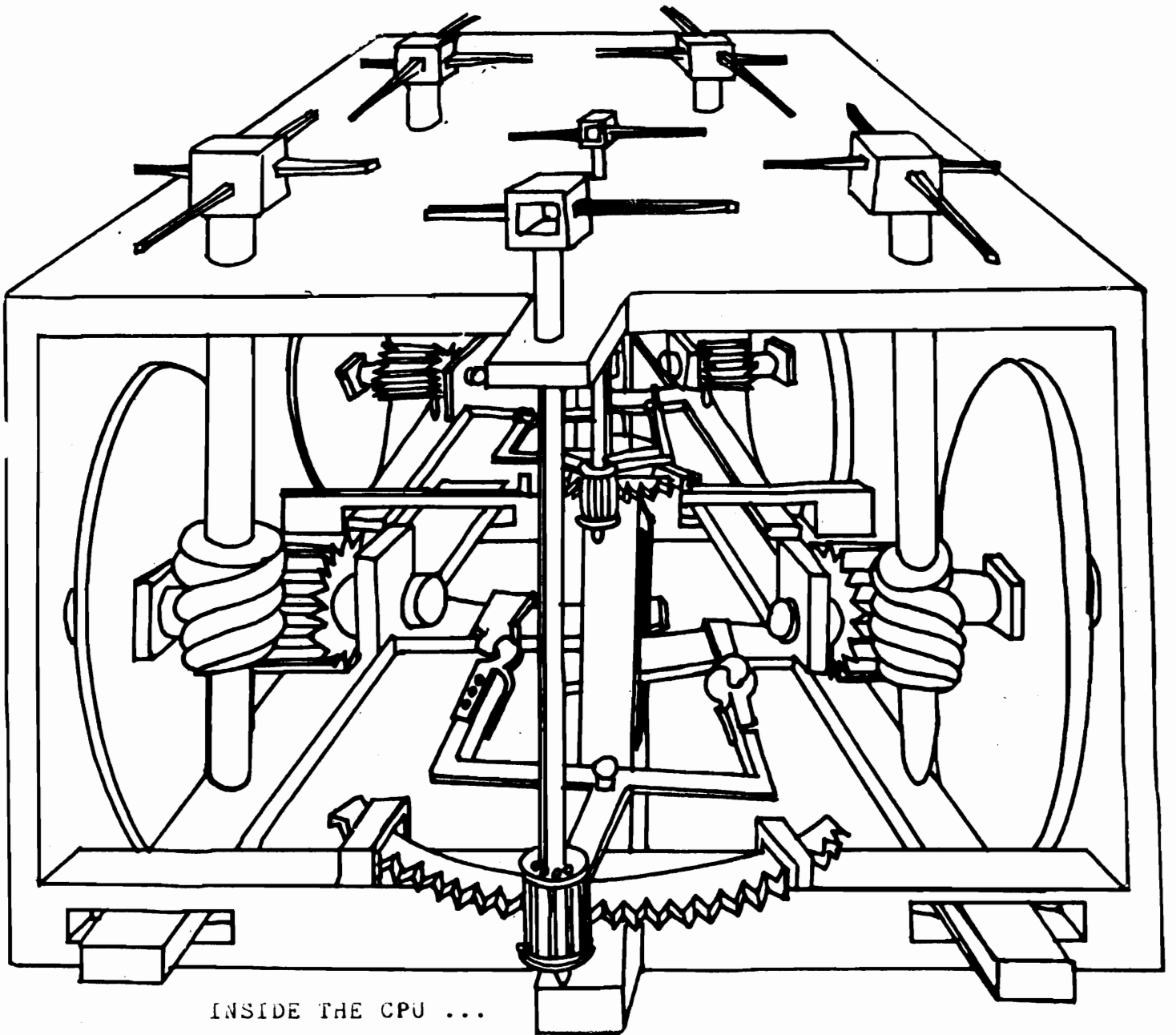
```
VARIABLE HOURS
```

Here, VARIABLE creates HOURS to act as a variable. When HOURS executes, the ADDRESS of HOURS

EXAMPLE 1

```
: PRINT-FLAG ( assumes flag is on stack )
  IF ." TRUE " ELSE ." FALSE " THEN
  ." FLAG" CR ( carriage return ) ;
```

The **COMPASS**



What is Forth? (cont.)

available buffer, returning its buffer address onto to the data stack. If the block was already in memory the memory address is returned as if it were read from the disk. Thus if you're running a disk sort, you will have much more in memory and thus greater speed.

OTHER FORTH FEATURES

Some other nice features of FORTH include not being limited to just decimal input and output. Numeric output can be in any base desired just by loading the system base variable. This comes in handy for base conversion. Print formatting can be controlled to print any format desired with NO limitations. Arrays can be created in any format imaginable. Compilation can be completely controlled to produce any code you want, including code to run on a different processor than in the host system. FORTH is also a STANDARDIZED language which makes source code transportable to other systems regardless of the hardware environment. MULTI-TASKING is common.

FORTH APPLICATIONS

FORTH is used in all applications used by other languages. It is excellent for machine control, to write operating systems, data base systems, and other languages. Assemblers in FORTH are almost considered a standard system tool. Also quite common is to have FORTH written in FORTH so that if the user finds a better way to multiply, all he has to do is change the source code and recompile the system. Word processors are easily written in FORTH. In fact, the first program that I wrote was a screen editor and I wrote it the first

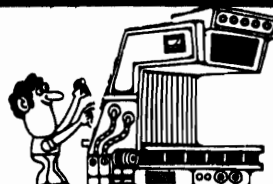
day. FORTH is useful in any application requiring SPEED as it is usually 20-100 times faster than BASIC and uses half to a third as much memory. Applications can be target-compiled to pure object code that no disassembler will touch, with check routines that can't be found so as to protect tampering such as the removal of copyright notices.

FORTH AVAILABILITY

FORTH products are available from different vendors for every computer that exists. Information is available through the FORTH INTEREST GROUP, P.O. BOX 1105, SAN CARLOS, CA. 94070. The group has an annual membership fee of \$15, which gets you 6 issues of FORTH DIMENSIONS MAGAZINE to let you know what's going on in the FORTH world. They also sell for around \$15 assembler listings for FIG FORTH for any processor you specify. That's how I got started.

If you already have a FORTH system and you would like a better one, PERKEL SOFTWARE SYSTEMS, 1452 NORTH CLAY, SPRINGFIELD, MO 65802 (That's me!) offers Marx FORTH source code listing written in FORTH to compile a Marx FORTH system from your existing system. This sells for \$30 on paper and I must warn that it requires quite a lot of familiarity with FORTH to get going. Or I should say that by the time you adapt it to your system, you will know FORTH inside out. This is an excellent way to improve your programming skills.

As far as complete FORTH systems go, I only sell for the NORTHSTAR, RADIO SHACK, and ATARI systems. (IBM-PC and APPLE soon to come.) There are plenty of FORTH vendors for whatever system you own.



WHAT IS FORTH?

By Marc Perkel
Perkel Software Systems

FORTH is a relatively new computer language whose popularity is increasing rapidly. Its attractiveness is that it gives the programmer absolute control over the machine, like an assembler, while maintaining an interactive environment superior to BASIC. Different programmers have different tastes in languages just as different people have different tastes in cars. For illustration, BASIC can be compared to a luxury RV, with automatic transmission, power steering, cruise control, big, slow, and clumsy. FORTH, on the other hand, would compare to a racecar; fast, small, maneuverable, but you shift the gears, you step on the gas, you control all, and you know what's under the hood.

Upon seeing FORTH source code, the first thing most people notice is that there are no LINE NUMBERS, no GOTO'S and no GOSUB'S. This is due to the structured nature of FORTH. Unlike BASIC, FORTH does not have a fixed vocabulary. In fact, programming is accomplished by defining new words out of existing words. The new word may be executed, or it may be used in the definition of another word and so on. Thus you custom-build your FORTH for each application, choosing definition names that describe the operation being done. A program is executed by executing your main word which in turn executes all its support words. More on this later.

The next most outstanding aspect of FORTH is its mathematics and the way it passes arguments from one word to another. FORTH uses 16 and 32 bit INTEGER math. Integer math has the advantage of speed, simplicity, and accuracy. Of course, since

FORTH is an extensible language, FLOATING POINT can be loaded into the system, but once you get used to it, floating point is rarely needed.

Unlike most traditional languages, FORTH uses POSTFIX or REVERSE-POLISH NOTATION. This is like most Hewlett-Packard calculators. Again this is chosen for speed and simplicity. The use of RPN eliminates the need for parentheses and operator precedence found in most traditional languages.

TRADITIONAL INFIX NOTATION:

$(6*7)+(2*11)/16$

POSTFIX NOTATION:

6 7 * 2 11 * + 16 /

FORTH systems use a data stack to pass arguments between operators. If a 6 is typed, a 6 goes on the stack. If a 4 is typed next, a 4 goes on the stack, leaving the stack containing 6 4. If + is then typed, + takes the 4 and the 6 off the stack, adds them, and pushes a 10 on the stack. If 2* is then executed, the 10 is removed and a 20 put in its place. Generally all words get their arguments from the data stack and leave their results on the data stack.

FORTH includes special stack manipulating commands to arrange things on the stack. Here are a few.

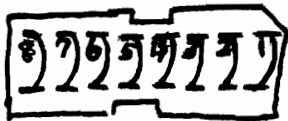
COMMAND:	BEFORE:	AFTER:
DUP	1 2 3 4	1 2 3 4 4
SWAP	1 2 3 4	1 2 4 3
OVER	1 2 3 4	1 2 3 4 3
ROT	1 2 3 4	1 3 4 2
DROP	1 2 3 4	1 2 3
2DROP	1 2 3 4	1 2
2DUP	1 2 3 4	1 2 3 4 3 4

correctly--for 300 or 1200, depending on your modem.

If this still does not work, you may need a more sophisticated connection between your computer and your modem. The Advantage manual does not go into the modification which you need to make to your SIO card in order to run a modem. (This type of modification may also apply to the Horizon.)

The RS-232 Communication Standard describes the interface between Data Terminal Equipment (terminal) and Data Communication Equipment (modem). If the computer is communicating with a serial terminal (such as a CRT, teletype, or a hard copy printer), then its serial interface must be configured to play the role of a modem. If the computer is communicating with a modem, then it must play the role of a terminal. Computer-to-computer communication is possible if one computer plays the role of a terminal, and the other the computer. In order to accomplish this, two things are necessary. First, you have to wire up a "header", which is a little connector whose appearance resembles that of a chip. The wiring should be as explained in Fig. 1.

Before:



Location 3A

after:



Note: Be sure to remove any header at location 1A!!

Fig. 1

You must make sure that you have given Crosstalk the correct command at the time of making the initial connection. Modem communications are possible by virtue of using several different audio tones, each one having its own frequency. One set of pitches is assigned to the computer making the original connection, and the other set to the the receiving, or answering computer. Proxima is in the position of answering your call, so you must use the Crosstalk command for putting your Advantage in the ORIGINATE MODE. Most modems or modem programs have some means of doing this.

Finally, try using a 25-conductor cable between your modem and the computer. Not all of these lines are actually used, but a number of them are, and it may be easier to make sure all pins are connected than to worry which ones should be. And next time your printer doesn't work, remember that you changed that little header on your SIO card. You will need to change it back before using the printer. [If you have to do this often, try wiring in a switch--see the Compass articles mentioned above. --Ed.]

2) I am a little confused by your reference to Font programs, since high resolution hardcopy is contingent upon possession of a high resolution printer or plotter. However, assuming you currently own an Epson MX-80 or 100 with Graftrax, or an Okidata, or another such printer, you may be in luck or at least something close to it. I say this for two reasons.

First, North Star has announced that it will be offering N* BASIC / GBASIC to run under the CP/M / GCP/M operating systems. With a little work in GBASIC you could execute a program to do your fonts for you. But I expect that a more immediately applicable approach will be afforded via the use of another soon-to-be-released pro-

What is Forth? (cont.)

is left on the stack. To read the variable, the word @ (fetch) must be executed. @ removes the address on the stack and replaces it with its 16 bit contents. Conversely, the word ! (store) takes an address and a value from the stack and stores the value into the address. See EXAMPLE 2.

As you can see by now, once you get used to it, FORTH code describes what is happening better than BASIC. This is important because you don't need a bunch of REMARK lines to remind you of what you were doing. Which of the two statements in EXAMPLE 3 is more descriptive?

FORTH COLON DEFINITIONS


Now we get to the real power of FORTH. As you can see from the examples I've already used, a colon definition begins with a : (colon), and ends with a ; (semicolon). : is a FORTH word that creates a dictionary entry for the name after it, just as CONSTANT and VARIABLE do. It also sets the compile state. This signals the text interpreter that the words following will be compiled instead of executed. Compilation in FORTH is simple. After a word is found, its address is added to a

chain of addresses that is built after the word being defined. Thus, in general, every word used inside of a colon definition compiles only TWO BYTES, the address of where the code is. This makes for extremely compact code. This technique is called threaded code. All addresses point to other addresses which eventually point back to executable code somewhere. At this point it will be interesting for the reader to note that most of FORTH itself is high level code and most FORTH systems have only about 500 to 700 bytes of actual running code, with the entire language only 7K long.

FORTH DISK I/O


FORTH supports virtual memory disk I/O. This means that any number of disk buffers is supported, not just one like in most BASIC systems. Marx FORTH also allows dynamic disk buffer allocation and deallocation. This is a feature not supported by any other vendor. FORTH looks at the disk as 1024 byte blocks regardless of the actual sector size. Disk I/O is accomplished by the word BLOCK. BLOCK expects a block number to be on the data stack. It takes that number and brings that block from the disk to the next

EXAMPLE 2. --- ---



```
60 CONSTANT MINUTES/HOUR    VARIABLE HOURS    VARIABLE MINUTES
: CALCULATE-MINUTES
  HOURS @ MINUTES/HOUR * MINUTES ! ;
: PRINT-MINUTES CALCULATE-MINUTES
  MINUTES @ . ( . means print ) ;
```

EXAMPLE 3. --- ---



```
6730 GOSUB 7530\GOSUB 2450\IF X=40 OR Y<>5 THEN 940 ELSE 390
: START-CAR BEGIN CRANK-ENGINE ?FIREING UNTIL
  OIL-LIGHT IF ADD-OIL THEN WARM-UP ;
```

North Star Answers (cont.)

called EXT/ACK protocol to run faster. Some programs can handle this for you, but as I mentioned two issues ago, the best arrangement is to use "hardware handshaking" instead of a software arrangement. This will allow higher speed printing with all programs in both DOS and CP/M. [See "Hints for Installing WordStar" in Compass, Vol. II, No. 3 --Ed.]

Just when it sounds as though you are stuck, I happen to have lucked across solution for all you 1620 (and 1610 by the way) owners who have been suffering along at 300 baud, or trying to write your own EXT/ACK drivers. The miracle of the micro-electronics cottage industry comes to your rescue.

A newly announced product, by The Printer Works in Hayward, Ca., sounds as though it does it all. This is a single printed-circuit board which:

1. Allows most Diablos to use
 - a. EXT/ACK
 - b. Xon / Xoff
 - c. or Hardware Handshaking
2. Has a 64 K buffer on it to free up your computer while you are printing a document.
3. Has a Centronix parallel and RS-232 port on it
4. Has a Z-80 on board.
5. Will run as high as 19.2 Kilobaud.
6. Lists at \$379.00

I encourage anyone who is having troubles with one of these printers to investigate. The number is: (415) 887-6116.

Q: #3

How can I get MY question in print (and possibly even answered) in the COMPASS? -- Curious, CA.

Dear Curious:

Send a copy of your question to the Editor of Compass c/o the current P.O. Box address of INSUA, and prepare to wait. These things take time, but we are doing our best. Our apologies to those of you who are STILL waiting. -R.C.



PERSONAL PEARL

Personal Pearl is a database management program designed expressly for microcomputer users. Personal Pearl (henceforth PP) presents itself as a system so user-friendly that its internal workings and files are entirely hidden from view. All the user sees is highly formatted screens of forms and menus.

The emphasis on external appearances is evident throughout: PP is beautifully packaged, with a tutorial for beginners and another for more advanced users, plus a handsome loose-leaf reference manual. All these manuals are actually photo-typeset, and not merely printed on the ubiquitous daisy-wheel printer (like the present issue of Compass).

Is PP merely Madison Avenue hype? or does it offer something for the garden-variety micro user who knows a thing or two, and is unlikely to be impressed by slickness of packaging?

I will try to answer this question at the end of this review. First, however, some comments on using PP.

PP comes with six disks. Thinking I understood how to go about trying any program for the first time, I copied all six

NORTH STAR ANSWERS

By Bob Cowart
NorthStar Technical Consultant

Welcome again to the Answer-Man column. Now that this is becoming an established feature of Compass, I am faced with the consideration of what to call it. Bytes from Bob? Robert's Ruminations? Maybe next time you will have to hunt for a different title to find the proper column.

Letter #1.

Q: In reference to the article in Compass, Volume II, no. 3, on Proxima, the North Star Computers Electronic Bulletin board, I set up my computer using Crosstalk, made the phone call to (415) 357-1130, and attempted to open up a dialog. This was done on various days, but in every case I have been cut off before a communications link could be established. Something seems to be missing from the write-up.

On other matters, I use an Advantage for editing mainly, and would like to set up a variety of headline FONTS in some of my articles. Font programs exist for some of the more popular computers but I have not been able to locate any that will produce large character fonts within wordstar. The option of preparing a document with high resolution graphics within the Wordstar program certainly would be of considerable advantage in the preparation of an engineering report. If someone can do or has done pixel graphics in MBASIC or CBASIC, then it may be possible to inject this into a Wordstar program. However, the N* GBASIC graphics cannot be accessed in CP/M (at least I have not been able to do this).

There are a number of programs and data files available on the IBM 8" format that I would like to be able to use. However, I have not been successful in locating the 8 inch system which will do this for me. -- P.W. Manchester, MO.

Dear P.W.:

1) Regarding the use of Proxima: for those of you who do not know, there is a 24-hour N* Bulletin Board called Proxima which offers lots of information on NorthStar topics, as well as providing a forum for discussion and, sometimes, answers to your technical questions. All you really need is a modem program designed to run with the computer you are using, a phone, and a modem (you can pay Ma and Pa Bell later). My friends and I check in with Proxima often, and it always works, so the problem must be in your hardware or software.

The Crosstalk modem program comes in several configurations, one of which is for the NorthStar computers. This version covers both the Advantage and the Horizon. Included are the assembly language files you need to modify Crosstalk for various hardware configurations you might be using (an HSI0-4 serial board in the Horizon, or putting your SIO in slot 2 of the Advantage for example). There is even a section in the manual covering the procedure.

Connecting to a modem may be a simple job: make up a three-wire cable to run between your RS-232 printer port and your modem: connect the two pin 7's together, connect pin 3 at one end to pin 2 at the other, and pin 2 at one end to pin 3 at the other (see the "Easy Switch" articles in Compass, Vol. II, no. 1). Set your baud rate

Personal Pearl (cont.)

That reduced the number of user-supplied disks to a mere eleven. O.K., insert the PP welcome disk, and try the P routine once again.

Instruccions tell you when to insert each of the six system disks in drive A, and each of the ten blank disks in drive B. Quite straightforward, except that copying and verifying all the files took almost two hours, with coffee breaks while the disk drives did their thing. I didn't try timing the entire process with a stop-watch, but I guess that it would take at least an hour with no time lost.

With my set of ten labelled disks, I thought I would give the system a try. I turned to the manual and read through to p. 13, where I was instructed: "Place your DESIGN FORMS Disk 1 in drive A," and "Place your PROGRAM AND DATA Disk in drive B." I found the DESIGN FORMS disk, but none of the ten was called PROGRAM AND DATA. I took a guess and decided (correctly enough, as it turned out) that #3, called START PEOPLE, would serve. I then called up the program by typing PEARLDF, but received a charmingly erroneous error message: INCOMATIBLE VERSION.

On going through the "Welcome" routine again, I discovered that I had missed the step of actually personalising PP. Essentially this means that the user's name or code is inscribed onto each of the four program stages, and ends up on each of the data files. This insures that files and forms are not confused among users.

This time typing PEARLDF brought up the program nicely--I had finally become a successful user of Personal Pearl!

Assessment

If I have given the impression that the user-friendliness of PP is purchased at a consid-

erable overhead of time, disk purchases, disk swapping during run time, and general bother, that is exactly the impression I meant to convey. Is PP worth the effort and the expense?

The answer to this question probably depends on the conditions of use; it also depends on the answer to the question, "Compared to what?"

The answer to the second question is quite easy: compared to dBASEII, perhaps the most widely used of all database programs.

Using dBASEII requires some understanding of databases. Giving dBASEII commands, and especially writing command files, is very much like programming. The average hobbyist may feel right at home with such programming, and may feel comfortable working directly with the database, using the edit or the browse function.

In many applications, however, dBASEII systems are tailored for use by persons who are not expected to understand databases. A programmer may install menu-driven routines for data entry and report generation. In the last analysis, dBASEII programs may end up working much like PP programs, since after all the stages of database routines are inherent in the idea of databases, and are not the invention of a particular vendor.

In short, PP may be thought of as a pre-programmed database package.

Is Personal Pearl a good program to purchase for your micro system? Should you choose it over dBASEII? In general, my advice would be as follows:

- 1) If you are a relatively unsophisticated microcomputer user, and if you like the idea of menu-driven procedures, and if you have a friend or a dealer who is willing to lead you once or twice through the paces, give PP a try: if it suits your needs, go with it.

duct called IMAGEMAKER. This is one of several products which N* will be releasing this Spring, all aimed at allowing complex and sophisticated graphics images to be created, printed, and interfaced with existing CP/M programs. A graphics library will also be available to assist in the creation of more intricate images. Assuming you have an Epson or Okidata graphics printer, or a Hewlett-Packard, Houston, or Nicloet plotter I suggest you wait to see what N* has to offer sometime this Spring. From what I have seen so far, it will probably do what you need. Whether interaction with Wordstar is planned is another story.

[Using Epson's expanded or compressed font mode from WordStar is a simple matter--see Jim Woolley's "WordStar with Epson" in Compass, Vol. II, no. 3. You should also investigate a software program called "Fancy Font," which will generate different fonts of various sizes on an Epson printer. Call or write Soft Craft, 8726 S. Sepulveda Blvd., Suite 1641, Los Angeles, CA 90045 (213) 641-3822. --Ed.]

While on the issue of graphics, I want to pass on a tip for anyone interested in scientific/technical/ or Greek word processing. The WPS Computer Mart, 1395 Main St. Waltham, Mass. 02154 appears to have a very sophisticated solution for you. The phone is 617-899-4540. The programs seem to be designed with the Advantage in mind.

As for your last question, rather than repeat myself, please see my column in Compass, Vol. II, no. 4, where I addressed the 8" to 5" conversion issue.

Letter #2.

Q: I am in the process of putting together a North Star Horizon II. After having been a user for about two years, I am a

bit out of touch with things and would like to hear from someone who is a bit more up to date. If you think that North Star is on the decline, and that I might be wise to go with some other system please tell me so, and which system you think might be better. I will be using the system for business, that is word processing, and accounting and inventory.

Also, I will be using a Diablo 1620 printer and would like to know if any special changes in my software are needed to run it and if you have the software to run it of that change is needed.

D.R.C. Knoxville, TN.

Dear DRC:

If it is any consolation, I am putting together a Horizon for myself--not that I am the epitome of the neoteric digit head as evidenced by my writing this on my IMSAI--and consider it to be a hale and hearty product. I can assure you, though not in detail at this time, that North Star does have intentions of continuing development of the Horizon in the very near future. In general terms this means additional software and hardware. I personally feel that the Horizon is a very well built, time-tested and durable product, and that North Star as a manufacturer, though aware of the stiff competition in the marketplace, has a viable gameplan which appears to be working. Considering that N* is one of the six first microcomputer manufacturers, and has survived the demise of many, I wouldn't worry about the company folding. As for software, there are few micro manufacturers who offer as extensive a line of business software as N*. (Excuse my soapbox.)

As for the Diablo, special software is necessary if you want to run above 300 baud, since the printer needs what is

Personal Pearl (cont.)

manual is irrelevant and even confusing for me.

Cost

Personal Pearl retails for \$295.00, less than half the list price of dBASEII. Of course the latter can be purchased at a considerable discount through mail order or bulk sales firms, but then it may be possible to find a deal on PP as well.

Summary

I personally would prefer dBASEII to PP, but I have a bias for programs which allow vir-

tually direct access to files and compatibility among computer programs; I also feel that PP's user-friendly approach sometimes serves more as a barrier than an inviting open door.

Since so many subjective considerations must bear on the evaluation of such a complex program, however, I conclude by inviting INSUA members who have used Personal Pearl in day-to-day data work or over a longer period of time to supplement or challenge my observations and judgments.

--Reviewed by Alan H. Nelson

RAM TEST FROM CP/M

By John A. Evans
Tempe, Arizona

CP/M has been my mainstay for the last 2 years since I overcame my fear of it and decided that it should be my main operating system for my Horizon over North Star's DOS. However, the DOS was nice to have around for the good utilities that were available. Whenever I wanted to do a complete memory test, I would just pop in the DOS disk and type GO RMTST3 or GO RMTST5.

Well, I am a bit lazy so I decided to let the computer do the work for me. Besides, if I could exercise my brain, I just might learn something about my system and programming in general. What follows is a complete implementation of the RMTST3 and RMTST5 programs under CP/M.

First off, the memory test programs as supplied by North Star on the release 5.2 disk are specifically designed for the Horizon only, with a terminal on the standard serial port. Two phases of the test require a double-density controller but

the program will run without one. The code is also written for a Z80 processor. That should signal to some of you to move on to another article and inspire others to dig a bit deeper. All I/O is built in, which makes it perfect for stand-alone operation under CP/M. The setup description that follows assumes that you have the required hardware configuration and that you have DOS, RMTST3, RMTST5, and CP/M system software. No attempt will be made to reveal the secrets of proprietary software.

RMTST3 and RMTST5, while running, "live" at 3000 Hex and 5000 Hex, respectively. All that is needed is a small routine written at 100 Hex to move these programs to their proper locations and then jump to them (sort of like diving to a life preserver you've just thrown into the water). Our CP/M COM file must contain the original program immediately after the relocation routine.

Figure 1 lists an assembly language relocation routine I

Personal Pearl (cont.)

disks to freshly formatted blank disks (with CP/M installed). I then placed the "Welcome" disk in drive 1, booted up, and typed WELCOME.

After a terminal installation routine and a neat bit of graphics, a menu offered to take me through a tutorial introduction to PP. I typed D, and in the course of the tutorial was introduced to the four-stage concept of database management:

1. Design form
2. Design report
3. Enter data
4. Produce report

These four stages are kept quite distinct from one another in PP. Each stage is broken down into logical sub-procedures, all menu-driven.

1) To design a form, you first "paint" the screen more or less as the form will ultimately look, assigning names to the fields ("Name", "Address", etc.), then marking the fields themselves, either underlined (for characters), designated by 9's (for formatted numbers), or defined as mm/dd/yy (for dates).

The next step in form design is to walk through the form once again, this time defining the fields more precisely, specifying, for example, whether the field is a character, a number, or a date, whether it must always be filled in at data-entry time, and whether it is required to be unique among all equivalent fields in the data base.

Optional extra steps are to assert indexing strategies, or to indicate whether a field is to be filled in with information taken from another form.

2) Similar steps are required in designing reports, which may virtually duplicate the forms, or may take on an entirely different appearance, such as column list-out.

3) Data entry is quite straight-forward, since the form has been designed with such specificity. A function to reproduce the information from the previous record for any given field is a particular time-saver.

4) Producing the report is also quite simple, since so many particulars are already specified at the time of designing the report. The principal options are to have the report sent to the screen, to the printer, or to a disk file.

PP not only takes you through all these steps in the screen-oriented tutorial, but requires the same steps in virtually every subsequent application, whether developing routines (using stages 1 and 2), or entering data and generating reports (using stages 3 and 4).

PP provides a library of three pre-designed routines, an address-book, an appointment calendar, and a cash-management routine.

Once through the tutorial, I decided to install PP and try some forms of my own. Remember that I had already copied all six PP disks onto blank CP/M disks. Now when I followed the install routine (typing P for Personalize), I was told to prepare ten blank disks which I should get from my "supplies" (a typo for "supplier," I suspect), and to put labels on them. All ten were required for my standard 5 1/4" SSDD system. DSDD systems, 8" systems, and hard disk systems required fewer disks.

But wait a minute: I had already used six disks, and was now asked to prepare ten more. **sixteen disks in all!!!!???** I reread the manual. Perhaps I was wrong to have copied all six disks from the outset. Best now to erase the files from all but the first of my own disks, and begin over (after write-protecting the PP-supplied disks).

```

; BOOT FIRST UNDER NORTHSTAR DOS
+GO M0000
MONITOR 5.0
>FM 3000-3500 20 ; CLEAR OUT RAM SO THE END CAN CLEARLY BE FOUND
>OS ; AND VERIFIED
+LF RMTST3 3000 ; LOAD TEST PROGRAM INTO MEMORY
+JP0 ; BACK TO MONITOR
MONITOR 5.0
>DH 33F0-3410 ; SEARCH FOR BOUNDARY BETWEEN CODE & 20H
33F0 XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX XX ; PROPRIETARY CODE
3400 20 20 20 20 20 20 20 20 20 20 20 20 20 20 20
3410 20

; REBOOT NOW UNDER CP/M
C>DDT MOVRMTST.HEX ; LOADING HEX FILE PREVIOUSLY
DDT VERS 2.2 ; ASSEMBLED

NEXT PC
0116 0000
-M3000,3400,116 ; MOVE THE TEST PROGRAM TO END OF RELOCATION CODE
-G0 ; WARM BOOT ( CONTROL-C WILL DO )

C>SAVE 5 RMTST3.COM ; SAVE IT IN A .COM FILE
C>RMTST3 ; CHECK IT OUT

```

===== figure 2 - sample terminal session required to
append program to relocation code and
save as a .COM file under CP/M

COMMENTARY

By Burt Andrews

[Once again, Compass offers an opportunity for a member to comment at length on INSUA and on matters raised in recent issues of Compass. --Ed.]

INSUA:

1) More articles in Compass for less experienced members? PLEASE--be awfully careful about this. I speak as one of those 'less expert' people but I don't want to stay that way. There are plenty of publications out there aimed at folks who just want to use the box without any more than a necessary knowledge

of its insides. (That's not meant as a put-down. It's the huge mass market that makes it possible to have systems like this at all).

But what has made Compass unique and valuable is the technical insight it has provided about the North Star hardware and software. Articles like Maguire's (Vol. I, no. 1), later discussions on the problems of parallel port interfacing to printers, the piece on performing disk operations through ports, and others too numerous to detail, have all spurred me to dig deeper, learn more, and to try things I would not have even known how to start on be-

Personal Pearl (cont.)

2) If you are a more sophisticated micro user and you are looking for a database program for yourself or for a member of your family or for your office, and if you want to avoid the bother of programming input and reporting routines, give PP a try at your dealer's: if you like what it does in any given application, go with it.

3) If you are the kind of micro-user who likes to do have direct access to his data files, and likes to do his own programming, or if you like the feeling of knowing exactly what's going on in your software programs, stick with something like dBASEIII.

4) If you are providing software for the use of relatively untrained data-entry operators, but you like the idea of designing your own menu-driven programs from start to finish, stick with something like dBASEIII.

5) If you are in sole control of your micro system, you may feel that the personalizations and security features of PP are a pain in the neck; but if your system is shared among users who could conceivably interchange disks and files to disastrous effect, you may prefer PP, with its file-protection features.

6) If PP offers advanced functions which uniquely meet your needs, buy it; if dBASEIII has other functions which are more useful to you, stick with that program.

Advanced Functions

Each program excels in certain advanced functions: PP has a specialized date function; allows for easy transfer of single records from an indexed file to another file; provides for more direct sorting

on primary and secondary fields; generates files which can be incorporated directly by SuperCalc.

dBASEIII affords more immediate access to datafiles with the "browse" function; allows for large-scale changes to fields with the REPLACE command; handles boolean operations more directly; allows for changes at any time in the type of any field (PP does not); allows for individual tailoring and virtual automation via programming facilities; and can incorporate as well as generate ascii files.

ASCII and Back

For me, the possibility of two-way transfer of files between dBASEIII and CP/M word-processing programs would make dBASEIII my first choice even if PP won out on every other point of comparison. The current PP can generate files which can be edited, but cannot assimilate an ascii file. I asked the Pearl people about this and was told that an upcoming release will indeed be able to assimilate ascii files. My advice would be to wait for this new release, or to have your dealer guarantee that you can get an update at a nominal cost.

North Star Notes

Advantage users will be glad to discover that their machine is one of the built-in options in the terminal-installation routine. Some Horizon owners may be disappointed to learn that 56K of available memory is not quite enough for full use of all PP functions. The personalization program informed me that my 56K system had only 49.1K Transient Program Area available, whereas PP wished for 51K; hence the maximum size of my forms would be somewhat limited. My ADM3a terminal also lacks highlighting, so that certain information given in the



or a CTL-P printer toggle, or an automatic left-margin set (0-9) on the printer, or a function key to reset the parallel port printer ready line, etc. etc. (I haven't got Double-Density yet, and if I find that I can't get all this stuff into a new and longer DOS with DD I may stick with what I've got until I figure out how to do it.)

5) Re the last sentence on p. 43 of Compass, Vol. II, no. 3: just change the byte at 570B of

the 5700 monitor to the high byte of the origin of your DOS and OS should work ok.

This got longer than I expected, but I guess that if you didn't want comments you wouldn't ask for them. Let me close by expressing again my appreciation for the greatly increased pleasure I've been able to get from my N* just because of knowledge I've gained from INSUA-Compass, and the members who've contributed to it.

PASCAL NOTES

Delivery of JRT Pascal Disk #1012 has been delayed because JRT was dissatisfied with earlier releases, and had promised a release with known bugs fixed. The disk should be available by the time this Compass is printed; however, further delays are possible, and INSUA begs its members' patience. The following comments may apply only to earlier releases.

INSUA:

I was very happy to see your column in Compass, Vol. II, no. 3., as I have a copy of JRT Pascal 2.1 procured from JRT in August of this year. As I am just leaving BASIC and learning to "swim" in Pascal, I regret that at this time I am unable to

contribute any programs to your review. My only problem was that initially I could not load EXEC but could load the Compiler JRTPAS2. Finally I read the manual (like I originally should have done) and it cautioned that PASCAL.LIB should be available on Drive A or B when using EXEC or the Compiler. My Pascal came on two discs from JRT with only PASCAL.LIB on the disc with the Compiler: as soon as EXEC had PASCAL.LIB available it loaded fine. I presently have JRT PASCAL 3.0 on order--

Sincerely,

A. J. Horner
Van Nuys, CA

RMTST from CP/M (cont.)

found in the BYE69.ASM file on one of the many remote CP/M bulletin boards. The value for the end of moved code and the JP address must be changed to convert the RMTST5 program. These are noted in the figure. Assemble this program using CP/M's ASM utility. The PRN listing generated should agree with figure 1.

Figure 2 shows how to load the programs to memory under DOS, load the HEX file into memory at 100hex under CP/M, and then move the program down to the proper location (116hex) before saving as a COM file.

Filling memory with spaces (20h) and then loading the

RMTST3 file permit you to view the end of the file and verify that your memory test program is the same size as mine.

The tests run just like before. A Ctrl-C jumps to E800 to reboot your CP/M (or whatever you have in drive A). Typing any other character during memory testing will reprint a memory map with any memory errors flagged. Happy RAM testing!!!

(This article is dedicated to all those responsible for keeping the public domain spirit alive!!!)

P.S. This program will work for themonitor program M5700also!!

```

-----
;
;      RELOCATION CODE OBTAINED FROM THE BYE69.ASM PUBLIC
;      DOMAIN UTILITY ORIGINALLY BY DAVE JAFFE WITH
;      RELOCATION CODE CREDITED TO BILL PRECHT
;
0100      ;      ORG      100H
;
;Move memory test program code up to high RAM and jump to it
;
0100 010004      LXI      B,400h      ;NUMBER OF BYTES TO MOVE
0103 210034      LXI      H,3400h    ;END OF MOVED CODE :5400h for ramtest5
0106 111505      LXI      D,515h    ;END OF SOURCE CODE
;
0109 1A          MVLPLDAX D      ;GET BYTE
010A 2B          DCX  H      ;BUMP POINTERS
010B 77          MOV   M,A    ;NEW HOME
010C 1B          DCX  D
010D 0B          DCX  B      ;BUMP BYTE COUNT
010E 78          MOV   A,B    ;CHECK IF ZERO
010F B1          ORA   C
0110 C20901      JNZ   MVLP    ;IF NOT, DO SOME MORE
0113 F20030      JP    3000h   ;JUMP TO THE TEST CODE :5000h for ramtest5
0116            END    movrmtst ; ramtest3 or 5 code will be placed here

```



figure 1 - assembly language code to move program to proper ram location



ON/OFF (cont.)

this, but the North Star does not seem to be, at least not my old one. (I can't speak for newer models.)

Burton Andrews
Potomac, MD

INSUA:

You asked about the power up and power down procedures we use. I was told when I bought my Horizon that I could leave a disk in the drive while I powered up and down. I believed this until I zapped a couple of my disks. Since then I remove all disks from the drives before I power down. I have my computer, terminal, and printer all connected to a power line filter (left over from my TRS-80) which I turn and off with a single switch. I then place a disk in drive #1 and press the reset. This is because my terminal doesn't always initialize properly when I power it and the

computer up at the same time with a disk in the drive. I have remoted my reset and the power on/off switch to a small chassis which sits on the desk next to my terminal. I found the reset on the back of the computer awkward.

Robert Floyd
Mountain Home, Arkansas

INSUA:

Re Switching question. My NorthStar is in business use and all my working disks give an automatic load to a menu showing what is on the disk. The whole system runs off a power bar with a single switch. Just insert the disk you need, switch on the power bar, and the system comes up running.

Mike O'Byrne
Ottawa, Ontario, Canada



baZic LISTINGS



By J. Burdeane Orris, Ph.D.

This short program might be of interest. It shows how to get a file directory listing using baZic and CP/M.

```
10 REM—'DIR'—DISK CATALOG USING baZic and CP/M
20 REM—J. B. ORRIS, BUTLER UNIVERSITY, INDIANAPOLIS, IN 46208
30 DIM C$(16)\CLS\ REM—C$=FILE CONTROL BLOCK
40 C$(1,1)=CHR$(0)\C$(2,12)="?????????????"\FOR J=13 TO 16\C$(J,J)=CHR$(0)\NEXT
50 !\!\!"ENTER: A, B: ",\Q$=INCHAR$(0)
60 D=ASC(Q$)\IF D<65 OR D>66 THEN END
70!CHR$(13), "DIRECTORY FOR DISK ",Q$,": "\!
80 T=CPMFN(14,ASC(Q$)-65)\F=17\Q=CPMFN(26,128)\L=0
90 T=CPMFN(F,ADDR(C$))\IF T=255 THEN 140
100 B=129+T*32\!": ",\FOR X=B TO B+7\!CHR$(EXAM(X)),\NEXT
110 !" ",\FOR X=B+8 TO B+10\!CHR$(EXAM(X)),\NEXT\!" ",
120 L=L+1\IF L<4 THEN 130\!\L=0
130 IF F=17 THEN F=18\GOTO90
140 Q1=CPMFN(13)\GOTO 50
```

Commentary (cont.)

fore. I subscribe to almost everything, but Compass is the first thing I read, the last I'd drop, and the only one I save for its continuing value. Like the old song goes, 'Stay as sweet as you are'!

2) Best software vendor in the US of A--Allen Ashley. Although I'm certainly not competent to judge it in depth, his PDS package does everything I expected it to do, and I have yet to find anybody or any company that comes close to Allen in providing support.

SUPERSOFT has been very good, within my somewhat limited experience with them. Their TFS (Text Formatting System) actually comes very close to being a full-fledged word processor at a price that's hard to beat.

3) If you haven't already done so I suggest that you check out the SCAN and RENUMBER packages offered by SOFTWARE SERVICES out in Carpentaria, CA (805 684-8259). They install themselves easily into your BASIC and are powerful and useful extensions to an already top-notch BASIC. SCAN searches your program for specified variables or strings, prints out each line in which they are found, and does substitutions globally as directed. RENUMBER provides lots of flexibility in renumbering groups of lines within programs, hence moving them around, etc. It is rather slow on long programs, but very useful. I'd find it hard to do without them now (around \$40 for the pair).

4) A few thoughts about the printer-screen echo problem in 5.2 BASIC. Extensive rewriting of the I/O can be avoided by using a patch. The patch itself is simple and can be stuck in with the DS command of the North Star monitor. Assume DOS ORG = 100H, and the patch at address nn00 in RAM. Then change the following bytes--

The 4 bytes starting at AE7--

C3 00 nn 00 JMP to patch

At the patch--

nn00	78	Char, in A
nn01	D3 02	Out to console
nn03	C9	Return
nn04	3E 01	Device # in A
nn06	CD DA 0A	Call prtr OUT
nn09	3E 00	Dev. # back in A
nn0B	C9	Return

To activate, FILL nn03,00
(in place of RETURN)

To restore, FILL nn03,201
(put back in the C9)

Where to put the patch?? Well, in Single Density 5.2, you need something that North Star not only doesn't provide on their disk, they don't even tell you is needed--a BOOT-PROM program. As has been noted in Compass, Pavel Breder has one available, and one of the things which he tells you is that the first 30 bytes of the PROM program on his disk are not used. So put the patch there. An alternative (and one maybe useable in Double-Density too, though I can't say for sure) is to add an extra block (100H) to your DOS, and ORG the BASIC etc. one block higher.

But there's a catch to this approach. After you have done SF to save the longer DOS file, you will find that when you turn on or reboot the system, the extra block didn't load. With SD and Breder's BOOT-PROM programs the fix is easy. After you have done all the relocating and made the discovery that your new relocated systems disk runs fine but won't load your extra DOS block, load the new DOS into RAM. The byte at 0018H above where you loaded the DOS is '06'. Change it to '07', then save the DOS back to disk. Now it will do what you wanted, and you've got plenty of space in which to write all the new I/O goodies you can think of--like Maguire's variable speed scroll,

CLUBS CLUBS CLUBS

Atlanta Users' Group

The purpose of this organization is to create a public domain software library, exchange tips and information on hardware usage, supply technical advice to members, and promote fellowship.

Membership requirements are very minimal. You must have a North Star system and pay a \$20.00 annual fee. There will be a monthly newsletter and, if the members desire, there will be a monthly meeting at a yet to be determined place.

Please respond with your expectations of this group and what features you would like to see in the program. Write to:

Atlanta North Star Users Group
P.O. Box 98397
Atlanta, GA 30359

Fremont Users' Group

As members of INSUA since last summer, we are most interested in establishing a Users Group here locally in Fremont, which would include current members of INSUA, and also new members that we are able to generate in our territory, the Fremont-Hayward-Pleasanton-Dublin-Union City area of California. The main purpose is to get together regularly and learn something together about our common interest, the North Star system.

Very truly yours,
Peggy Ruffin
Asnar Enterprises
4127 Bay Street, Suite 3
Fremont, CA 94538
(415) 490-5050

Buffalo Users' Group

INSUA:

I am a North Star dealer in Buffalo, NY, and I would like to orchestrate the formation of a chapter of North Star users in the Buffalo area. There is quite a large base of NS users here, especially at the University of Buffalo, where NS IS the micro of choice.

Of particular interest to me is putting together a database of all people who are interfacing the NS with electronics instruments using A/DA (analogue/digital) boards. If you could assist me with any names, I would be very grateful.

Sincerely,

David Stott
Computer Resource of Buffalo
5560 Main Street
Williamsville, NY 14221
(716) 633-9510



INSUA:

You are probably aware of this, but in the Nov. 8, 1982 issue of Infoworld, p. 31, an error in JRT Pascal, Version 2.0, was reported: "... it cannot cor-

rectly multiply floating point numbers in the form X.Y*0.0. Instead of 0.0, the answer it gives is X.Y ... its LN function (also) terminates its floating-point multiplication underflow-error during execution."

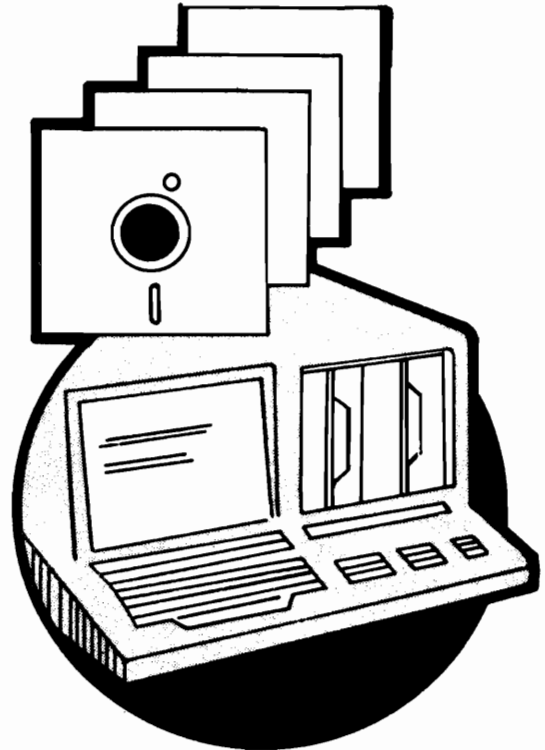
Wendell Allred
Provo, Utah



FOR SALE

I have purchased another system that runs under MP/M and has DMA disk controllers. I currently have my North Star system for sale. It is a Horizon II with 2 Quad floppies, 288K, with 18 meg disk running TSS/C. The disk drive is under 90 day warranty. I am asking \$7,000. There are no terminals or printers in the deal. There would be a \$300 discount for any INSUA member purchasing it.

Ron Morrison, Morrison Family Home Center, 171 E. Sunset Dr., P.O. Box 188, Cannon Beach, OR 97110. Tel. (503) 436-1122.



ON/OFF

[In Compass II, no. 4, Saul Levy confirmed the approach to power-up and power-down implied by North Star documentation, i.e. leave the disks in the drives if you wish. Here are some other responses. --Ed.]

INSUA:

Re disks in, or not in at ON or OFF times. I've had my HOR-

II Single-Density system for about 4 1/2 years now, running versions 4.0, 5.1, 5.2, and CP/M. Except for a few accidental occasions, the system has never been turned ON or OFF without disks in both drives. I have never detected a single problem with the hardware, the disks, or the software as a result of this practise. Other computers may be sensitive to

40 TRACKS

INSUA:

I have received several letters about my article "Freed from 35 tracks" (Compass Vol. 1. no. 3, p. 30).

Most have asked for help when the writer couldn't get the patch to work with his drives.

The problem seems to be that some drives supplied by North-Star, after about 1980, were 40 track and some were not. Since they only claimed 35 track operation, they didn't bother to inform you of this.

To see if your drives are 40 track, patch the DOS as per the article and, with the cover off and the drives exposed, type the IN command using a scratch disk. Count the step pulses as you see the head move across the disk. If you count 40 pulses you have a 40 track drive. If, after 35 pulses, the head stops moving, you have a 35 track drive.

Most of the newer drives are 40 track if you are thinking of upgrading.

Joe Maguire
2321 Foxhall Dr.
Anchorage, AK 99504

INSUA:

I have a North Star Horizon 2D (Double Density). I converted my DOS to 40 track using the information given by Joe Maguire in Compass, Vol. I, no. 3. I further researched and converted BASIC, CO, CD, DT, and CF to 40 track.

I couldn't get DT to perform properly past 35 tracks so I thought that I had made a mistake or had missed some code that needed changing in DT. I went ahead and converted all my programs to 40 track without further checking. When I ran

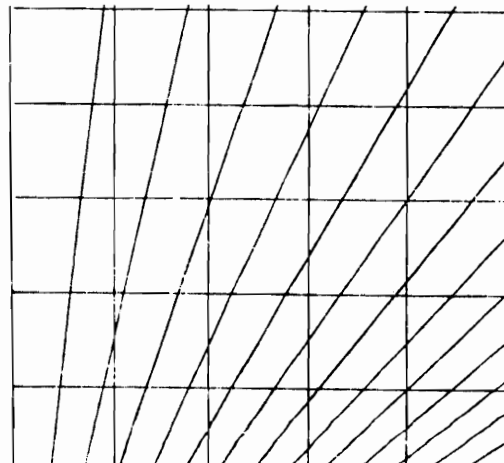
the mailing list for the local VFW post, I found that the data was scrambled and useless.

At first I thought that the procedures given by Joe Maguire were incorrect, but have since learned that I do not have disks that will operate satisfactorily at 40 tracks.

After talking with Bob Hogg of Micro Complex and Mike Boland of S.A.I.L., I learned that the Shugart SA-400 disk had a detent that drops into a groove which corresponds to the 35th track. When the head is positioned further in from this groove, it jumps and skips some tracks and gives unpredictable results. So, if you have an SA-400 drive, forget the 40 track option. If you have an SA-400L disk drive unit or one of the 40 track drivers, you are in luck.

I have attached a listing of the various patches which are needed to change your drives to other track configurations. I have no way of accurately testing these patches so I am sending them for information only and in hopes that they will prove useful to some other North Star user.

Yours sincerely, Robert Floyd
Mountain Home, Arkansas



DOS ASSEMBLER

By S. Joseph Toy
Chico, CA

A couple of weeks ago I received a copy of INSUA disk #1006, the assembler-disassembler. Although I haven't tried all the programs or all the commands, I think I have things puzzled out enough to do a little work with it.

The SCS21 assembler appears to be a good one but there doesn't seem to be any way to route such commands as LIST, ASSM, ASSME, or SYMB to the printer. The BASIC program, PRTDOS, will allow printing out the source code directly from disk but it has a few problems.

The first problem is the printing of an extraneous character at the beginning of most

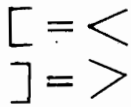
lines. This appears to be a character counter, which does not need to be printed out.

The second problem is that the printing will not stop until the last line is printed. A listing of the program seems to indicate that the printing should stop after 15 lines, and can be resumed by pressing the RETURN key. In printing out EXAMPLE, the printer just kept going past the bottom of a sheet of paper.

To correct these problems I have made a few changes; deleted a few lines and added a few of my own. If output is to the CRT, a maximum of 23 lines will be printed; if to the printer, a maximum of 50 lines.

A copy of the modified program, which I have named PRTDOC1, follows.

```
900 REM PROGRAM "PRTDOC1".
910 REM FOR LISTING ASSEMBLER SOURCE CODE DIRECTLY FROM DISK.
920 REM A MODIFICATION BY S. J. TOY OF "PRTDOC", FROM INSUA DISK #1006,
930 REM ASSEMBLER-DISASSEMBLER.
940 REM LAST MODIFIED 821217.
1000L=0
1010INPUT"FILE NAME & ,DR :",F$
1020 INPUT "FILE TYPE :",T
1030INPUT"TO PRINTER?",P$
1040IFP$(1,1)[ ]"Y"THEN P=0ELSEP=2
1050OPEN#0%T,F$
1060READ#0,&A
1070READ#0,&A
1080IFA=1THEN1150
1090IFA]13THEN1110
1100 !#P:L=L+1:REM SEND CR, INCR LINE COUNTER
1102 IF P=0 THEN 1107:REM OUTPUT PORT
1105 IF L=50 THEN GOSUB 1190:REM MAX NO.LINES ON PAPER
1106 GOTO 1060:REM READ ANOTHER LINE
1107 IF L=23 THEN GOSUB 1190:REM MAX NO.LINES ON SCREEN
1108 GOTO 1060:REM READ ANOTHER LINE
1110A$=CHR$(A)
1120PRINT#P,A$,
1140GOTO1070
1150CLOSE#0
1160END
1190INPUT"PRESS RETURN TO CONTINUE",X$
1200L=0
1210RETURN
```



BUSINESS APPLICATIONS



By Wm. R. Harmon, Vice President
Matson Navigation Company
San Francisco, CA 94119

I'll take this opportunity to make a few comments impelled by Mr. Spelman's letter in Compass, Vol 2, no 2. I, too, am primarily a business user of North Star's products, although I have recently purchased an Advantage for my own use at home, as well as one for a non-profit organization I head, and have supplemented our workhorse Horizon here at Matson with an Advantage.

We had an Advantage at Matson for some months on a Beta test for North Star, and, to put it simply, fell in love with it. From the standpoint of the non-computer-genius user, it is an ideal machine, rugged as a Land Rover, most forgiving of klutz input made by temporary help. One hopes that varied user-oriented software for business graphics will soon be marketed (by North Star or whomever).

I find much of Compass's content well over my head. What I would find helpful would be more novice-oriented information useful in expanding the machines' capabilities. A good

article on programming short-cuts, algorithms for North Star Basic, how to get machine-language routines into the machine and where, etc., would be most welcome. Remember, we ain't ALL that well eddificated on the more esoteric aspects of computers. We can always read more than we may want to know about CP/M etc. in other publications; what we need in Compass is strictly North Star-related stuff.

In the course of putting our entire tariff publishing activity on our North Star, I found that denizens of other departments (freight scheduling, marketing, equipment control, etc.) were hanging over our shoulders. I wound up writing a couple of (I think) neat programs to produce our entire fleet schedules and competitive schedule reports, including a routine which permits shifting the data within the schedule to allow updating without rewriting the whole blessed thing.

Emboldened by this success I went on to write another program which automatically rates out the huge, exotic 'widgets' we sometimes carry to Hawaii, including long-length charges, heavy-lift charges, and a number of other variables of somewhat capricious nature, with both metric and our antiquated Western system of weights and measures displayed.

Recently we have had the pleasant experience of being able to report to our President that a database management application to our fleet of leased containers (using good old Info-Manager!) plus a little ingenuity and experimentation has resulted in an annualized saving of well over \$130,000. Nobody around here is sneering any more at my passion for the North Star micro.

40 Tracks (cont.)

ROBERT FLOYD
1109 Maple Street
Mountain Home, Arkansas 72653

February 12, 1983

Patches to Reconfigure DOS 5.2 for Different Numbers of Tracks

	Patches	Code for 40 tracks
<u>DOS 5.2</u>		
ORG	+ 36AH Ld A, Tracks-1	3E 27
	+ 370H Add A, Tracks	C6 28
	+ 54FH Ld A, Tracks*2	3E 50
	+ 551H Ld DE, -(Tracks*20)	11 E0 FC
	+ 556H Ld DE, -(Tracks*10)	11 70 FE
Joe Maguire, COMPASS 1-3		
<u>CD</u>		
ORG	+ 073H Ld BC, Tracks*20	01 20 03
	+ 08AH Ld BC, Tracks*10	01 90 01
	+ 24DH Ld A, Tracks*2	3E 50
	+ 24FH Ld DE, -(Tracks*20)	11 E0 FC
	+ 254H Ld DE, -(Tracks*10)	11 70 FE
Ron Hayter, COMPASS 2-1		
<u>DI</u>		
ORG	+ 1BCH Ld A, Tracks*2	3E 50
	+ 1BEH Ld DE, -(Tracks*20)	11 E0 FC
	+ 1C3H Ld DE, -(Tracks*10)	11 70 FE
	+ 098H Ld A, (Tracks)	3E 28
<u>CO</u>		
ORG	+ 0A4H Ld HL, (Tracks*20)	21 20 03
	+ 160H Ld DE, (Tracks*10)	11 90 01
	+ 44BH Ld A, Tracks*2	3E 50
	+ 44DH Ld DE, -(Tracks*20)	11 E0 FC
	+ 452H Ld DE, -(Tracks*10)	11 70 FE
<u>BASIC</u>		
ORG	+ 665H Ld HL, -(Tracks*20)-1	21 6F FE
<u>CF</u>		
ORG	+ 3A4H Ld A, Tracks*2	3E 50
	+ 3A6H Ld DE, -(Tracks*20)	11 E0 FC
	+ 3ABH Ld DE, -(Tracks*10)	11 70 FE
	+ 04EH Ld DE, -(Tracks*20)-1	11 6F FE
	+ 182H Ld DE, -(Tracks*20)-1	11 6F FE

Note: These patches have been briefly tested and appear to work.

CP/M '83

Two items in particular at the CP/M '83 Show in San Francisco caught our attention. Inappropriately, both were hardware items rather than CP/M related software; but then hardware has an edge over software in making a quick impact on the casual observer.

The first item was a "mouse," a small box-shaped device which the user can roll around (or slide around) on a flat surface: the mouse is interfaced to the computer in such a way that movement of the mouse governs the movement of the cursor. Using a mouse is said to make graphics generation and text editing significantly easier and quicker.

Compare the mouse to the key-generated cursor motions of the standard screen-oriented editor. With WordStar, for example, to move the cursor from the top left corner to the middle of the third line from the bottom of the screen, the user hits a series of ^X's to move the cursor down the screen, then a series of ^D's or ^F's to move across the line. With a mouse, the user moves the mouse diagonally on the table-top surface in one smooth motion. The cursor "follows" the motion of the mouse.

The mouse-driven cursor is probably faster by a factor of two or three, perhaps more, than the key-driven cursor.

Many mice (mouses??) we've read about have a ball which turns as the mouse is moved about: the rotational motion of the ball is tracked, and interpreted by sensors and a software program.

The Mouse Systems Corporation mouse comes with and requires the use of a flat pad inscribed with a two-color grid: the pad is 228mm x 280mm--about 9 x 11 inches. The user slides the mouse around the board: photo-

detectors in the mouse's "stomach" detect movement across the inscribed lines, whether horizontal movement, vertical movement, or rotational movement. An interface box and software interprets information generated by the photodetector.

The MSC handout cites common myths associated with mice. The first of these is that the mouse is inferior because you have to take your hands off the keyboard to use it. In reply the following "fact" is offered:

Your hands must leave the keyboard to use most pointing devices. Overall pointing time is dominated by selection time, not homing time (the time to reach the pointing device). The mouse has the fastest total time.

This defense of the mouse, however, compares it to other pointing devices like joysticks and light pens. It would take a real trial and not just a written argument to convince a potential user of the insignificance of the time lost in moving the hands from the keyboard. (I think that if I purchased a mouse, I'd find some way to control it by foot-power, so that I could do my wordprocessing in the manner of an organist who uses his feet as well as his hands.)

For more information, write Mouse Systems Corp, 2336H Walsh Avenue, Santa Clara, CA 95051.

* * * * *

The brains behind the Texor company must have had some experience with the old Wurlitzer juke boxes, since Texor uses approximately the same principle to handle multiple diskettes.

Disks are loaded into cartridges which are placed side-by-side to make up a long rack. When the user specifies a given disk, the rack moves right or left until the correct disk is positioned in front of a disk

NEVADA EDIT

By Dana M. Schaul
Sacramento, CA

I recently purchased "NEVADA EDIT" from Ellis Computing to use as my main text editor when writing source code for JRT Pascal. I had a problem when trying to configure it for use with my Advantage (64K). Unfortunately, the configuration file (NVEDIT.COM) has no provision for setting up the Advantage configuration.

I called Ellis Computing, and they were most helpful. I was lucky enough to reach one of the technical assistants there, who had just finished talking to another Advantage user, earlier that day. With his help (and the help of Mr. Sydney Ranter (415) 542-7689), I was able to get Nevada Edit installed, up, and running within 5 minutes. Attached are the responses which must be made when prompted by NVEDIT.COM.

Follow the instructions on p. 42, Section 5, "Configuring and Unknown Terminal", to the point where "U" is selected for Unknown Terminal. Then give these answers:

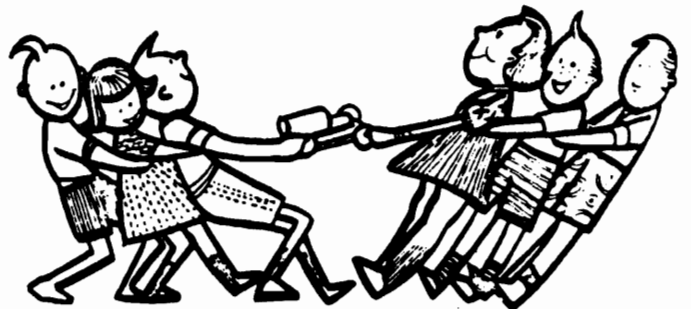
Number lines in display = 24
Characters per line = 80
Serial-connected term? = T
Term follows pattern? = Y
Chars for cursor pos = 2
First lead char = 1B
Second lead char = 3D
No. separator chars = 0
No. ending chars = 0
Offset to line = 20
Offset to column = 20
Column before line? = N

No. chars to clear screen
and home cursor = 2
Hex chars = 1EOF
Chars to insert line = 2

Hex chars = 1B4C
Chars to delete line = 2
Hex chars = 1B4D

File type used as default
(3 chars) = CBL/TXT/PAS
Default tab change = N
(optioal)

Configuring of EDIT.COM is
now complete!!



CLIP TIPS

Punctuation Protection

By Rick Ahlgren
Davenport, Iowa

In Compass, Vol. I, no. 4, the NONUMBERS article listed four fill statements which altered BASIC 5.0 to protect colons, semicolons, and right and left square brackets. For BASIC 5.2DQ at E00, try the following:

FILL 4593,58 (protects colons)
FILL 4586,59 (protects semicolons)
FILL 4579,93 (protects right bracket)
FILL 8600,40 (protects left bracket)

Ribbon Renewer (cont.)

izer is that each machine can handle only one kind of ribbon: if you have two kinds of printers, you will probably need to purchase two **Ribbonizers**.

We suspect that other re-inking machines must be available; since we have made no comparative tests, however, we cannot say that one is better than the other. A re-inker which could be adapted for different ribbons might be considered an even better buy.

For more information on **Le Ribbonizers**, write to Ben Torres Ribbon Service, Box 1727, Redlands, CA 92373. This company also sells renewal cartridges for printer ribbons: with some care, you can pry your plastic ribbon cartridges open and replace the ribbon, whether cloth or carbon filament, at a fraction of the cost of a new one.

LETTERS TO THE EDITOR



INSUA:

Has anyone modified the N* DOS to insert automatic spaces after the line numbers on printouts? I don't want to reinvent the wheel but if no one else

has--I will have to do it. Also I am using single density DOS release 4 from 1978--and I can't seem to get any information on how current that is. The distributor in Canada has changed twice in the last year, and they aren't interested unless you are going to buy a new machine.

Fred J. Looker
Goederich, Ontario

[The sad news is that DOS 4 is not really current. However, INSUA's disk library includes a 5.2 DOS and BASIC (disk #1001) single-density as well as double density. (Keep an eye out for further news on DOS and BASIC updates in these pages!!)

You may be able to add spaces on printouts with a modification of the NONUMBER program in Compass, Vol. I, no. 4 (in fact, the earlier the DOS release, the more space in the I/O block for installing your own routines). Alternatively, a BASIC editor like N-BUS would probably serve your needs.

If your local distributor refuses to assist you, report this matter to North Star Computers Inc., 14440 Catalina St., San Leandro, CA 94577 --Ed.]

INSUA:

Please send me any information you may have regarding the subject of software. I have a Horizon 64K with 2 dbl density floppies and a Soroc terminal and am having great difficulty finding accounting software in the New Orleans area--or any software at all--and I would appreciate anything you could do to help. Thanks.

Louis J. Ulmer
1705 Madison St.
Metairie, LA 70001

[INSUA can't respond to a general request like "software" except by recommending all back issues and all future issues of Compass. However, maybe in-

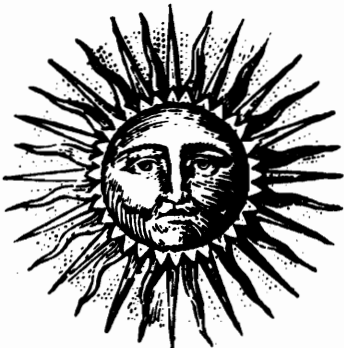
drive: a mechanical finger pushes the disk into the drive, and the disk is then used in the normal way. Later, the disk can be withdrawn from the drive, the rack can move right or left, and a new disk can be pushed into the drive.

Texor devices come in two main formats. First, the user who must swap disks in and out of drives with great frequency can use the Texor like a set of mechanical hands and fingers to perform this task for him. A rack of fifty disks can serve in place of a larger drive, for example, an expensive hard disk drive. Of course, this system cannot be quite as convenient as a hard disk, since files will be limited by the size of the individual floppies, and disk-swapping is hardly instantaneous. Still, certain applications can make perfectly good use of such a battery of floppies.

The other format is for disk copying: a rack can be loaded with 80 to 160 blank disks, and each disk will be loaded in its turn into the drive for copying. At the end of the process, all 80 or 160 disks will contain copies of a master disk.

For more information, write Texor Corporation, 1901 Old Middlefield Way, Suite 1, Mt. View, CA 94043.

--Ed.



RIBBON RENEWER

Do you do a lot of printing? Are you always buying new cloth ribbons at a hideous price? We've found a nifty solution to this problem in **Le Ribbonizer**, a motor-driven re-inking machine.

When a continuous-loop printer-ribbon begins to fade, it can be renewed in about twenty minutes. **Le Ribbonizer**, a Rube Goldberg name for a Rube Goldberg device, contains two absorbant rollers which are first inked by the user: then a ribbon cartridge is placed on the machine in such a way that the ribbon is drawn out of one side of the cartridge, pulled across the two rollers (where the ribbon absorbs fresh ink), and is then pulled back into the cartridge.

It isn't hard to be convinced that **Le Ribbonizer** is a good investment. The basic machine sells for \$39.95 plus tax. My local dealer charges \$8.00 each for Diablo Hy-Type II ribbons. Ten ribbons would cost \$80.00 plus tax. Using **Le Ribbonizer**, I can get one ribbon and at least ten renewals for about \$50.00--so the machine pays for itself the very first time around. Thereafter, of course, the saving is even greater, about \$7.90 for each ribbon renewal (ink costs about a dime each time). If you're like the Compass editor, you'll already have a drawer full of old (but renewable) ribbons, and may not have to buy a new ribbon for the next twenty years. If you're starting from scratch, you may want to have two ribbons on hand, since a ribbon which has just been renewed should be allowed to remain overnight before use, to give the solvent a chance to absorb evenly and to do its work on the old ink still left in the ribbon.

One drawback with **Le Ribbon-**

LETTERS TO THE EDITOR LETTERS TO THE EDITOR LETTERS TO THE EDITOR
LETTERS TO THE EDITOR LETTERS TO THE EDITOR LETTERS TO THE EDITOR
LETTERS TO THE EDITOR LETTERS TO THE EDITOR LETTERS TO THE EDITOR

dividual readers or vendors will wish to reply to Mr. Ulmer. -- Ed.]

INSUA:

I am located out here in West Virginia with no North Star users in hailing distance. I have a Horizon II with 40K memory (1 NS board, 1 Godbout Ram II, and a Godbout Ram XIV). I have developed my own software, both machine language and BASIC, which I use extensively in my law practice. I find that I can do without a secretary by proper use of the computer.

Peripherally, I have a Heathkit H-19 terminal, an Olympia ESW103 typewriter and (recently purchased to replace failure-prone Centronics 737) Seiksha GP 100-A printer. I have a TTY model 33 presently inactive, and an American Anderson-Jacobson 841 which has never worked satisfactorily with the system. As far as I am concerned, it is simply an IBM typewriter.

I have developed certain DOS interfaces in my Monitor which may be of use to some of your other members. I got my indoctrination in computers with the XDS 940 timesharing systems, and I have duplicated the "NEW FILE" and "OLD FILE" messages when writing to the disk. This saves many a blasted or misnamed file error.

I recently tried a Cal Comp 2065 board with disastrous results. I added a jumper on the Disk Controller to pull the PHANTOM/ line from PROM SEL/ (7G pin 9), but this was apparently not fast enough, and I got intermittent disk errors. I plan to try again with blank 4 disabled, as I want to populate the all-important 0-1FFF region. I would appreciate any input from

other hackers who have tried to use the PHANTOM/ line on North Star.

Yours truly,
Robert G. Durnal
Box 68
Junior, W.V. 26275

[Check members' list in Compass, Vol. II, no. 3, for INSUA members in your zip-code area -- Ed.]

INSUA:

We have been using North Star computers for over 3 years to control our medical imaging systems. Since February we have been exclusively using the ADVANTAGE, and have designed some novel hardware including high speed data aquisition and stepper motor control.

Put in our vote for more ADVANTAGE articles in your newsletters.

Scott A. Wiener
Lunar Radiation Corp.
(Microprocessor-based Medical Instrumentation.)
10 N. Charter St.
Madison, Wis. 53715

[Scott Wiener's letter was accompanied by an ad/description of a "Dual Photon Spine Scanner"--write him for more information. --Ed.]

INSUA:

Does any member have a program to implement graphics on a NEC Spinwriter?

Clive Howe
778 Adair Avenue, N.E.
Atlanta, GA 30306

LETTERS TO THE EDITOR LETTERS TO THE EDITOR LETTERS TO THE EDITOR
LETTERS TO THE EDITOR LETTERS TO THE EDITOR LETTERS TO THE EDITOR
LETTERS TO THE EDITOR LETTERS TO THE EDITOR LETTERS TO THE EDITOR

INSUA:

I would like to see an article giving a clear explanation of patching, particularly large patches. Also an article on how to use source code. I have been waiting for years for these but have never found any good articles in the various magazines.

Al Bloom
Laguna Hills, CA

[See Joe Maguire's article in Compass II, no. 4 --Ed.]

INSUA:

I am particularly interested in the article on upgrading the N* 16K RAM board, and I would like to see more articles on the future on upgrades. The tips on using WordStar, and the disassembly with comments on the disk controller PROM are more than worth the price of membership already!

Hubert H. Love, Jr.
Los Angeles, CA

[See Steve Leibson's article concerning brain surgery on the 32K memory board in Compass II, no. 4. --Ed.]

I enjoy the reviews of equipment and software that are in the Compass. I subscribe to Byte and Microcomputing, mainly for this purpose, but there doesn't seem to be much for North Star in these publications any more. I personally wouldn't mind advertisements in the Compass provided that they were all North Star oriented and were informative. I have bought the Secretary word processor and the relocated prom for my computer based on information from the

Compass. In addition I have been told about a CP/M look-alike DOS and a NS compatible BASIC which has been developed and about which I have asked for more information, all as a direct result of articles in the Compass.

Robert Floyd
Mountain Home, Arkansas

INSUA:

Do you have, or plan to run, any articles on getting more than 64K memory via bank selecting.

Harry Binswanger
New York

[No plans, but we'd like to have some articles--how about it, readers? --Ed.]

INSUA:

I own a NorthStar Advantage, MX-100 printer, and a modem. I am looking for scientific and engineering applications.

Mark Eisenhauer
6603 Park Place Dr., Apt. 205
Fort Worth, TX 76118

INSUA:

In response to your question, what do we want? I can probably best describe what I want by telling you what I do with the NorthStar. At home I have the N* Horizon which I use for work processing and to learn and review software for business. At work we use the Advantage computer (2) for word processing, spreadsheets, etc. Our business presently is run under an Alpha Micro w/90 .eg. CDC disc. I am interested in the new NorthNet and the possibility

LETTERS TO THE EDITOR LETTERS TO THE EDITOR LETTERS TO THE EDITOR
LETTERS TO THE EDITOR LETTERS TO THE EDITOR LETTERS TO THE EDITOR
LETTERS TO THE EDITOR LETTERS TO THE EDITOR LETTERS TO THE EDITOR

of running our business on this network with or without the Alpha Micro. We are learning CP/M and I find this a more flexible operating system. I don't know much about programming but can follow about 3/4 of what I read in Compass.

Tony Lerner
Sun Valley, CA

INSUA:

How do I disable the 'at the' symbol (ASCII 064D)? I need to use this for printing foreign characters on my printer.

Re Wordstar--enough already. Too much of the last issue devoted to WordStar with Epson, a subject of narrow interest covering what has to be the most overrated, overpriced and outdated pieces of software available. No one should need all that advice and all those mods to make a word processor run on their computer.

Re Software listings--please do not omit them from Compass. The article 'User friendly input' in Vol. II, no. 2, sounded interesting but I don't want to purchase a disk just to find out.

Re software: Heaps of congratulations to those responsible for arranging the JRT distribution--we could do with more of this. How about doing the same with other software BAZIC for DOS or CP/M for example, or perhaps a spread sheet program or a GOOD word processor like Select? With a bulk purchase we should be able to keep the price down. I have written to NorthStar to suggest that they should sell a

NorthStar BASIC for CP/M at a very low price and thus encourage all CP/M users to use NS BASIC; good advertising for NorthStar and good programs for the rest of us!

Re: Modems: There is a company in Ottawa that sells modems in bare board, kit, or completed form. They can be acoustic or direct connect, and work very well. I hooked mine into the Horizon motherboard for power but they also sell a power supply if you want a completely portable modem. The Ottawa computer group got an excellent price by arranging a bulk purchase of these. Current cost is \$57 bare board and manual or \$194 complete (Canadian dollars). The unit is a 300 baud originate/answer type. The company is: Warren Instrotech Ltd., P.O. Box 5739 Station F, Ottawa, Ontario, Canada K2C 3MA. Phone (613) 728-9120.

Mike O'Byrne
Ottawa, Ontario, Canada

INSUA:

I would be very interested in any experience anyone has had in networking Horizons. We have 14 Horizons in our Microcomputer Laboratory and we need a method to share a hard disk and a high speed printer.

J. Burdeane Orris, Ph.D.
Associate Professor
College of Business
Administration
Butler University
4600 Sunset Avenue
Indianapolis, IN 46208

[Write to NorthStar about their new NorthNet--we were mightily impressed by a pilot demonstration!! --Ed.]



DISK LIBRARY

The following disks are available in the INSUA disk library. A full description of each disk will be included in the next issue of Compass--or see the recent Special Issue for details. Disks marked with an asterisk are \$10.00; all others are \$15.00. **Specify single or double density for all orders.**

Order from INSUA, Disk Library, P.O. Box 2910, Fairfield, CA 94533.

- DOS FORMAT
- #1001* N* DOS and BASIC 5.2, Single Density
 - #1002* Ditto, Double Density
 - #1003. TELESTAR Modem for DOS
 - #1004 Compass, I, 1

- #1005 Compass, I, 2
- #1006 DISSASSEMBLER, CODER
- #1007 Compass, I, 3&4
- #1008 MICRO-COUNT II
SD 2 disks: \$20.00
DD 1 disk: \$15.00
- #1010 Compass, II, 1

CP/M FORMAT
- #1011 CP/M Utilities #1
- #1012 JRT Pascal
- #1013 MODEM for CP/M
- #1014 CP/M Utilities #2

NO. 1001, 1002, 1003, 1004

Insua

**Publishers of the Compass Newsletter
International NorthStar Users Association
PO Box 2910 • Fairfield, CA 94533**

FIRST CLASS PERMIT



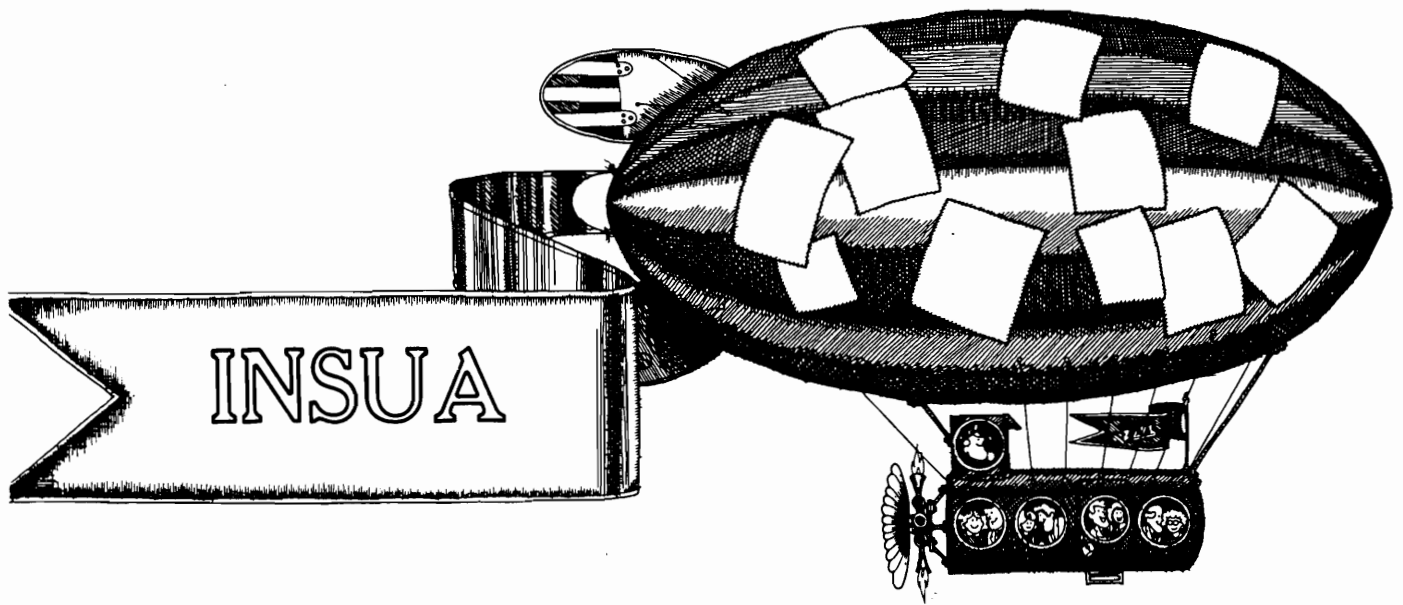
2766 ()
JAMES TATE
23914 SPRING DAY LN.
SPRING, TX 77373

POSTMASTER: ADDRESS CORRECTION REQUESTED
Please enter key number on form 3579

THE COMPASS NEWSLETTER

INTERNATIONAL NORTH STAR USERS ASSOCIATION

VOLUME III, NO. 2



NOTE FROM THE EDITOR

THE PHANTOM NULL

Hugh Power of Power's Computers in Richmond, CA, first told us of the "phantom null" phenomenon in the 5.2 release of North Star BASIC. To see the phenomenon in action, do the following (5.2 BASIC only):

```
20 A=1
RUN
READY
10 REM XXXXXXXX
EDIT 30
LINE NUMBER ERROR
LIST
10 REM XXXXXX
```

Notice that line 10 has six X's instead of the original eight. Inspect the program in memory with a monitor: a display (using DA) of the 10Hex bites starting at 4220 will reveal that the middle two X's (the fourth and fifth) have been turned into nulls (00Hex). With the monitor they are evident, but in the BASIC LIST they disappeared without a trace or an error message.

Given the information that true nulls (as distinguished from ASCII 0's) are not permitted in a BASIC program, except in (tokenized) line numbers, it becomes evident why the characters are not displayed to the screen, and, more significantly, why program statements with phantom nulls **will not RUN, but will crash ignominiously.**

What to do? First, be aware of the problem with 5.2 BASIC: **if you attempt to LIST or EDIT a non-existent line, you run the danger of contaminating your BASIC program with phantom nulls.** The first rule is to take care to EDIT or LIST only existing lines. The second rule is to be alert to the possibility that a BASIC program which you have already edited may have failed not because the syntax or procedures are incorrect, but

because it contains one or more phantom nulls. The third rule is to examine your program, perhaps with a monitor, for phantom nulls. Once these are found, it is sufficient to re-type any lines with the nulls. (Remember to distinguish line-number nulls from nulls within the lines of the BASIC program.

NorthStar and INSUA are both aware of the problem. There will be an announcement either in this issue of Compass or the next of a general solution to this problem. The timing is such that as your Editor is writing this note, neither North Star nor INSUA is certain of the exact actions which will be taken for the remedy.

NorthStar's candor with INSUA over this vexation has been admirable, as has North Star's declared intention to get to the bottom of the problem once and for all. Keep on the alert for further news concerning the phantom null!!

NEED MORE MEMORY?

The astonishing news from the Eighth San Francisco Computer Faire is that the price of hard disk drives is falling rapidly. A Morrow 16 **Megabyte** drive, for example, was selling for about \$1500, which is about \$100 per million bytes, or \$10.00 per hundred thousand. The drive is so fast its action is practically indistinguishable from RAM memory. One INSUA member plugged the S-100 controller board into his Horizon and had the disk up and running in minutes.

North Star owners interested in increasing memory would do well to consider adding high-speed drives (whether floppies or hard disks) rather than adding to internal RAM. The choice is yours--but be alert to the options!

MULTI-LINE USER-DEFINED FUNCTIONS



STRUCTURING BASIC PROGRAMS WITH MULTI-LINE USER-DEFINED FUNCTIONS

By J. Burdeane Orris
BUTLER UNIVERSITY
Indianapolis, IN 46208

[Indented, numberless lines in BASIC program statements in this article belong to the previous numbered line--the lines are divided here only to fit the margins. --Ed.]

One of the most powerful features of a programming language is the multi-line user-defined function. Indeed, the relatively new language C is largely organized around the concept of functions. As Kernighan and Ritchie state: "Functions are really the only way to cope with the potential complexity of large programs" (The C Programming Language, 1978, p.22).

Unfortunately, most popular versions of BASIC do not include multi-line functions. The single-line function provided in most versions of BASIC provides a short hand reference to an arithmetic statement, but for those versions of BASIC such as North Star BASIC with multi-line functions, the power of the function concept is greatly increased.

While multi-line functions can certainly be used for numerical procedures, one of their most important but little recognized uses is for various utility purposes. Several examples are discussed below to illustrate how multi-line func-

tions in BASIC can be used to write more effective and efficient programs. Not only do functions make the program and the programming cleaner and simpler, function modules made up of commonly used functions can be easily appended to other programs with little concern for line numbers or variable names.

A user-defined function is somewhat like a GOSUB except that it is called simply by making reference to the function name. Although this may seem like a fairly minor difference, functions have some very useful advantages, largely due to the fact that functions can be referenced directly in IF and PRINT statements without a separate calling statement. A function can be thought of as a variable or a string that takes on its value as a result of the operations specified in the function definition. A user-defined function is used exactly like a built-in function (SQRT, SIN, CHR\$, etc.) except it is the programmer who defines the function. (See Appendix for rules of function usage.)

To show how functions work and how they can be used advantageously, let's look at a fairly common programming situation, namely checking a variable for a legal range of values. For instance, at one place in the program we may want to check an input variable, L, for values between 1 and 8 (inclusive) with a transfer to line 160 if L is within range and line 300 if out of range. At another place we may want to make sure a numeric variable, P, is between A and B (inclusive) with a transfer to

MULTI-LINE Continued

line 610 if correct and line 750 if out of range. The two statements in the program to accomplish this might be:

```
150 IF L>=1 AND L<=8
      THEN 160 ELSE 300
```

```
600 IF P>=A AND P<=B
      THEN 610 ELSE 750
```

Whenever program statements with similar structure are repeated, more efficient procedures should be considered. One possibility would be to use a subroutine that sets S=1 if within range and S=2 if out of range and then use an ON-GOTO to switch to the appropriate line number, for example:

```
110 H1=1
120 H2=2
130 G=L
140 GOSUB 1000
150 ON S GOTO 160,300
.
.
.
560 H1=A
570 H2=B
580 G=P
590 GOSUB 1000
600 ON S GOTO 610,750
.
.
.
1000 IF G>=H1 AND G<=H2
      THEN S=1 ELSE S=2
1010 RETURN
```

But wait a minute--wasn't this supposed to be more efficient? It is true that there is only one IF statement, but we have a total of ten statements instead of two. The problem is that the program variables have to be 'pre-set' to the subroutine variables, and after the subroutine call we still have to switch to the appropriate line number.

Now let's see how we would handle this situation with a

multi-line user defined function:

```
150 IF FNW(G,1,8)
      THEN 300 ELSE 160
```

```
.
```

```
600 IF FNW(P,A,B)
      THEN 750 ELSE 610
```

The function definition for FNW is shown in Listing 1. While the function definition adds a few lines to the program, the call statements are much shorter and simpler. The first argument is the variable to be checked and the next two arguments are the range. If an error is detected the function returns a value of 1 (flag set) which is the numerical equivalent of True in the IF statement. If the range is valid then a value of 0 (flag cleared) is returned which is interpreted as false. An added embellishment is the "beep" (CHR\$(7)) when an error condition is detected. The usefulness of this particular function is illustrated by the fact that it is called by several of the other functions discussed below, but, more importantly, this example shows how a function can be used to accomplish a procedure other than arithmetic computation.

MORE FUNCTIONAL FUNCTIONS

Below are discussed several other utility functions, most of which were designed during the development of ECOSOFT's MICRO-STAT statistical package. I have found it useful to store all of these functions in one file. This function module can be APPENDED to new programs so that even 'quick-and-dirty' test programs can have many of the embellishments of a 'master-work', and by using these functions on all programs a further advantage is consistency of op-

erating procedures. The functions discussed below are shown in Listing 1. If necessary, removing the REM's and using multiple statements per line, the functions used in any given program would require a modest amount of memory and would probably reduce the total program size by making the program more efficient and structured.

"blinking lights"

There are often times during extended computations or during repetitive disk operations that the video display screen will appear to 'go dead' because nothing is being output. The function FNR(J) gives the appearance that the computer is 'alive and well' by flashing numbers on the screen. It also lets you check the progress of a loop. Often a lengthy procedure will not seem as time consuming if you can see the end is in sight.

The carriage return (CHR\$(13)) and the comma at the end of the line keep the numbers superimposed on the same line rather than scrolling.

The function is called by setting some variable equal to the function. For example,

```
T = FNR(J)
```

Actually this is a dummy statement since the 'action' of the function takes place in the function. The argument would be any value you want printed, usually the control variable of a loop or successive approximations of an iterative procedure.

Although this is a short function, it is preferable to repeating the print statement several times in a program and it is preferable to a GOSUB since any variable can be used as the argument.

Use of the function does slow execution slightly, but for most

purposes the advantage of its inclusion far outweighs the slight delay it causes. For example, the following loop requires about 13.4 seconds (with a North Star Horizon):

```
100 FOR J=1 TO 1000
110 T=FNR(J)
120 NEXT J
```

However, if the loop also included 1000 computations and/or disk accesses, the marginal increase of 13.4 seconds would probably not be noticed. Incidentally, the timing is the same no matter where the function definition or the calling statement appear in the program, which points out another advantage of functions vs. GOSUBs. As an experiment I checked the above timing loop converted to a GOSUB with the subroutine placed at the beginning and end of a 350 line program. With the subroutine at the beginning the time was 9.8 seconds; with the subroutine at the end the time was 32.1 seconds. Since the BASIC interpreter has to scan for the GOSUB line number each time it is called, the position of a subroutine in a large program is critical. Obviously the subroutine placed at the beginning of the program is a bit faster than the function, but you would have the traffic problem of directing program flow around the the subroutine. (If you are a real 'speed-freak' you could save a few more seconds by embedding the loop counter inside the loop, or not use it at all and stare at a dead screen.)

"What's on the menu?"

Due to the interactive nature of microcomputers, many programs are menu-oriented; i.e., at certain points in the program a list of options is displayed and the user makes a selection.

The function FNG(T\$) provides

an efficient method of menu selection. First the program would print a list of options by letter rather than by number. This allows for up to 26 options selectable by a single key-stroke. The function FNG will prompt for an input and will return the option number. For example, if you had four options (A, B, C, and D) you could use a statement like the following:

```
50 ON FNG("ADA")
    GOTO 80, 400, 600, 999
```

If the user selected option B, the function would return the value of 2 which would transfer control to line 400 where the computation for option B begins.

The argument is a three character string. The first two characters are the legal range of options and the third is the default value that will be selected if the RETURN key is pressed with no entry. If an illegal character is entered, it will 'beep' and again prompt for an option selection. Since the function uses the built-in function INCHAR\$(0), it is not necessary to press the RETURN key after selecting an option.

"Yes-sir, No-sir"

Often there is a special case of menu selection where there is a single option that requires a YES or NO response. The function FNQ(D9,Q1\$) provides a convenient method for handling this situation. If the argument D9 is set equal to 0 the prompt will be displayed "(N,Y)?", if D9 is set equal to 1 the prompt will be "(Y,N)?". In each case the first option in the pair will be the default. For example, if the prompt is "(Y,N)?" then N will yield NO and any other key will yield the default, YES. It is not necessary to press the RETURN key after a selection, and after the selec-

tion is made, the appropriate word, YES or NO, is displayed on the screen.

The second argument is the prompt which can be either a string variable or a literal constant. The function returns a value of 1 (True) if the default was selected or 0 (False) if the second option of the pair was selected. Thus the function is usually referenced directly in an IF statement.

An example: In North Star BASIC you can route output to the screen or the printer by specifying

```
100 PRINT#P, . . .
```

where P=0 for the screen or P=1 for the printer (assuming 0 and 1 are the appropriate device numbers). The following statement would be all that would be necessary to set the print switch:

```
200 IF FNQ(1,"PRINTER OUTPUT")
    THEN P=1 ELSE P=0
```

Another similar situation that can be handled with a function occurs when you want to clear the display screen if output is going to the screen or you want to do a form feed if the output is going to a printer. The following statement would use the function FNF(P) to send the appropriate code.

```
800 PRINT#P,FNF$(P), . . .
```

Similar functions can also be written to provide short-hand reference to various control and/or escape sequences that are used to select various display attributes of video display terminals and printers.

"There's no point in it."

Suppose a program involves the entry of many dollar and cent values. To speed entry,

the user enters the values without a decimal point and program divides the entry by 100. However, it is easy for the operator to enter a decimal point inadvertently, and subsequent division by 100 will lead to an incorrect entry.

The function FNP(X\$) checks the input for a decimal point or any other non-numeric entry except a minus sign in the first position. The input number is entered into a string which is the argument of the function. Each position is checked. If the input is valid the function returns a 0 (flag cleared); if there is a bad entry a value of 1 (flag set) is returned. By placing FNP in an IF statement, the entry could either be converted to a numeric value and divided by 100 or a "beep" would signal the operator to repeat the input, depending on the result of the function.

"Day by Day"

Occasionally in business and accounting programs there is a need to convert a date to its corresponding day of the week. The function FND\$(T\$) provides such a conversion. The argument is a string in the form of "MMDDYY" and the function returns the day of the week. In order to use this function, the statement DIM A\$(63) must appear in the main program. (See Rule 7 in the Appendix.)

This function also makes reference to a single-line function, FNS(A\$,J,L), which will be of interest to users of North Star BASIC. FNS\$ is a single-line function that provides for substring extraction and should be useful in many programs. The first argument is the name of the string, the second argument is the substring you want, and the third is the length of each substring. In this example it is used to select the appropriate

day of the week from the array, A\$.

"Control That Cursor"

Finally, let's take a brief look at a powerful application of functions for cursor control. Most video display terminals have certain control or escape sequences that allow positioning of the cursor on the screen prior to input or output. By using the FNC\$(R,C) function shown in Listing 1 you simply have to reference the function in a print statement and the following output will be positioned according to the arguments: R is the row position and C is the column position (counting from the upper left corner of the screen). For example, the following statement will prompt for the input of X in the middle of the screen and will print underlining to indicate the size of the number expected.

```
10 PRINT FNC$(12,40),
    "ENTER X: _____",FNC$(12,49),
20 INPUT" ",X
```

With direct cursor control you can pre-print forms on the screen, give movement and graphics effects to games, and other "really neat things." A subsequent article will explore further the technology of direct cursor control.

To quote Kernighan and Ritchie again: "A function provides a convenient way to encapsulate some computation in a black box, which can then be used without worrying about its innards. ... With a properly designed functions, it is possible to ignore how a job is done; knowing what is done is sufficient."

The BASIC functions discussed above were certainly not intended to be an exhaustive set, but I trust these examples will suggest other applications.

Most programmers who have worked with multi-line functions have found them indispensable. Hopefully multi-line functions will soon become available on other versions of BASIC, and perhaps some of the functions discussed above could become 'built-in' functions since they would certainly be as useful as many currently supplied functions.

APPENDIX

Usage Rules for Multi-line User-defined Functions

The following rules apply to North Star BASIC. The rules for other languages will be similar but may vary in some details.

1. The first line of a function specification must be of the form: DEF FNn (a,b,...) where n is the function name which can be any legal variable or string name depending on whether the function returns a numeric value or a string. The values a,b,... in parentheses are one or more variables or strings that take information to the function. The technical name for these variables is "arguments". Some examples of function definitions would be:

```
DEF FNB$(Y,Z)
```

```
DEF FNA(X)
```

```
DEF FNX2(X$,T)
```

2. Using a variable name as a function name does not affect a variable or array with the same name. For example, DEF FNB(N) does not affect the variable B.

3. Numeric arguments are 'dummy variables' that are treated as local to the subroutine. I.e., if these variable are changed during the execution of the function, they are re-set to their original pre-call values

when the function is completed. On the other hand, if a string argument is modified by the function, it remains changed.

4. A function must have at least one argument listed in the DEF statement even if it does not require one. The argument can be ignored in the function and is often set to zero in the calling statement.

5. Numeric arguments cannot refer to an entire array, but they can be individual elements of an array.

6. Functions can reference variables, arrays, and strings of the main program that are not listed as arguments, and if any non-argument variables or strings are modified by the function they remain changed after the return. This is both good news and bad news. The good news is that a function can reference many different variables and arrays without having to list them as arguments and, in effect, multiple values can be returned by the function. The bad news is the fact that it is easy to think of all function variables as local and thus make inadvertent changes.

7. If a function references a dimensioned array or string, the DIM statement must appear in the main program and not in the function.

8. At one or more places in a function there must be a statement of the form: RETURN x, where x is a variable or a string expression depending on whether the function was defined as string or numeric. This is the quantity that will be assigned to the function at the point it was called in the main program.

9. The last statement of a

MULTI-LINE Continued

function must be FNEND which is a non-executable statement denoting the physical end of the function definition. Unlike GOSUBs, functions can be embedded anywhere in a program and they will be ignored by the program flow until they are called. However, good programming practice would dictate that user-defined functions be grouped together at the end of a program. Incidentally, a multi-line function must have more than one line even if it will fit on one line. In other words, the DEF statement must have a different line number from FNEND.

10. To call a function in a program you simply make reference to the name of the function followed by appropriate arguments. The arguments must correspond to the sequence in the DEF statement. For example, if the DEF statement contained one numeric and one string argument such as:

```
DEF FNT(A,B$)
```

then each calling statement must contain a numeric argument followed by a string argument. For example,

```
PRINT FNT(C,"XYZ")
```

```
T=SQRT(FNT(89.36,D$))
```

would be examples of legal ways to invoke the function. Note that the arguments in the DEF statement must be variables, but may be constants or literals in the calling statement.

On the other hand, the following are examples of illegal calling statements:

```
PRINT FNT("XYZ",C)
...wrong sequence
```

```
T = FNT(A)
...wrong number of arguments
```

Listing 1. Some Examples of Multi-line User-defined Functions.

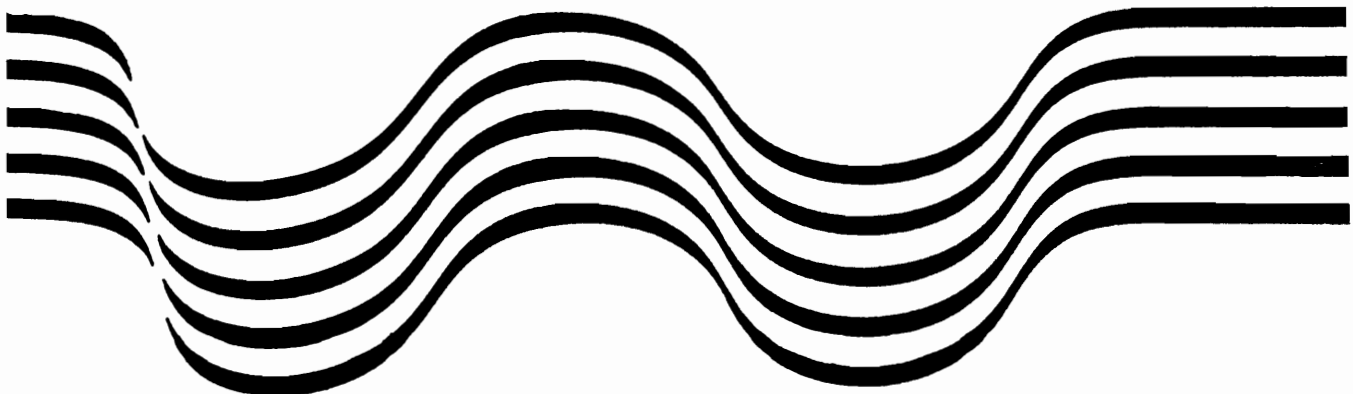
```
1000REM-----FNW
1010 DEF FNW(G,H1,H2) \REM--RANGE CHECK
1020 IF G>=H1 AND G<=H2 THEN RETURN 0
1030 !CHR$(7),\RETURN 1
1040 FNEND
1050REM-----FNR
1060 DEF FNR(J) \REM--LOOP COUNTER
1070 !CHR$(13),%4I,J," ",
1080 RETURN J
1090 FNEND
1100REM-----FNG
1110 DEF FNG(T1$) \REM--MENU SELECTION
1120 !\!"ENTER: OPTION: ",\G$=INCHAR$(0)
1130 IF ASC(G$)=13 THEN G$=T1$(3,3)\!G$,\G=ASC(G$)-64
1140 H1=ASC(T1$(1,1))-64\H2=ASC(T1$(2,2))-64
1150 IF FNW(G,H1,H2) THEN 1160!\RETURN G
1160 !CHR$(13),\GOTO1120
1170 FNEND
1180REM-----FNQ
1190 DEF FNQ(D9,Q1$) \REM--YES/NO RESPONSE
1200 IF D9 THEN D9$="YESNO " ELSE D9$="NO YES "
1210 IF D9 THEN D8$=" (Y,N)? " ELSE D8$=" (N,Y)? "
```

LISTING 1. Continued

```

1220 !\!Q1$,D8$, \Q$=INCHAR$(0)\IF Q$=D9$(4,4) THEN 1230\!D9$(1,3)\RETURN 1
1230 !D9$(4,6)\RETURN 0\FNEND
1240REM-----FNF$
1250 DEF FNF$(P) \REM--CLEAR SCREEN/FORM FEED
1260 IF P THEN F$=CHR$(12) ELSE F$=CHR$(27)+"*"
1270 RETURN F$
1280 FNEND
1290REM-----FND
1300 DEF FNP(X$) \REM--CHECK FOR DECIMAL POINT
1310 IF X$(1,1)="-" THEN K1=2 ELSE K1=1
1320 FOR K=K1 TO LEN(X$)\IF FNW(ASC(X$(K,K)),49,57) THEN EXIT 1340\NEXT K
1330 RETURN 0
1340 !CHR$(7), \RETURN 1
1350 FNEND
1360REM-----FND$
1370 DEF FND$(T$) \REM--DAY OF WEEK
1380 A$="SUNDAY MONDAY TUESDAY WEDNESDAYTHURSDAY FRIDAY SATURDAY "
1390 IF FNW(LEN(T$),6,6) THEN 1470
1400 T1=VAL(T$(1,2))\T2=VAL(T$(3,4))\T3=VAL(T$(5,6))
1410 IF FNW(T1,1,12) OR FNW(T2,1,31) OR FNW(T3,1,99) THEN 1470
1420 Y1=T3+1900\D=T2\M2=T1\IF M2>=3 THEN 1440 ELSE M2=M2+10\T3=T3-1\GOTO 1450
1430 M2=M2+1\T3=T3+1\GOTO 1450
1440 M2=M2-2
1450 D1=INT(2.6*M2-.2)+D+T3+INT(T3/4)\D1=D1+INT(19/4)-38\D1=D1-INT(D1/7)*7+1
1460 RETURN FNS$(A$,D1,9)
1470 RETURN "***ERROR**"
1480 FNEND
1490REM-----FNS$
1500 DEF FNS$(A$,J,L)=A$(L*(J-1)+1,L*J) \ REM--SUBSTRING EXTRACTION
1510REM-----FNC$
1520 DEF FNC$(R,C) \REM--CURSOR CONTROL
1530 R=INT(R)\C=INT(C)
1540 IF FNW(R,1,24) OR FNW(C,1,80) THEN 1570
1550 REM--THE NEXT LINE WILL VARY FOR DIFFERENT CRT'S. CHECK YOUR MANUAL.
1560 RETURN CHR$(27)+"="+CHR$(R+31)+CHR$(C+31)
1570 !CHR$(7),"ARG ERROR IN FNC$, R=",R,", C=",C
1580 STOP\RETURN ""
1590 FNEND

```



BASIC AND OTHER PROGRAMS TO CP/M

By George Eldredge

Before I got my SoHo Group Matchmaker program which has North Star BASIC on CP/M with the ability to go back and forth between ASCII format BASIC program and North Star format BASIC programs, I made a simple program to decode a Northstar BASIC program so that I could use it for a BASIC interpreter in CP/M. I also made a translator for my PDS Assembly Language programs and Editor programs going to Word Star. Here they are for your use.

A. BASIC to CP/M

There is a simple way to convert N* BASIC programs to CP/M format that can be edited by CP/M Edit, or in my case by Word Star. It may be wanted for compilation by a CP/M BASIC or just for inclusion in an article like this one. Table 1 is a listing of the Assembly Language program, CBASE, that will do it. The steps for using it for a BASIC program named NAME are listed in Fig. 1. Some garbage will be on the end of the file because of the lack of a CP/M end of file mark, Control Z.

Fig. 1

LOAD NAME	To get it in memory.
FILL 11539,0	To avoid the "PRESS RETURN" response at the end of each 24 lines.
BYE	To get back to DOS
GO CBASE	To patch the Output address at 2009H in DOS. CBASE will return to BASIC.
LIST	Will list the BASIC program and store it in memory at 100H.
Boot up CP/M	
SAVE 100 NAME	To make a CP/M file of the Program.

B. Assembly listings to CP/M

Program CBASE can be modified to program CASMB by a modification of line 40 in Table 1 from 2D04H to 2D00H. This will make a program to put a PDS Assembly listing into a CP/M file. This is the program that I used for these Tables. Appropriate changes should make it work for some other N* Assembler.

The directions for program CASMB are:

B	To return to DOS.
GO CASMB	To patch the return address of DOS. CASMB will return to Assembly.
A 0 6000	To start the Assembly and store the listing in memory at 100H.
Boot up CP/M	
SAVE 100 NAME	To put listing in a CP/M file.

C. EDIT files to CP/M files

My PDS Editor has its working file in memory at 3800H. Program DROP will drop it down to 100H for saving by CP/M. Since a large file may overlap DOS at

BASIC Continued

2000H, program DROP does not return to DOS, but just cycles. See the listing in Table 2. For WordStar I usually remove all single carriage returns as op-

posed to double returns marking paragraph ends and change all long strings of blanks to only two blanks.

Table 1. Program CBASE to move a N* BASIC program to CP/M

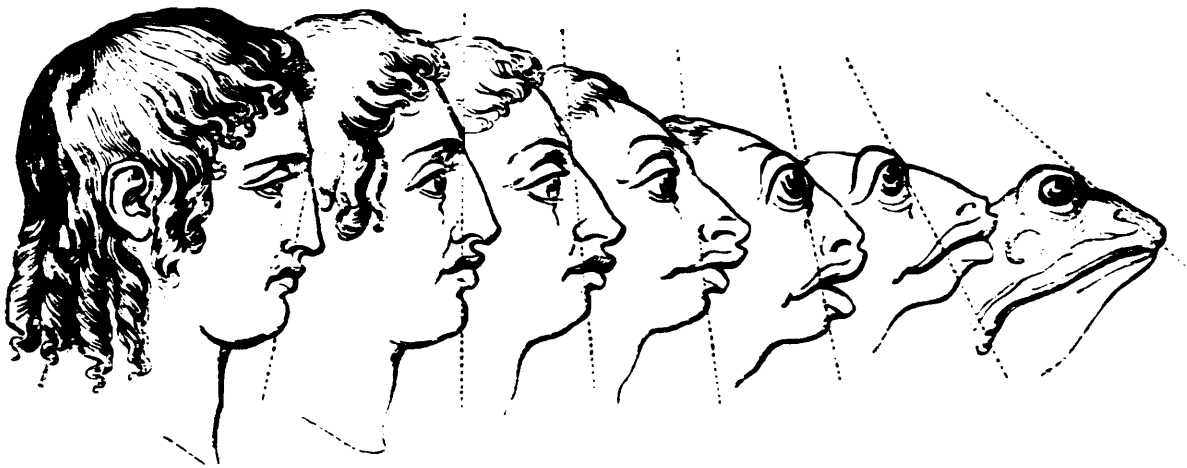
00		0030	DOS	EQU	2028H	
00		0040	BASIC	EQU	2D04H	To return to BASIC
00		0050	TPA	EQU	100H	where CPM files start
00		0060	LOC	EQU	20H	Skip some memory
00	C3 20 00	0070		JMP	LOC	
03		0080		ORG	LOC	
20	21 0E 20	0090		LXI	H,200EH	DOS output location
23	5E	0100		MOV	E,M	Save the address there
24	36 33	0110		MVI	M,33H	Address of PATCH
26	23	0120		INX	H	
27	56	0130		MOV	D,M	
28	36 00	0140		MVI	M,0	Part of the new address
2A	21 38 00	0150		LXI	H,PATCH+5	Use the saved address
2D	73	0170		MOV	M,E	
2E	23	0180		INX	H	
2F	72	0190		MOV	M,D	
30	C3 04 2D	0200		JMP	BASIC	The patch is finished
33	E5	0210	PATCH	PUSH	H	
34	C5	0220		PUSH	B	
35	D5	0230		PUSH	D	
36	F5	0240		PUSH	PSW	
37	CD 4A 29	0250		CALL	OFFFHH	Dummy address for out
3A	21 4F 00	0260		LXI	H,LABL	Pointer to memory starting at 100H
3D	5E	0270		MOV	E,M	Save pointer in DE
3E	23	0280		INX	H	
3F	56	0290		MOV	D,M	
40	EB	0300		XCHG		Pointer to H
41	70	0310		MOV	M,B	Put a char to memory
42	EB	0320		XCHG		Pointer saved in DE
43	13	0330		INX	D	
44	21 4F 00	0340		LXI	H,LABL	Pointer saved in LABL
47	73	0350		MOV	M,E	
48	23	0360		INX	H	
49	72	0370		MOV	M,D	
4A	F1	0380		POP	PSW	
4B	D1	0390		POP	D	
4C	C1	0400		POP	B	
4D	E1	0410		POP	H	
4E	C9	0420		RET		Return to DOS for next character
4F		0430	*			
4F	00 01	0440	LABL	DW	TPA	

Table 2 next pg.

Table 2. Program DROP to move a PDS EDIT text to CP/M

00 01 00 70	0030	ORG	0	
03 21 00 38	0040	LXI	B,7000H	Max text length
06 11 00 01	0050	LXI	H,3800H	Loc of EDIT text
09 ED B0	0060	LXI	D,100H	CP/M text loc
0B 00	0065	LDJR		Z80 command moves all
0C C3 0B 00	0070	NOP		
		JMP	HOLD	Loop till CP/M boot

APC BASIC



A NORTHSTAR-COMPATIBLE BASIC FORDOS AND CP/M

Review by Bob Stek

APCBASIC is advertised as the "Rolls Royce" of BASICs. I was a bit skeptical of such an obviously exaggerated claim. But since APCBASIC was touted as a NorthStar compatible BASIC for both DOS and CP/M, I asked for and received a review copy from John Cleckner of the American Planning Corporation, 4800 Duke Street, Suite 423, Alexandria, Virginia, 22304, (703) 751-2574. Well, it may not be a Rolls Royce, but it is certainly a Mercedes-Benz!

If you learned to program in BASIC using a NorthStar floppy disk system, you may have come to appreciate its multi-line

functions, its binary-coded decimal arithmetic, its ability to read files at the byte level, the optional floating point processor, and other handy features. But you may also have become disappointed that NorthStar has not upgraded its BASIC like Microsoft has upgraded MBASIC.

With very few exceptions, APCBASIC probably has all those "extra" features you have had on your "wish list" for a better NorthStar BASIC. Such features include: faster operation (usually 2 or 3 times faster!), ability to load or save programs as standard ASCII files, a global search and replace command for editing, a RENUMBER command which supports rearrangement of whole groups of lines as well as "simple" renumbering, an execut-

able MERGE command, a very sophisticated TRACE command (including a conditional TRACE), the ability to re-dimension arrays under program control, more flexible sub-string handling and extended string indexing, an ON <expression> RESTORE <line# list> statement, extended format specifications for PRINT formatting, ability to PRINT anything to a text file using 'devices' 8-15, a SWAP statement to exchange the contents between pairs of variables (including arrays), direct BIT manipulation, compound statements executed as a group within IF statements, multiple ranges within a FOR...NEXT loop (e.g., FOR X = 1 TO 10, 20 TO 100, 200 TO 1000), a WHILE loop construct, LOCAL variables within GOSUBs and FNs, a RENAME command, variable passing between CHAINED (LINKED in APCBASIC) programs, an executable DELETE statement, an extended machine language CALL statement, many new functions including CEIL(X), TRUNC(X), MOD(X), FRAC(X), MIN(X,Y,...), MAX(X,Y,...), a polynomial evaluation function, a TRIM\$(S\$) function to remove leading and trailing blanks, substring MATCH and FIND functions, a SPACE(X) function to return the amount of remaining disk space--to name but a few!

Admittedly this is a non-standard BASIC, but it is about 98% upward compatible from NorthStar BASIC. The author, Chris Cochran, has done an outstanding job of adding many valuable enhancements to BASIC, going beyond a "me too" version of MBASIC. APCBASIC is especially well suited for systems developers who want as many useful programming conveniences as can be reasonably provided in a language. However, it can also be appreciated by the "average" BASIC programmer who may be running into some of the limitations of BASIC as a

programming language but who really doesn't care to learn a new language such as Pascal, 'C', or PL/1.

DOCUMENTATION

In a word--excellent! This 136-page manual will not teach anyone how to program--it is meant to be a language reference manual only. If you already know how to program in BASIC, you can quickly become adept at APCBASIC by using the logically arranged Table of Contents which provides: an Introduction; the APC-BASIC Commands for program entry, storage and retrieval, editing, execution control and debugging; a section on numerics; one on strings; APCBASIC Statements; function library; miscellaneous information; utility programs; and finally a special section on APCBASIC for NorthStar BASIC users.

Even better, you can use the 17-page index (2 columns per page). Anyone who is familiar with micro software documentation has to be impressed by that alone! I did find some inconsistency in terms of examples provided, however. While many of the statements, commands, or functions did have concrete examples given, many others did not. This was especially missed when some of the more advanced features were described, even if the description and explanation were reasonably clear. More suggestions on typical uses of these more powerful features would be appreciated.

EASE OF USE

This is the area where APC-BASIC really shines. There are many "nice" features built into this interpreter--small things, perhaps, but once you have become accustomed to them, you find yourself annoyed that they

are not features of other languages.

For example, the SAVE command prompts you with a message telling you whether you are saving a new or an old file, requests confirmation of the SAVE (with a Y/N response), and then SAVES the file. If you enter SAVE without a file name, APCBASIC uses the file name previously specified in the last successful SAVE command. This feature makes program backup during development fast, repeatable, and mistake-proof. The EDIT command has a built-in string search facility. The command EDIT 150,A1\$ displays each line after line # 150 that contains the variable A1\$, allowing you to EDIT each one sequentially.

In addition, the search string may contain 'wild card' characters. If you want to perform a global search and replace, APCBASIC provides a CHANGE command which replaces one string with another everywhere or selectively within a range of line numbers. CHANGE will also request a response to its VERIFY? prompt to allow you to control each replacement as found. The REnumber command is very flexible: REN 500,5,900,999 will move all lines in the range 900-999 to 500, renumbering them by 5.

The facilities for debugging programs are much expanded. The TRACE command allows single-stepping through a program one line at a time, skipping detailed tracing of loops or sub-routines if you wish, and stopping program execution temporarily to allow for examination (or changing) of variable values before continuing. The TRACE may also be executed conditionally such as TRACE IF A > 50; this allows you to determine dynamically how and where erroneous values originate. Finally, TRACE allows you to direct its output to a device other



than the console by supplying the device number immediately after the TRACE keyword.

A control-C interrupt allows for direct execution of any executable program statement or line of statements (without exceeding its 159 character line limitation). That is, the entire contents of an array may be examined by entering a single line FOR...NEXT loop in the direct mode. And any alterations you make to the contents of a program variable will be carried over if you CONTINUE execution. Three utility programs are provided with APCBASIC: ZBIG, CRUNCH, and CONFIG.

ZBIG is a machine language cross-reference index generator for APCBASIC programs. It provides an alphabetized instant directory to all user defined functions, GOSUB's, variables, GOTO's, and other line referencing used in a BASIC program.

CRUNCH performs some simple optimization of a program by creating a new version without spaces and REMarks and by concatenating multiple lines into one line where possible.

CONFIG 'personalizes' APCBASIC in regards to floating point board address (if used), console backspace sequence, system interrupts flag, and control-C detection flag.

PERFORMANCE

APCBASIC does require a Z-80 processor, so some of its extra

speed can probably be attributed to judicious use of the Z-80's expanded instruction set. But it is obvious from bench mark testing that APCBASIC's author has a few tricks up his sleeve as well.

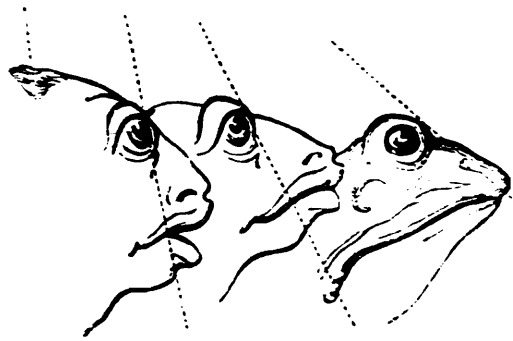
I compared APCBASIC with NorthStar BASIC and Micro Mike's BAZIC, both running under CP/M (APCBASIC under N* DOS gave similar results). I used three programs: a quicksort of 1000 random numbers, a chi-square test for randomness of 1000 random numbers, and a 'do nothing' program with several math functions, subroutine calls, and an IF statement within a FOR...NEXT loop. The results:

	N* BAS	BAZIC	APCBAS	APCRUN
1.	193'	121'	86'	65'
2.	138'	91'	67'	57'
3.	62'	52'	40'	39'

One further explanation: the APCRUN referenced above is a special 'run-time' version of APCBASIC furnished with the regular version. It is not used for development work as it does not include the interactive editing commands. It provides about 3600 bytes more free memory, automatically deletes all REM statements and extra blanks from the program, and runs up to 50% faster (depending upon the type of processing involved).

But it is apparent from the chart that for a 'real life' task (the sort), APCBASIC can be three times faster--certainly significant enough an improvement to make one consider the use of APCBASIC for many applications. And the availability of the run-time version can give the developer some security in the protection of source code.

There are some differences between APCBASIC and NorthStar BASIC; these are detailed in a special section of the manual. The most noticeable changes are very minor ones--some of the



keywords have changed (or are alternately defined). AUTO may be used, but the preferred command is now ENTER or ENT; CHAIN works but LINK does too; NOMARK replaces NOENDMARK; there are several others as well. These will not cause you to make any changes in your existing programs; the tokens used internally are the same but they will list differently under APCBASIC.

The only error I encountered in running previously written NorthStar BASIC programs was that APCBASIC would not allow two single-line user-defined functions on a single program line--they had to be placed on two separate lines.

EVALUATION

Overall I have to give very high marks to APCBASIC. It is now my standard NorthStar compatible BASIC for CP/M. I won't call it the Rolls Royce of BASICs because of some small oversights (some of which will be implemented in the next version, I have been told). The major oversight is that it still sticks with NorthStar's (and ANSI's) variable naming conventions; a new version will allow expanded variable names. In a BASIC with so many new functions, I was surprised that a SORT verb was not implemented.

The SWAP statement helps speed things up, but a simple in-memory SORT statement would be even better. And, like BAZIC, terminal customization for 'clear screen' and cursor posi-

tioning for a PRINT AT statement would be icing on the cake. And if you are going to provide a MIN and MAX function, then they should work on an array (ala APL) to return the minimum or maximum value in the array. But these are all really minor quibbles on a fine BASIC interpreter.

I would be remiss if I did not comment upon the price of APCBASIC --\$400. That seems a bit steep for the casual programmer though perhaps justifiable for a systems developer. My guess would be that APC could sell more than twice as many copies of their BASIC if they cut the price in half. \$200 for a NorthStar compatible BASIC with many significant improvements which operates under CP/M at 2 to 3 times the speed (maybe obviating the need for a companion compiler) would be much more attractive in my opinion.

All in all, APCBASIC is the BASIC interpreter that NorthStar BASIC should have evolved into. APC tells me that they have been in contact with NorthStar trying to interest them in providing APCBASIC with new NorthStar systems. As far as I am concerned, it would be a good move on NorthStar's part.

[In something like a record for speed for INSUA, we received and now reprint the following message from APC, written in response to Bob Stek's review, a copy of which he sent to APC -- Ed.]

The review of APCBASIC is thorough and accurate. One major improvement has been completed during the lag time between the review and publication. APCBASIC now runs on 16-bit, 8086/8 processors. This version of APCBASIC currently runs under CP/M 86 operating

systems and will run under MSDOS in April. The 16 bit version provides 62K for programs and an additional 63K for data. APC plans several improvements for this version and will issue updates as these improvements are completed.

APC is offering a special APCBASIC package at a special price to INSUA members. The standard \$400 APCBASIC package includes:

- Development interpreter--floating and nonfloating point
- Run-time interpreter--floating and nonfloating point
- Configuration program
- Library of 20-plus functions, including a 2000 record per minutesort and matrix multiplication/inversion routines
- Crunch program that scrambles and compacts source code
- Cross-reference generator that is extremely fast
- A 136-page reference manual with a 17-page index

APC offers the development version of APCBASIC and the reference manual for \$199 to INSUA members who order directly from APC and enclose payment. When ordering, please specify Horizon or Advantage, numerical precision (8,10,12, or 14), and operating system (DOS, GDOS, HDOS, CP/M). Contact:

American Planning Corporation
4600 Duke Street, Suite 423
Alexandria, VA 22304
(703) 751-2574



NEWS FROM CHICAGO

Chicago User Group Starts Newsletter

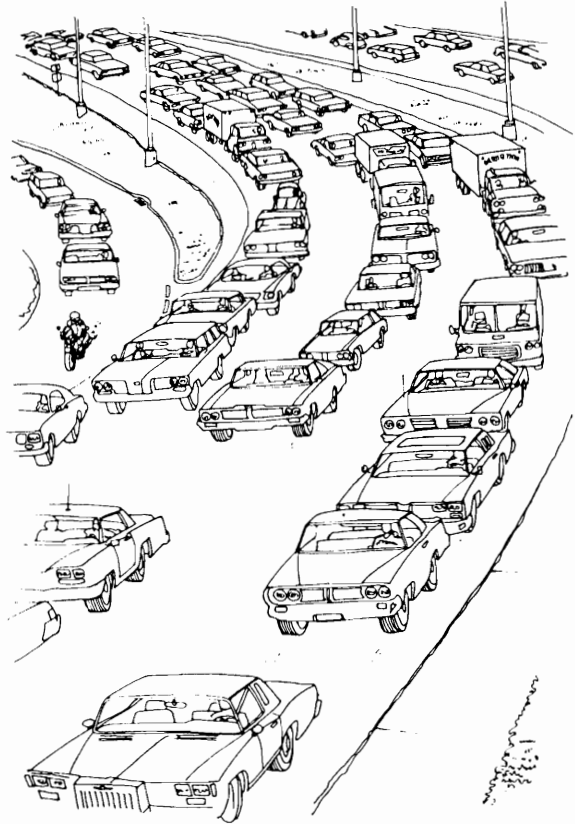
By Ed Coudal
627 S. Crescent Ave.
Park Ridge, IL 60068
(312) 823-3834

The Chicago Area North Star Users Group now has its own publication, **POLARIS NOTES**. Like Compass, the aim of CANSUG's publication is to give users of North Star Horizons and Advantages and other machines that are compatible with N* a medium for exchanging information and hints on the care and feeding of these machines. **POLARIS NOTES**, now a two-page newsletter, is published monthly, though the editor makes no promises on that score. Any readers of Compass can get on the **POLARIS NOTES** mailing list by joining CANSUG (\$5 a year) by writing to the editor, Ed Coudal (who is also the author of this article).

NOTE: Free exchange of newsletters with other local N* User Groups!

CANSUG meets before the Chicago Area Computer Hobbyists Exchange (CACHE) meeting on the **third Sunday** of each month. The CANSUG meet starts at 11 a.m., at Triton Community College, River Grove, IL (a Chicago suburb). A "technical" meeting also is held, at 1 p.m. on the **first Sunday** of each month in the home of one of the members. Current "officers" of CANSUG are Bob Fincutter, President; Tate Yoshida, Treasurer; Joe Alonso, Treasurer and Technical group program chairman; Steve Keith, Librarian; Ed Coudal, Editor.

THE CANSUG LIBRARY of common domain software is shaping up finally under the new librarian, Steve Keith. Firm details on



what's in there, how to get it, cost of copying, etc. can be had from Steve at either the monthly CANSUG meeting at CACHE or at the Technical meeting.

CANSUG MEMBER Steve Keith received nice mention in the March issue of Microsystems in a review of the DIRALPHA program which he has written. DIRALPHA not only alphabetizes the N* directory automatically, but closes it up, eliminating all those holes in the directory caused by the deletion of files. That means any new files saved to the disk are saved to the end of the directory, not stuck in somewhere among the files created last year. Another member, Joe Alonso, is quoted extensively in a major article on micro payroll applications in the March issue of Business Computer Systems, the new and excellent Cahners publication.

DYNACOMP, which seems to have the broadest library of N* software around, is including the following on a flyer with North Star formatted programs:

Older North Star disk systems and the North Star Horizon could read single density disks. The newer double density Horizons can read both double and single density disks, with one proviso: If you are running CP/M, the single density disk must be in drive B. The Advantage **cannot** read single-density disks.

ONE HEARS occasional rumors that North Star is in trouble, partly because of the company's lack of presence in the hobbyist magazine ad pages. In meetings for dealers around the country in the fourth quarter of 1983, North Star said:

- Sales had grown at a rate of 69% a year over the last 5 years.
- 1982 was expected to show sales of 120% over 1981.
- North Star now has 4% of the total micro market, and is targeting growth to 6% of the expanding market over the next 2 years.
- Loonim (Hamilton/Avnet) is now distributing the North Star ... one of the biggest and most respected distributors in the industry.
- There are some 44,000 North Star systems in the field.

...all of which doesn't sound too shabby.

PROGRAMMING QUICKIE (from the Washington State North Star group): If you want to delete from some line in a North Star BASIC program, say 240, to the end, you must know the ending line number and specify it in the DEL argument, i.e. DEL 240,3440. That means scrolling

through the program to find out what the last line number is. Instead, add a line number like 9999, **followed by a space**, that is higher than any in the program, then enter DEL 240,9999. Ends the bother of searching the end of the program.

ANOTHER PROGRAMMING quickie, this one from Microcomputing: You can speed BASIC programs by replacing "slow" operations with "fast" operations. B + B, for instance, computes faster than B * 2, yet they are mathematically equivalent. Similarly, X * .5 is faster than X / 2. Obviously, the time saved is unimportant in single instructions, but could become noticeable in long loops, for instance. Besides, such programming practice is just good technique.

THE 8/16 CO-PROCESSOR available for the Advantage carries an 8088 CPU which runs at 8 Mhz instead of the originally announced 5 Mhz. This is a significant competitive advantage (pun intended) over the IBM PC, which runs at 5 Mhz. The Advantage's Z80A processor runs at the standard 4 Mhz. North Star also announced a 64KB RAM board for the Advantage, with a retail price of \$349. There's room in there for 3 such boards, which when added to the 64KB already in place makes a 256KB Advantage possible.

Finally, the DEVIL'S DP Dictionary defines "firmware" as "a neutral, nowhere zone between hardware and software, free to deflect instructions, data and blame in either direction, and enabling problems to be solved with three sets of modifications instead of one."



IN MEMORIAM

THE NORTH STAR USERS GROUP

Founded 1977

Expired March 3, 1983

By John Schweppe
2011 Key Blvd
El Cerrito 830303



The North Star Users Group, Mountain View, CA, has met for the last time. During the March 3, 1983 meeting the members decided the time had come to disband. The group bequeathed its library of public domain software and the few dollars in its treasury to INSUA.

The North Star Users Group was formed shortly after the North Star Computer Company began delivering floppy disk systems in 1977. The company was one of the first to produce a S-100 disk controller for the new Shugart 5 1/4 inch minifloppy disk drives. The system came complete with DOS (disk operating system) and North Star Basic. North Star also offered the Horizon computer with the drive system. The hardware was very reliable and the software was easy to learn and use. It became the most popular 5 1/4 inch disk system and many suppliers furnished their software in North Star format.

When membership outgrew the living rooms of its members the group met at the Palo Alto Byte Shop and became known as the North Star Users Group of Palo Alto.

Members began to write their own software. A library of several hundred programs was collected and placed in the public domain for all to use. The Byte Shop allowed the members to use its computer to copy library programs--a vital service.

Later the club moved to the Digital Deli in Mountain View. Membership grew to about 150 and attendance was limited to the forty or so who could squeeze

into the Deli. The Deli has expanded but the users group has declined. Recently attendance has dropped to about 20.

Several factors are believed to have caused the decline. Many of the original members have gone on to become entrepreneurs in the computer industry and are just too busy for club meetings.

The membership is largely made up of people who have had a computer for several years. Their hardware is up and running; their software is sorted out and runs reliably. Now their interest is in adding printers and other things. These people are able to write software programs but good software is available in the market.

Most meetings are devoted to what is new in commercial hardware and software. The members have become computer users rather than computer builders or software programmers.

New members are a little lost when they come to a meeting. They have just bought a factory-assembled computer with software and are trying to learn to run the whole thing. They are not ready to think about added features and problems.

Recently the CP/M operating system has largely supplanted the North Star DOS. By installing CP/M the North Star user has access to a much larger body of software. Only a few members of the users group use the North Star DOS exclusively.

The North Star Computer Company has found the market in business computers more lucrative than hobby computers and has directed its efforts toward the business market. Their lat-

est computer, the Advantage, is a packaged ready-to-run computer complete with software aimed at the business market.

The final blow came when the Digital Deli decided to drop the North Star line. Each make of computer seems to have a different system of recording on floppy diskettes which is completely incompatible with any other system. The Deli no longer has a computer that will run North Star diskettes; therefore members can no longer copy programs from the club library.

Many members will join the PCNET club whose interest is in CP/M based software and is not tied to any particular make of

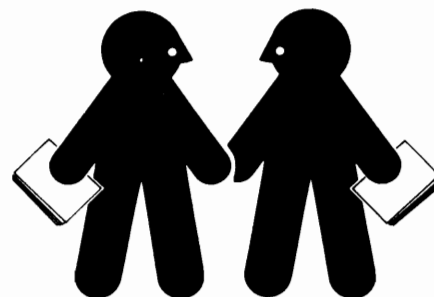
computer. The club meets at the Dysan Auditorium in Santa Clara on the second Thursday of each month at 7:30 PM.

The North Star Users Group is gone, but it will be remembered with nostalgia by many.

[Just to allay the misapprehensions of unwary readers, this article does not describe or portend the demise of INSUA-- though it is fair to say that there are probably lessons here for INSUA, for any North Star club, and for the North Star company itself. --Ed.]

COMMENTARY

By Charles W. Stevenson
849 Contra Costa Ave.
Berkeley CA 94707



I have several comments about items in Compass, Vol. II no. 4.

32K to 64K

Hooray for the Steve Leibson article on making a 64K board out of a 32K version. It is a project that I had contemplated but never had taken the time to investigate properly. The earlier article on converting 16K boards had caught my attention, but I don't have any. Now all I have to do is to buy the 4116s and then spend 20 minutes making the changes.

I believe that Steve's comment about having to deselect the top 8K holds only for boards of recent manufacture. Earlier double-density controller boards and 32K boards allowed for phantom on buss pin 67, but the wizards at North Star removed

that capability along with a number of the gold-plated pins when the price of gold went through the roof. The phantom line allows both RAM and controller to cohabit address space at E800H, but the RAM goes to sleep when addresses between E800H and EBFFH are generated. With the latest version of CP/M from North Star, you can take advantage of that capability. Also, their 64K boards allow you to deselect the 1K at E800H, but Steve's modification on older 32K boards gives me the best of both worlds, and cheaply. Thanks, Steve.

5 1/4" to 8"

The "North Star Answers" column gave me several concerns. The first one that bothered me

COMMENTARY

dealt with Robert Cowart's answer to the 8" to North Star disk transfer question. I have been running a system with North Star double density and a Morrow DJ-1 single density 8" controller for about two years. I bought the connector software from John Dvorak and it was the best \$60 I have ever spent on software. It installed easily and worked the first time and ever since. I can boot CP/M from North Star disks and then run the connector program. (I called mine DJ.) You must provide space in "safe" memory for DJ, about 256 bytes. The DJ-1 is memory mapped at E000H and does not handle the phantom line. (Has anyone modified a DJ-1 for this?) Contrary to what I said in the last paragraph, I deselected the top 8K and then added a Godbout 8K RAM board with only one bank turned on and addressed at F000H. This gives me 4K of "safe" memory.

Now, the problem I would have with a Morrow-generated package is that I like the Horizon to be in charge, not the DJ-1. The DJ-1 boot address is E000H so the jump address header on the Z80 board would have to be changed. Then, North Star DOS wouldn't boot. So, I would place the use of the Connector software as the number one choice. By the way, I believe that John Dvorak sold his software business before he became editor of InfoWorld so he would not be the current source of the Connector programs.

[John Dvorak **does** still sell his connector, plus a few other software items. His address is 704 Solano, Albany, CA 94706. --Ed.]

Nothing I have said above necessarily applies to the DJ-2, Morrow's double density controller. I don't have one and don't

know about its ins and outs. However, I believe that a Connector is also available for this board. Be sure you get the right one. Contacting Pavel Breder is the best way to make certain. [Pavel Breder may be contacted through INSUA. --Ed.] In fact, Pavel should have answered this question in Compass.

Compal-80

The next one dealt with a Compal-80. This machine was based on the Poly-88, but the ROM/RAM address was moved to E000H from the Poly standard 0000H address and other changes were made to make it more bomb-proof. It was originally a cassette system, but the first disk systems were North Star. However, Compal switched to Micropolis drives very soon thereafter and left the North Star contingent out in the cold.

Now, with the ROM/RAM taking the entire 4K at E000H, they had to move the North Star disk controller address somewhere else. They chose F400H (or F500H), and they moved the video RAM from its original location at 8800H to F800H. So, these boards, which were single density only, were very non-standard. I used to have one of these systems and still have access to one. I have a 5.2S DOS system that runs on it, as well as a version of CP/M 1.4. I never bought CP/M 2.2 for single density. Anyhow, I would like to assist the person who asked the question to get up to date. As I recall, there was a problem in saving files with their DOS Version 2 system, but I fixed that in my 5.2S system. If the Compal person, or anyone else with Compal/North Star problems, contacts me, I will try to get them running again for a nominal fee.

MIO/SIO

Finally, the MIO question.

COMMENTARY

It is true that there is a method of setting the SIO port on the MIO to look like the Horizon. The questioner didn't say if the board was known to be working at all, and jumpering this board is extremely complex. He also didn't say what version of CP/M he had bought, but it was probably 1.4. My advice to him is to upgrade to Lifeboat 2.2 (they still have an upgrade policy, I believe), which is much easier to install on a Horizon (and also on an MIO). Keep in mind that there is only one serial port and that the parallel port cannot be addressed at all like the Horizon. Port addresses on the MIO are in the range 0-3 only, and the control port serves the serial port, the parallel port, and the cassette port. Therefore, the busy/ready bits have to be al-

located to not conflict with each other. I do have a spare MIO manual which I would be willing to swap with Robert for a current set of Horizon docs.

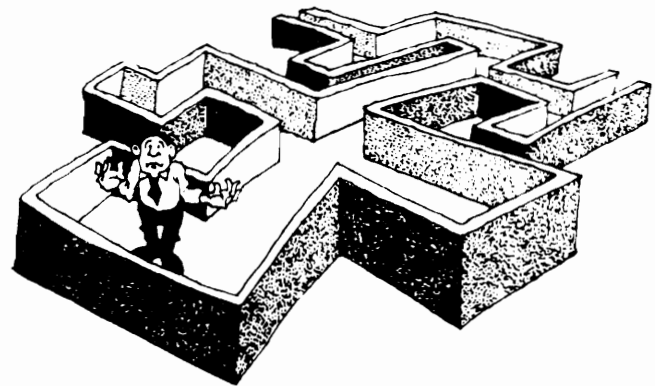
8085 to Z80

One last comment from the question/answer column in the newsprint issue of Compass. The person with the 8085 who wanted to run North Star Basic programs under CP/M should seriously consider swapping his CPU board for a Z80 so he can run BaZic. It works perfectly, even on such a complex program as Whatsit (remember that one?). Versions are provided that work with the floating point board, so you are not limited as you are with the InfoSoft package. The copy utility is very good.

GO FOR FORTY

By Siegmund F. Kluger
11025 Vista Del Sol #250-II
El Paso, TX 79935

I feel I must respond to a comment made by PM (Peter Midnight) in the Special Issue re: 40 tracks. I have been using 40 tracks (per side, on TANDON 100-2's) for 4 months now, with no ill effects. I know of a number of other users who have switched to 40 tracks per side without problems. Tracks 35-39 are perhaps considered unreliable by North Star because (as the rumors go) N* is buying drives which are not tested for inner-track reliability. So if you buy your drives from another source, you should be safe.



I have written a BIOS for CP/M+ that runs under the North Star DD controller, which I'm selling for \$40.00 per copy. I have made 40 track/side recording optional, so quad density users can get 50K more per disk. Especially with the large number of utilities under CP/M+, this helps quite a bit.

HARD DISK WITH ADVANTAGE

By Phil Rusin
Morning Star
962 Kaiser Rd.
Napa, CA 94558

The following is some rather important information for Advantage users, especially those running a Hard Disk.

For those using CBASIC on a HD Advantage, take care how you set up the block allocation when you create your CP/M logical drives. You can really speed things up by using the larger allocations.

While running a Manufacturing system on a HD, I saved 1-1/2 hours of the computer's time by using 16K blocks compared to 4K blocks. It took the Advantage 5 hours to do a complex file restructure. Now it takes less than 3-1/2 hours with the 16K block allocation, and the files are larger than they were before. It was wasting its time accessing the directory. Remember, if you use large blocks

like this and your files are small, you will waste disk space. So, take the time to decide what size of blocks you really need.

Here is a very nasty sort of problem I ran into on the same system. While adding data to these CBASIC files, the computer just crashed the files for no apparent reason. Getting RE errors was quite strange. It appeared that CRUN was missing the End of File marker. Very strange indeed! What finally did fix the crashes was that I started using CRUN 2.38 instead of CRUN2. It seems that CRUN2 did not like the large block allocations. CRUN2 is supposed to handle 500K files, but the crashes occurred on files less than 68K. So, make sure you use the newer version of CRUN with the HD.

I would like to hear from any other users who have encountered this problem or discovered a solution.

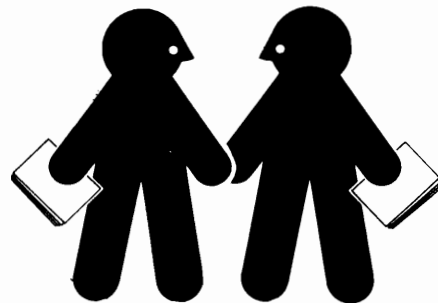
COMMENTARY

By James C. Matthews (Tiny)
Montgomery, AL

I received my first issue of Compass, and am very impressed by the quality of the articles. It reminds me of the early days of Doctor Dobbs Journal, before they went slick.

I did, however, notice some rumblings of possible future problems in the matter of articles concerning the heavily copyrighted and trademarked op-

erating systems from Digital Research. There are probably a hundred "operating systems" drifting around which are based on somebody's half-remembered ideas of what the DECSystem they used in school was like. They range from very very good (Heath's HDOS 2.0) to butchered



COMMENTARY

almost beyond recognition (CP/M).

Now, I realize that there are thousands of ignorant and gullible people out there who would insist that their computers come with a hand crank and manual spark lever, if BYTE magazine told them that was the latest way to go. For many others, though, one of the great appeals of North Star equipment is the fact tht we aren't crippled by total dependence on the worst system software in common use today, at least on micros (there are some REAL dogs on the big machines).

I have recently dropped my subscriptions to both DDJCC&O and Microsystems because of their growing obsession with CP/M, to the almost total exclusion of anything else. I don't really think that People's Computers intended to become unpaid agents for D. R., but it happened anyway.

You have got to be very careful about this. I would suggest a quota system, with at least 50% of the space devoted to "pure" North Star topics, and not more than 10 or 15% to any single other subject. A special precaution for CP/M, UNIX, and PASCAL articles would be to check the last 6 months or so of the "Big Slicks" for similar articles. One of the great

abilities of modern word processing is the creation of a dozen or so "original" articles from the same basic material.

If you have not already done so, it might be interesting to do a membership survey on what other magazines and newsletters the members read. I would guess about 70% for BYTE, maybe 30% for Microsystems, and 20% for DDJ and John Dvorak's flyer. A more interesting survey might be what (computer) magazines we have tried and dropped, and why!!

New subject: On the "ON" and "OFF" question, I think that there is less danger here than advertised. Heath specifically warns about this, but I normally leave a system disk in SYO of my H89 and nothing bad ever happened. For most of 1978 and 1979, my N* MDS had a decoding problem which would select drive 1 whenever an "=" was typed from the console, and randomly when running memory tests. This also turned out to be harmless. I suspect that some hotshot new Computer Owners tried to fire up their new toys without reading the books, and wiped out their distribution diskettes. Then, rather than admit the truth, they laid a song-and-dance on the vendor for a new copy, and this is how the story got started.

A/D

At the San Francisco Computer Faire we spotted Serial Data System's Das 16-4, an A/D (analogue to digital) interface device. SDS advertises the following features and applications:

Features:

Interface to Any Computer with RS-232

16 Channel CMOS A/D Converter
4 Latched Digital Outputs
Crystal-controlled Baud Rate
Low power consumption

Applications

Laboratory measurement and control
Remote data logging
Environmental monitoring
Process control

Description

The Das 16-4 is a single-board remote data acquisition and control sub-system with serial I/O. This board offers a low cost solution to users wanting to apply distributive processing concepts to data logging and control problems. The Das 16-4 board integrates an industry standard with CMOS A/D converter and UART with a three-wire RS-232 link. Optoisolators can optionally be put in the serial link to allow for large common-mode voltage differences. Throughput ranges from approximately 15 samples/sec at 300 baud to 400 samples/sec at 9600 baud.

Software for the Das 16-4 is extremely simple since no complex handshaking is required. To initiate an A/D conversion

the host computer sends an 8-bit byte to the Das 16-4. The four LSB bits select one-of-sixteen 0-5v analog inputs while the 4 MSB bits form the latched digital outputs. After a ~400µsec conversion time the 8-bit conversion is returned to the host computer over the serial link.

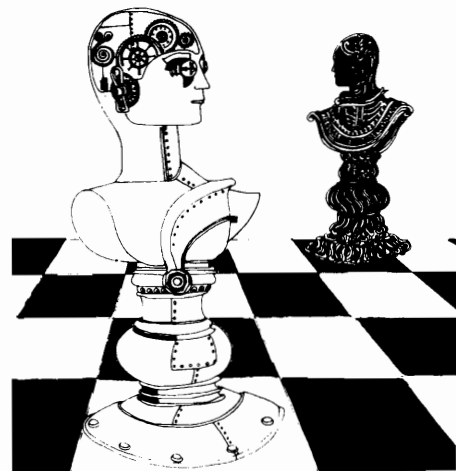
[The advertised price for the Das 16-4 is \$220--less than half the price of A/D S-100 boards we've seen advertised in the magazines. --Ed.]

Write to:

Serial Data Systems
22968 Victory Blvd., Suite #28
Woodland Hills, CA 91364
(213) 703-6390

WANTED

Original software package for Morrow DJ1 (with DISK-ATE, etc.). Someone who has "upgraded" to CP/M must have this around. Will answer all responses. James C. Matthews, 2028 Merrily Drive, Montgomery, AL 36111.



NORTH STAR MONITOR

5.2 MONITOR WITH ITS OWN I/O, ANYWHERE, THE EASY WAY

By Burt Andrews

[The following procedure essentially assumes a full 64K of installed RAM --Ed.]

It takes longer to describe than to do it, but follow these easy steps. When you're done, you will have a Monitor that will run with any DOS, or with none at all, and (if you've got it above the PROM) will even run when you're using CP/M.

1. Take a blank initialized disk. Do:

```
CR <*> 4 0
TY <*> 3
```

then use CF and copy the following files from the ORIGINAL 5.2 disk: DOS, M2A00, -DOS, -MONITOR, BASIC, MOVER. Now GO BASIC, LOAD MOVER, and RUN.

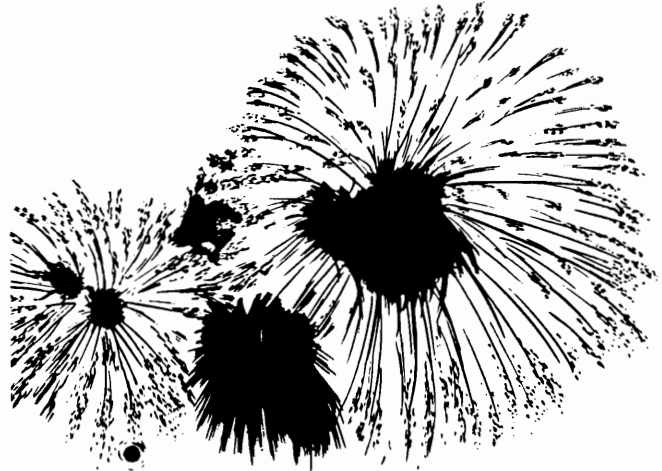
2. Decide where you want the final Monitor to run (I'll use F400H as an example). Then:

When asked for the Origin of new DOS, give a figure 100H less than that (i.e. F300H).

When asked if Utilities should follow DOS, type 'NO', and then when asked where they should be, give the right origin (i.e. F400H). (Yes, they overlap--that's the whole point!!!)

3. Sit back and wait. When you get a 'RELOCATION COMPLETED', the job is almost done. All that's left is to change the Jump Table at the very beginning of your new Monitor, and save it to disk. Do it thus:

Create a file to save it in,



e.g., (back in DOS):

```
CR MF400 9 (note--now nine
            blocks long)
TY MF400 1 F400
```

4. Now load the files into RAM, **IN THIS ORDER:**

```
LF -DOS F300
LF -MONITOR F400
```

5. The new Monitor is now in place (i.e. from F400 to FCFF).

6. Now 'GO' any handy Monitor and use its 'DS' command to change the Jump Table as in table 1.

7. Now do 'OS' and 'SF MF400 F400' and you're done. Test it with 'GO MF400'.

Since the N* I/O block is self-contained, it'll run anywhere you assemble it. All you have to do is CALL the routines. What the above process did was simply assemble a new I/O block to run at FC00-FCFF, lay a Monitor assembled to run at F400 in front of it, and change the Monitor Jump Table to use the new I/O instead of the one in your DOS.

There's a bug in 'MOVER': if you're running Single Density,

when it asks about your PROM's origin you must say it's NON-standard, and then give the standard origin in answer anyway. If not using SD, forget about this problem.

Don't forget to put the right DOS ENTRY address in the new Monitor Jump Table, else your next 'OS' command will leave you in the middle of nowhere.

Final words: there is a lot more I/O than a Monitor really needs. For example, TINIT isn't needed, nor would the lengthy Device Status routines seem required (compare the I/O block in

the earlier 5.0 M5700 Monitor, for instance). But there seem to be no ill effects from all the extra stuff, and perhaps it's an easier way to get yourself an independent Monitor than writing a new I/O for it. If you run into problems with this procedure, let me know, and if I can help, I will. [Tel. (301) 299-3469]

While all the above specifically relates to SD 5.2, a similar if not identical procedure should work for DD as well.

Table 1

F400	C3 13 F4		ok as is.
F403	C3 nn nn	to	F403 C3 DA FC the COUT routines.
F406	C3 nn nn	to	F406 C3 50 FC the CIN routines.
F409	C3 28 nn	to	F409 C3 28 20 your DOS hot entry.
F40C	C3 nn nn	to	F40C C3 F1 FC the CTL-C routines.

FASTSTEP

By Tim Deaton
 Central Computing Company
 Shelbyville, IN 46176

Here is a routine that members of INSUA may find helpful. The information is in the Horizon manual, but it is not really obvious.

Horizon DOS 5.2 is normally configured for SSDD drives without fast stepping motors. If you have fast-stepping (e.g. Tandon) drives and use DSDD, you will want DOS to provide appropriate fast-stepping instructions. To solve this problem, do the following:

Boot up the system. Follow the instructions in boldface:

```
NORTH STAR DOS 5.2 at 100
+LF DOS 5000
+GO M2D00 {any monitor but 5700}
MONITOR 5.2
>DS 134
0134=FF
>OS
+SF DOS 5000
```

To check your work, simply do the following:

```
+GO CD
COPY FROM DRIVE:1
TO DRIVE:2
COPY BOTH SIDES (Y OR N)? ***
```

***Hooray, you did it!

You should be able to tell the difference with fast-stepping drives immediately; you will also be able to read and write to both sides of the disk.

BUGS BUGS BUGS

Conversion Bugs

While preparing the disk for Compass, Vol. II, no. 3, we uncovered two errors in the listing of Jim Lind's "Conversions," in lines 100 and 130. Here is a corrected version, also available on the disk as a North Star BASIC program. --Ed.

```
10 DJMA$(26),B$(26),C$(26),D$(26)
20 O=0\F=0\E=1\RESTORE30\GOSUB80\IFOTHER20\RESTORE30\GOSUB110\GOTO20
30 DATA"MILLIMETER",3.9370079E-2,"CENTIMETER",.39370079,"METER",39.370079
40 DATA"KILOMETER",39370.079,"INCH",1,"FEET",12,"YARD",36,"FATHOM",72
50 DATA"FURLONG",7920,"MILE",63360,"NAUTICAL MILE",72913.386,"ROD",198
60 DATA"MIL",.001,"MICRON",3.9370079E-5,"ANGSTROM",3.9370079E-9,"HAND",4,"",0
70 OUT2,13\OUT2,27\OUT2,84\REM ERASE TO END OF LINE
80 INPUT"CONVERT WHAT ? ",D$\IFD$=""THENEND\N=VAL(D$)\M=1\B$=""
90 M=M+1\IFM>LEN(D$)THEN70\IFD$(M,M)=" "THENA$=D$(M+1)ELSE90\GOSUB140
100 READC$,A\IFA$=C$THEN140\IFC$<>" "THEN100\O=1\GOTO180
110 INPUT" TO WHAT ? ",B$\A$=B$\GOSUB140\IFB$(1,3)="SQU"THEN130
120 IFFTHENB$="SQUARE "+B$
130 READC$,B\IFA$=C$THEN180\IFC$<>" "THEN130\O=1\GOTO180
140 IFLLEN(A$)<6THEN150\IFA$(1,6)<>"SQUARE"THEN150\F=1\A$=A$(8)\E=2
150 L=LEN(A$)
160 IFA$(L)<>"S"THEN170\L=L-1\A$=A$(1,L)\IFA$(L-1,L)="HE"THENA$=A$(1,L-1)
170 IFA$="FOOT"THENA$="FEET"\RETURN
180 IFB$=""THENB$="ANYTHING"\IFOTHER190!D$," =",%C,A^E*N/B^E," ",B$\!\RETURN
190 !": CAN'T CONVERT ",D$," TO ",B$,".\!\RETURN
```

VENDORS' COLUMN

INSUA:

I enjoy reading Compass and would like to inform fellow members concerning some North Star compatible software we are marketing. We have two communications programs. One runs under North Star DOS 5.2DQ, and the other under Advantage GDOS. The cost of either version is \$49.95. We also have KEYMASTER, a utility for programming any key of the Advantage. KEYMASTER runs under CP/M 2.2 and costs \$59.95.

Patrick Corry
The Computer Shoppe
283 Medford Ave.
Patchogue, NY 11772
(516) 758-6558



VENDORS' COLUMN

Regina Dancy of D&A computers, 636 Rockaway Parkway, Brooklyn, NY 11236, announces a 10% discount to all INSUA members:

We analyze your business needs and choose the appropriate hardware and software for your company.

We do custom design software.

We have software packages available in most languages for:

GENERAL LEDGER
ACCOUNTS PAYABLE
ACCOUNTS RECEIVABLE
PAYROLL
ROUTING SYSTEMS
INVENTORY

10% DISCOUNT TO ALL INSUA
MEMBERS!!!

Perkel Software Systems
1452 North Clay
Springfield, MO 65802
(417) 862-9830

North Star Utilities

"The following utilities from Perkel Software Systems will run on the Horizon (SD or DD) or on the Advantage without modification. Written in Marx Forth, these utilities contain a unique module that finds DOS and reconfigures itself automatically to the host system. All disk utilities try at least twice as hard to read or write to the disk compared to the standard North Star utilities. These utilities are as easy to use as any other North Star system utility and are completely menu driven so as to be able to be used without having to go to a manual."

{DISKSAVE}

This utility is used to recover data by either rewriting a disk or copying a disk on a sector-by-sector basis. DISKSAVE tries 20 times as hard as DOS to read or write a disk, occasionally moving the head from side to side in order to read sectors written by a misaligned drive.

{COPIES}

This utility will make multiple copies of a source disk. To take full advantage of its capability, three or more drives are recommended. A source disk is inserted into drive 1 and up to three blank disks into drives 2, 3, and 4. COPIES will ask you how many copies you would like to make. After you answer, it will make up to three copies simultaneously. COPIES tries twice as hard to read and write DOS and does not quit after encountering a hard error. With these features, COPIES will outperform the CD utility of North Star even on a two-drive system.

{RENAME}

This utility is handy for changing the name, file type, and execution address of a directory. First you type in the old name. RENAME will find it and display information about the file. Then RENAME will ask if you would like to change the name and what to change it to. After this it will ask the same about the file type. If the file is changed to a type 1 file, RENAME will ask you if the execution address should be changed. After all the information is entered, the directory is updated.

{DSORT}

This utility is handy for sorting directories that get out of order as you create and delete files. DSORT will allow sorting either alphabetically or by the physical position on the drive. The directory density is maintained.

{DUMP}

This utility is a fancy HEX/-ASCII memory dump intended to be used primarily with a printer. The display is laid out for ease of finding bytes or text. Printer port is selectable for use with multiple printers. A great programmers' aid.

{SUPER COPY}

This is a multiple file copy utility. After inserting diskettes, SUPER COPY will ask which is the source and destination drive. Then SUPER COPY will read the source drive directory and one by one ask if you want to copy each file. All answers are one

keystroke (Y/N) for convenience. After all files have been presented, SUPER COPY will begin the copying process. It will display the file name being copied. If it is a new file it will create it on the receiving disk, changing density if you wish. It will also indicate whether it was a new or existing file. If a hard disk error is encountered, SUPER COPY will keep on going. The copying process continues until all files requested are copied.

Disksave \$35
Copies \$45
Rename \$30
Dsort \$30
Dump \$25
Supercopy \$45

Disk with all six utilities is \$150.00

Mark Perkel announces that he is writing a modem program, and asks any INSUA members or North Star users interested in modem programs to inform him of the features they would like to see in such a program.

LETTERS TO THE EDITOR

INSUA:

Recently I bought an Advantage with 2 5-1/4" disk units and CP/M Graphics as well as G-DOS and G-BASIC. At the moment I have no software at all so I have to start from the beginning.

I am personally interested in business applications for my Advantage and of course sometimes my little children also want to play with the computer

or even want to learn through the computer.

It will be highly appreciated if one or two members want to correspond with me and want to exchange views, ideas, etc. If you could provide me with names it would be great.

Yours faithfully,
T. Koorevarr
336, Veenoord
3079 NM Rotterdam
Netherlands

LETTERS



[INSUA can't directly provide names, but this letter may attract some potential correspondants. Also, see the membership list in Compass, Vol. II, no. 3. --Ed.]

INSUA:

I bought my Horizon with 64K (2 32K boards) and CP/M for maximum versatility. Not long ago, the computer started to "lock up" occasionally and had to be reset. Usually, the condition manifested itself while I was using WordStar, but it also occasionally occurred while running BASIC under NorthStar DOS.

The LEDs on the memory boards and the RAMTEST utility identified a bad memory chip. The particular chip involved was an outdated device that is not readily available. I exchanged this chip with one which was located in the area of memory deselected because of the disk-controller and have not had any recurrence of the problem. It seems that NorthStar has kindly included eight spare memory chips on the board that the computer ignores!

Sincerely,
Larry Feigen
New Orleans, LA

INSUA:

Why does my Diablo 1640 hang up when running WordStar at 1200 baud (options set: no protocol, LST:), but it will run at 600 baud? The printer runs at 1200 baud otherwise. The only modification to the Diablo was made when I first got my system a couple of years ago. At that time the Diablo would overload on nearly every baud.

Someone at NorthStar knew someone at Diablo who said to remove the entire cover and on the right end of the front PC board was a black plug; move the black wire on the black plug up to the open slot next to the purple wire. It worked! I've had no trouble until WordStar, which won't work at all on anything above 600. I was never able to get Mince's Amethyst to work either except on the "plain" or "vanilla" options. It failed to recognize carriage returns and printed garbage. Got any suggestions?

Yours,
Nelson Lucas
Long Beach, CA

[Unless handshaking is arranged between a computer and a printer, problems may well occur over 300 baud. A printer is largely a mechanical device which cannot hope to keep up with the electronic transfer of data. Usually one of two bad things occurs: either the printer buffer overflows, resulting in error signals and loss of text; or the printer will type out the first section of text, but will then stop ("hang up").

The first situation occurs because the buffer overflows, is flushed out, accepts a new batch of characters, overflows, etc. etc. The second problem occurs because the computer sends a batch of characters ending with a special character, then pauses to wait for a "goahead" signal back from the printer before sending out another batch: but the computer fails to detect the "goahead" message, doesn't send out a new batch of characters--and thus waits forever.

Handshaking can be implemented in hardware (for some printers) or software. Bob Cowart

LETTERS TO THE EDITOR

swears by hardware--see Compass, special issue. Your editor mostly uses software--see "Hints for Installing Wordstar including Diablo Printers at 1200 baud" in Compass, Vol. II, no. 3. This preference is due to the fact that many printers (including some Diablo's) aren't equipped for hardware handshaking.

Your specific problem may be that WordStar has I/O routines of its own, which you should try to configure correctly. To repeat the simple instructions for the Diablo, use WordStar's INSTALL, choosing the "N" and "D" options in the first steps; select the printer option (Diablo 1640 in your case); choose the ETX/ACK option; then WordStar's Port Driver. Now shut your eyes and hit the Carriage Return again and again until WordStar comes up of its own accord. --Ed.]

INSUA:

There is only one thing lonelier than a Maytag repairman and that is the owner of a What??? computer. For this reason, I was extremely happy to hear from you.

I have owned a Horizon for about four years and an Advantage for about four months. Except for dealer support, I feel that I have the very best. Now that I will be a member of INSUA, my operation will be even better. I trade commodities and write my own trading system programs. I also publish a traders' newsletter, which is produced on the Advantage with an Epson printer. A copy of this program is available to any Advantage-equipped INSUA member.

I appreciate the work done by

your members and look forward to contributing to the effort.

Sincerely,
B.A. Thunman
4550 North 38th Street
Augusta MI 49012

INSUA:

My Compass usually comes badly chewed up in the mail. Can we get card stock covers?

Jeff Dixon
Arvada, CO

[We're looking into new ideas for printing and packaging Compass. The idea of heavier cover sheets or card stock covers is an interesting one, and we thank you for the suggestion. Meanwhile, INSUA will replace any issue which has been damaged in the mail. But it may cost you (and INSUA) less if you merely xerox your damaged cover sheets (if they're not entirely destroyed), and use the copies in place of the originals. --Ed.]

INSUA:

Saul Levy's comments (Compass, Vol. II, no. 4) re: my previous Compass item "After-crash" (Vol. II, no. 2) are absolutely correct and I did a rather poor job of explaining why I've found my "Reset-to-Monitor" button to be such a useful device.

While it is true that re-booting DOS does not disturb either BASIC or the BASIC program it was running, to be able to re-boot just to DOS implies two things--

1. You are not using one of the

* * *



very convenient features of the NorthStar package--its automatic GO-LOAD-RUN capability, and

2. You are willing to swap disks in drive #1 after every crash.

Since nearly all of my disks AUTO-GO something, and since I hate the routine of swapping disks, pushing a "Reset-to-Monitor" button has been a very convenient alternative that was easy to set up. To ensure having a Monitor to jump to, all of my systems disks (and many of my applications and development disks) hold a Monitor with its own I/O block and a GO address at F400H. [See Burt Andrews' Monitor article elsewhere in this issue. -Ed.] When any of these disks are booted, a short machine language program first loads the Monitor. Then it "keys in" whatever sequence of instructions I would otherwise have entered from the keyboard to use the disk. Finally, since I always have a Monitor out of the way up at F400H, it makes it easy to prowl through RAM and look at what's been going on.

The scheme is surely nothing that everyone would need, but I have found it very useful for my purposes. My apologies for not explaining it better the first time.

Sincerely,
Burt Andrews

INSUA:

I read with great interest Alan Nelson's article in the Special Issue on interfacing the NorthStar to the Sony Typecorder. I was especially intrigued by the last paragraph statement that he has tried this approach with the HP 75C. Since I am one of the owners of this portable I would be very interested in

knowing the method you have adopted to interface it with the NorthStar.

Very truly yours,
Norman L. Bloch
Chicago, IL

[I have had my hands on the HP75C, but I have not tried to interface it to NorthStar. If you can locate the output lines, connect one to pin 2 of the second North Star port, the other to pin 3. Make certain baud rates match; then try receiving the transmission with a modem program like modem7. If nothing happens, try reversing the leads. If nothing happens now, try a dealer or HP. Please report successes or failures to Compass. --AHN]

INSUA:

A method to produce a mosaic pattern on N* Advantge:

When running WordStar press Control-P then F15 Key (backspace-delete) and zap! you are off to never-never land with a nice mosaic pattern on your CRT screen. Haven't had time to investigate why. Perhaps some other Compass reader has.

Bob Smith
Oak Harbor, WA

INSUA:

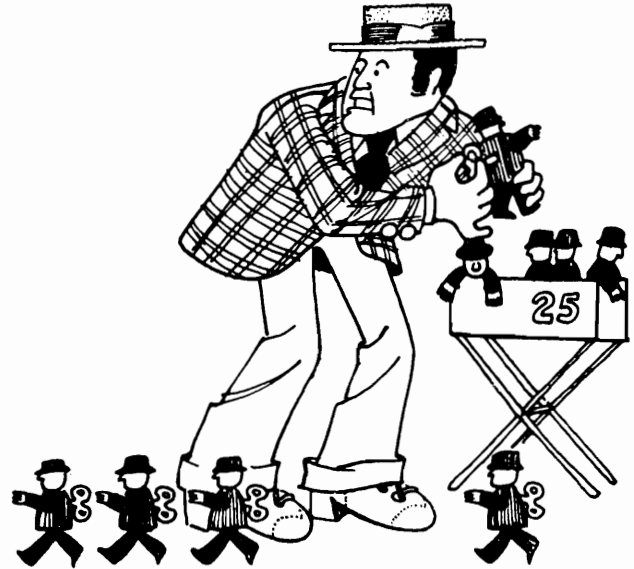
Is there anyone out there in North-Star land who is still trying to run PASCAL (I.5)?

Donald M. Waller
Department of Botany
Universiy of Wisconsin
430 Lincoln Drive
Madison, WI 53706

INSUA BOARD OF DIRECTORS 1983

The following were elected to the INSUA Board of Directors by votes cast either by mail or at the San Francisco Computer Faire, March 19, 1983. At the first Board meeting, March 25, the Board members were elected to the designated offices.

Bob Beaver, President
Clyde Steiner, Vice President
Sarah Wasserman, Secretary-Treasurer.
George Riddle, Membership
Mark Sheppard, Disk Librarian
Larry Hamelin, At Large
Alan Nelson, Editor, Compass



NORTH STAR ANSWERS

by
Robert Cowart
Technical Support Services
NorthStar Computers, Inc.

Hello there again. Another issue has come and gone, and of course many more questions have come in through the post, some of which have even been generated by my previous answers (a bad portent I fear). Anyway, first off, a brief update on things at NorthStar.

The new NorthStar manufacturing facility in Cork, Ireland announced that its first Advantages rolled off the assembly line this month. The Irish facility will handle all manufacturing for the European market, as well as providing a second source for all the components of the Advantage. The plant is located in an expandable 40,000 square foot facility in Kilbarry Industrial Park, and will employ 300 people. According to NorthStar's VP of Operations, Bruce McKay, "The European market is critical to

NorthStar's long term marketing plan. With our new plant, we are establishing a presence in Europe that will allow us to serve that major market much more effectively with the assistance of TRW Datacom."

The NorthStar NorthNet, a proprietary local area networking (LAN) system is in the final stages of testing at N* headquarters in San Leandro, Ca. The system will network up to 64 Advantages as well as allowing use of shared peripherals and hard-disk drives. All menus as well as the electronic mail and sign-on screens utilize the graphics capabilities of the Advantage. There are also plans to allow connection of Horizons to the system, as well as the ability to link (via NorthLink software) to IBM and IBM-emulating mainframes. The NorthNet will support the CP/M operating system to begin with, later expanding to include NorthStar's TBS (Total Business Solutions) operating system and possibly N* DOS as well.

NorthStar Answers

Finally, a new graphics package (which was demonstrated at INSUA's annual business meeting during the 8th Annual Computer Faire in San Francisco) is also in final development. The Graphics Family, as it is being called, will include BusiGraph II and Imagemaker in its first incarnation. BusiGraph II is a much expanded version of the program included on the Advantage Demo-Diagnostic disk, and Imagemaker is a true graphics editor which allows drawing of many shapes and types of letters on your Advantage with subsequent editing and scaling, rotation, etc. Data files from CP/M programs like dBASE II may be pulled into BusiGraph II and plotted. Hard copy of your screen image is possible on Hewlett-Packard, Houston Instrument, and Nicolet plotters, as well as on some dot matrix printers.

Well, so much for the news...on to the tips for the month. Since printer interfacing is always a popular issue, let's start there.

Printer Technical Tips.

Installation of the Diablo model 630 printer for use with NorthStar's Enhanced Wordstar can be accomplished by installing as a 1610/1620, and modifying the Diablo supplied interface cable as follows: 1) Desolder the wire going to pin 11 on the printer end of the cable, insulate and tie back, (this wire will not be used). 2) Desolder the wire going to pin 20 on the printer end of the cable and connect it to pin 11. 3). If you wish to "buzz" the modified cable to be sure the modification was performed correctly, you should have pin to pin continuity on pins 1 through 8, and continuity from the NS side pin 20 to Diablo pin 11. All other pins are "don't care."

When installing the printer driver during the WordStar installation, select NO PROTOCOL and CP/M LST: device.

To interface the Integral Data System (IDS) Microprism 480 parallel printer to the NS Advantage Computer PIO interface: Construct an interface cable with pin to pin connections as follows:

ADV	IDS
5	14
12	13
4	12
11	11
10	10
2	9
1	16
8	3
7	22
13	7

The typical installation problem on the NEC 3510 printer is buffer overflow. To prevent this, NS recommends hardware handshaking by modifying the NEC 3510 supplied interface as follows:

- 1) Desolder the wire going to pin 19 on the printer end of the cable, insulate and tie back, (this wire will not be used).
- 2) Desolder the wire going to pin 20 on the printer end of the cable and connect it to pin 19.
- 3). If you wish to "buzz" the modified cable to be sure the modification was performed correctly, you should have pin to pin continuity on pins 1 through 8, and continuity from the NS side pin 20 to NEC pin 11. All other pins are "don't care".

NOTE: If you purchased your 3510 from NS, this modification has already been done.

The front panel of the NEC 3510 containing four switch assemblies and two thumbwheels. To operate correctly with NS software all switches should be set as follows from left to right:

1-8 off, 1-4 off, 1-4 off, 1-3 off, 4 on. The thumbwheels should be set at 5 and 66. This sets up the printer for 1200 baud, NO parity, 6 lines per inch, and 10 characters per inch. In addition, if you change any of the switches or the thumbwheel settings the printer must be turned off and on before it will recognize the new settings.

Epson notes:

The serial interface boards currently being shipped by EPSON for their MX 800/100 printers are different than the ones which N* uses. N* has shipped model number 8145 and 8151. The new EPSON board is called the 8151, or the GOLD EAGLE board. For use with the Advantage, set jumpers as follows:

- J6 IN (9600 baud)
- J11 IN
- J14 IN
- J18 IN (out for MX-80)
- J19 IN (These set up for
- J20 IN hardware handshaking)

All other jumpers should not be connected.

- S1 - self test must be off.
- S2 - must be on for serial connection.

* NorthStar system software for the Advantage assumes that any serial printer connected is running at 9600 baud. CPMGEN allows you to modify the printer baud rate for CP/M applications. SYSGEN (a BASIC program not to be confused with the CP/M program of the same name) allows you to modify the baud rate for GDOS and GHDOS applications. ASP and HDSO have printer selection menus which ask you to specify the baud rate.

CP/M notes:

The GCP/M (graphics CP/M) Pre-

face, pg 6-25 makes reference to error codes that are returned from the GRAPHICS MANAGER subsystem during character I/O. Nowhere else in the documentation is any further reference to these error codes made. Here is a complete list for you graphics users:

- @GNOER == 0 ; NO ERROR
- @GVPDE == 1 ; VIEWPORT DIMENSION ERROR
- @GGPCDE== 2 ; POLYGON COMPLEXITY ERROR
- @GCNPE == 3 ; CHAR NOT PRINTABLE
- @GCOSE == 4 ; CHAR OFF SCREEN
- @GNSRE == OFFH; NO SUCH ROUTINE

HDOS notes:

The factory master HDOS 2.0 shipped with N* hard disk systems assumes 56K of contiguous RAM from 00 to DFFF. HDOS may appear to load successfully into less RAM than this, but problems will occur when trying to load and run application programs. HDOS will work with a minimum of 48K RAM but required a patch to do so. The patch is as follows:

1. Boot up a non-write protected copy of your factory Master HDOS.
2. Type: DS 013C <return> ;the system should respond "013C= "
3. Type: C0 <return> ;for 48Ksystem.
4. Type GO HBASIC <return>; loads HBASIC
5. Type LOAD SYSGEN,1 ;loads SYSGEN from floppy drive 1.
6. Type RUN<return> ;runs SYSGEN.

During program execution, SYSGEN will report that you have 48K. This is the indication that you have patched things correctly. Upon completion of SYSGEN, you will have a 48K HDOS system.

HD-5 notes:

When running diagnostics on or formatting an HD-5 or 15 (either Advantage or Horizon), the following symptoms indicate a reversed 34 pin cable from the drive controller to the drive:

1. Level one test fails to run at all.

2. Level 2 test reports a 113 error and then requests level 3 test to be run.

3. Level 3 reports a 113 error followed by continuous 120 errors.

If these are your symptoms, reverse the large connector and try again.

HD-18 notes:

For all HD-18 owners operating under dusty environments, we suggest cleaning the timing wheel located under the Seal Assembly once a month. This is not an easy task, but if you look on the under side of the sealed unit inside the cabinet, there is a small brass or plastic wheel with little slots in it. This turns along with the disk, and is used for sector timing. If some of these little slots get dust on them, the electronics have a hard time figuring out which sector is being read. Regular cleaning will prevent this. Please refer to your dealer if you do not have the HD-18 technical manual, or prefer not to undertake this kind of thing yourself.

HDOS users:

There is a new version of HDOS out. It is rev. 2.1.0. If you have the new copy and are wondering how to get it onto your hard-disk read on: 1) Boot up your new HDOS 2.1.0 system disk. 2) Select the appropriate Hard Disk Operating System, wither HD5,15, or 18. 3) When the HDOS "=" prompt appears, enter the following commands:

GO SHORTCUT,1 <return>

(The question "are you sure?" will appear. Answer YES. Commands will appear on the screen which will delete and change the type of several files. Some files may not be found. This is OK.)

4) GO TOTREC,1 <return>

(After completion of Shortcut, perform a TOTREC with one very important difference: answer **NO** to the first question, "Is this what you want?". This is the key. **IF YOU ANSWER "YES", YOU WILL WIPE OUT YOUR DATA !!.**)

5) At the completion of TOTREC you will have updated your HDOS to 2.1.0 and maintained the integrity of all your old application and data files.

If you have a new or totally backed up Hard Disk, then do not use the old diagnostics programs (format and HDtests) in preparation for loading the new HDOS. Instead, run the new diagnostics and format under the new HDOS 2.1.0 and run TOTREC answering "YES" to the "Is this what you want?" question.

*

*

OK. Now that I have all that out of my system, on with the REAL questions...

Q:...I do agree that I can't understand the bulk of what you talk about in the COMPASS, and I wish I could. It may be a good idea to put in different sections for the real beginner and the people who already have a good knowledge of computers...I read books and most of them are repeats of the first book I read. I want to know what you are talking about. -M.P. Chehalis, Washington.

A: An appropriate question since

the last 27 paragraphs were all technical. Sorry, but I had to do it as there were some important facts to get out. We are (I am) working on making this column more varied in its range of technicality. The board does realize that an increasing numbers of readers 1) are NOT hobbyists, and 2) are Advantage owners. Therefore, an attempt is being made to solicit less technical articles especially any which deal with the Advantage.

Q: I have a defective memory board I need repaired. Can you help me?

-M.G. Livingston, N.J.

A: Try calling Frank Kohzad at 415-276-5245. He is a factory trained technician knowledgeable in all aspects of Horizon repair, drive alignments etc.

Q: WordStar incorporates single line headers. My work, as a technical writer for Qume requires at least three header lines. Is there any way I can do that with my current WordStar 3.0? If not, do you have access to an update which does allow multiple line headers? Are you aware of a programmer who can and will modify the program? WordStar allows long cursor movements within the document via the FIND command. However, that can be, and is often, confusing. Is there a way to move the cursor to a particular page in the text? -A.L.P. San Jose, Ca.

A: As far as I can find out, the answer is no to both of these. If anyone out there knows differently, please let me know c/o INSUA. For the headers, you could use ^KR from a three line file which has your header on it.

Q: Does the EPSON 2K buffer RS-232 interface always crop the 1st character in a catalog listing from DOS @ 9600 baud, or is this an isolated problem (mine)? -R.A. Davenport, IA.

A: Sounds like your problem. Be sure you are set for 1 start bit, 2 stop bits, NO parity, 8 data bits. Make a fresh copy of your DOS from a known good source. Try your printer on another system if possible.

Q: I use an Advantage for editing mainly and would like to set up a variety of headline fonts in some of my articles. Font programs exist for some of the more popular computers, but I have not been able to locate one that will produce large characters within Wordstar. Can you help? -P.W.W. Manchester, MO.

A: The only program which I know of for this is called **fancy Font**. It requires an Epsom 80 or 100 with Graftrax and claims to work with Wordstar. I have not tried it, but I think they have a demo package so that you can try before you buy. You will have to look through some of the magazines for the address, etc. as I don't have it. Sorry.

Q: I would love to connect my Advantage to a large-screen television set to display graphics as well as charts and so forth for business meetings at my company. Is there any way to do this?

M.S. -Menlo Park, Ca.

A: I have been informed of a product marketed by a company in Redwood City, California which is supposed to do just what you need. You see, this really isn't a simple matter, as there is no "composite video" output available inside the Advantage. So

you need a circuit which will combine the horizontal sync., vertical sync. and video signals. Again, I have not tried it, but give them a call. It's **Television Systems Service, 415-365-8137**. Ask for Grayson Gibbs. The price is \$159.00 for the Interface, and \$89.00 for the cables. I have phoned them and verified that the product is available.

Q: I am a new member of INSUA and am a business-person. I need elementary school guidance when it comes to computers! Steer me to some good books, and I will learn in time. - T.K. Annapolis, MD.

A: I could recommend some, but they would be the wrong ones most likely, because my interests are probably different than yours. Your best bet is to hop on down to your local computer store and peruse what's on the shelf. If you can understand it, then it's a good bet. Make sure there is a little bit of Greek included so that you learn something new though. I found the **CP/M Primer** as well as the **CP/M USER GUIDE** by Thom Hogan to be of use. There are more than enough books to choose from with nearly 2,500 titles, accounting for more than 4 million dollars in sales this year. Nearly 3 million Americans bought computers last year with as many as 6 million expected to do so this year. It is estimated that each computer owner will purchase up to 10 books per year, so you can imagine what a plethora of styles and topics you may run into. Before you buy, skim the table of contents, make sure there is an index (if it is a reference book), look over one of the chapters and then check the price tag. If it seems a bit steep, maybe your public library has a copy!

Q: I would like to know more about the 8/16 bit board for the Advantage. Does this board require an additional 64K of RAM for full operation? What can you tell me about the different types of repair/warranty services which I can resort to if I decide to buy a NorthStar? As I have a physical disability, I am not physically able to take my machine in for repair. Therefore it is important for me to know about the reliability of the NorthStar Advantage. I am concerned about NorthStar switching from an S-100 bus system to an S-40. No one has been able to clearly explain the advantages and disadvantages of each. I would appreciate any clarification you could provide on this point.

A: When running MS-DOS, the operating system is resident on the 8/16 board. This takes up 32K of RAM, leaving only 32K for programs. This is just enough for the BASIC-16 compiler. Even the 8088 assembler will not fit in this space. Therefore we do recommend you use 128K, which is enough for any programs I know of. Regarding service, a new NorthStar computer is warranted to be free of defects for a period of 90 days from our day of purchase. You may send it to us (through your dealer, as you need a return authorization) and we will repair it and send it back to you post-paid. More appropriate to your situation since you have a physical disability, is the on-site Sorbus Service Contract. For an annual cost of about 10% of the machine, Sorbus will make on-site service calls to repair your machine at your location. Sorbus also offers a carry-in service option, but you must deliver and pick up. Of course, this is the less expensive alternative. Regarding reliability, the Ad-

vantage has a, very good track record. The factory claims that fewer than 5% of these units shipped arrive with problems. As for your last question, the decision to use a 40 pin bus in the Advantage is not a statement of non-support for the S-100. In fact NorthStar is very committed to the S-100 Bus and plans to continue development in that direction. The proprietary bus NorthStar used was simply a space saving innovation necessary for a desk-top sized computer.

Q: I have some programs on 8" disk which I need transferred to 5.25" N* format. Do you know of anyone who could do this for me?

I do not want to have to purchase the equipment you mentioned in the last issue.

A: You can call the following number and leave a message for John, indicating what you want done and your phone number of course. He will get back to you. The charge is \$15 per disk, not including the disk. The number is 415-540-6667.

Well, goodbye, and keep the letters coming. Your NorthStar correspondant, R.C.

P.S. I **AM** doing my best to answer all of the letters I receive. If you haven't seen the one you sent, It's just a matter of time until yours is answered.



PRESS RELEASE

Computers in Health Care '83: A symposium and Exhibition will be held at the Red Lion Inn, Sacramento, CA. September 29 - October 2, 1983.

The Eskaton Health Corporation, the University of San Francisco, and several major health professional associations will offer a 3 1/2 day symposium and exhibition. The event is designed for health care professionals including hospital administrators, physicians, dentists, pharmacists, pharmaceutical and equipment supply vendors, health and human services agency staff members.

More than 20 workshops and 150 exhibitors will be featured from September 29 - October 2, 1983. Topics, exhibits and demonstrations will include computer legal and purchasing considerations; data and word processing; administrative forecasting and problem solving; client billing systems; diagnostics; educational software; biomedical engineering; accounting systems; and case management.

Contact: Mary Tuma McAdams or
Irene Elson
916/927-3480 or
916/927-5722

Insua

**Publishers of the Compass Newsletter
International NorthStar Users Association
PO Box 2910 • Fairfield, CA 94533**

**Bulk Rate
U.S. Postage**

PAID

**Walnut Creek, CA
Permit No. 203**

TATE 276G
JAMES TATE
23914 SPRING DAY LN.
SPRING, TX 77373

POSTMASTER: ADDRESS CORRECTION REQUESTED
Please enter key number on form 3579

THE COMPASS NEWSLETTER

INTERNATIONAL NORTH STAR USERS ASSOCIATION

VOLUME III, NO. 3



EDITOR'S COLUMN

The following letter touches on many topics which have been close to your Editor's heart as he has prepared this issue:

* * *

INSUA:

I wish I lived closer so that I could be more involved with INSUA. Failing that I have some suggestions that I think could make for a more responsive organization.

A note in the Compass about INSUA's executives struck a cord with me: "please remember that we are all volunteers." We are, most of us, members of several organizations and I think the tendency is to get overcommitted. I now take the attitude that I will not accept "being a volunteer" as an excuse for work not done. I make this clear at the beginning and tell people not to volunteer for more than they can handle AND ENJOY. If a job becomes too big then we share it out among several members.

I have written a couple of articles for Compass and several letters. I have only received a short note in reply (once) stating that the Editor was an academic and did not have much time available but would be replying within one week (that was a couple of months ago). The Editor has the perfect right to use, edit, or discard all material and should use this authority to the full but I do feel that a column in Compass just to state "Letters and articles received from etc." could at least let members know that their efforts were not lost in the mail. I am not asking for a mailed reply (at today's postage rates) as everyone should realize that all material cannot be used.

Let us include next year's ballot with Compass to save on postage and envelopes. Also a short blurb by each nominee stating their intentions would be appreciated.

Please steer Compass away from using the bulk of a whole issue for a specialized item (e.g. using WordStar with

Epson). Many of us look forward with anticipation to the next issue of Compass and it is a major disappointment if 80% of an issue is of no use to us.

Above all I want to thank the executive members who have given so much time and effort towards an organization that I would like to join for yet another year.

Thanks for listening,
Michael O'Byrne
Ottawa, Canada

* * *

Your Editor's only defense against the charges is to repeat the old excuses: he is a volunteer, he is an academic with teaching and research commitments, he does bite off more than he can chew. In fact, he spent the period from 15 March to 24 August out of the country on a research project. Copy for Vol. III, no. 2 was prepared before that time, although the issue was not mailed until June. There was no way around this problem except to get a new Editor--and none was found lurking in the bushes.

Your Editor's feeling, to repeat an old tune, is that INSUA has its house in order in almost all respects, but that it has not yet mastered the art of responding to inquiries on technical and editorial matters. (Memberships and disk orders are dealt with quickly, I think, but through paid helpers.) As it is, many if not most technical inquiries end up in the Editor's hands, where they are disposed of according to his limited resources, often simply ending up in the LETTERS TO THE EDITOR column, with or without an answer; three or four months may pass before letters are published (and thus, in a sense, answered).

As for articles, many actually originate as letters explaining how the writer has dealt successfully with this or that problem. Other articles are submitted as such, and ought to be acknowledged--here Mr. O'Byrne has an entirely valid point.

A major problem for the Editor is that most submissions, whether notes or articles, come in typewritten, or printout

form, and must be typed into the computer to be used--this is a task undertaken by the Editor himself, as he has no paid secretary.

This is as good an occasion as any for the Editor to plead with INSUA members to submit articles and longer notes on disk as well as on hardcopy. It would be most helpful if material could be sent following the specifications outlined below. The Editor, you see, has only two programs which allow him to read disks as they were written, i.e. BASIC from North Star DOS, and WordStar from CP/M. He has, to be sure, used DOS monitors or POWER! to read other formats directly from disks, and has found:

Some editors (i.e. wordprocessing programs) write material to the disk in out-of-sequence blocks, e.g. 12563478 etc.

Some editors intersperse lines of text with lines of control codes.

Some editors use Carriage Return codes without New Line codes.

Some editors use (to me) incomprehensible codes for centering, indenting, etc.

Some INSUA members have quad drives, and send material which is written on the back of the disk; for your Editor to read this with his single-sided drives, he would have to be able to read binary code backward.

INSUA members please follow instructions below!

The Board of Directors is considering a change of format for Compass, which, if all goals were met, would result in more material on all subjects (including DOS), more interchange between manufacturers and users, and more professional responses to inquiries and submissions. INSUA members will discover the results (if any) of the Board's deliberations in 1984; meanwhile we will redouble our efforts to respond to inquiries, and promise to deal more quickly and more decisively with all material submitted to Compass. To all who have been disappointed with INSUA's responses, our sincere apologies.

--Your EDITOR

* * * * *

RULES FOR SUBMISSION OF ARTICLES

All articles and notes more than one half page in length should be submitted **on disk** (or eventually via modem). Material submitted in North Star BASIC or in WordStar (CP/M) may be submitted on disk alone. All other material should be accompanied by printed copy, with a note naming or describing the text editor which generated the file. Material which can eventually be transmitted by MODEM7 may be submitted on printed copy alone.

(These rules are not intended to inhibit submissions by persons who would encounter undue problems in conforming to them; the Editor will do his best to accommodate all submissions.)

DISCLAIMER

Programs printed in Compass and/or distributed through the INSUA disk library are offered to INSUA members in good faith. INSUA, however, is unable to guarantee the operation of any of these programs or to guarantee support. Users are advised to test the programs thoroughly for themselves in conditions under which they are to be used. Users

who employ such programs in serious business or financial applications must do so at their own risk.

Facts or opinions published about manufacturers and dealers, and all opinions expressed in articles and letters, are the responsibility of the authors, and not of INSUA or the Editor of Compass. INSUA offers the right of reply to members and non-members alike.

A LOOK AT SOME FEATURES IN NORTH STAR BASIC

By Edgar F. Coudal

There are many accomplished programmers among us who write their grocery lists in Assembler, multiply in binary in their heads, know instantly the hex value of 3147 decimal, and understand the difference between octal and split-octal.

If you are one of those, go away. Read the next article. Kiss your mate. Leave us alone. We're going to talk a little about programming in North Star BASIC.

A couple of Compass issues back, someone asked for more material for the beginner, or least the user who is not as capable as those mentioned in the first paragraph.

Like many, I felt a bit guilty about programming only in BASIC...and that self-taught, largely out of the SoftDoc book and through fooling around with existing games. I sheepishly watched the arguments over BCDS vs. Whitesmith's vs. Aztec C on the bulletin boards. My, how they carried on! I was fascinated and bewildered by the contest between Green, Orange and Purple Ada. My heart sank a bit when I learned that Pascal was becoming the "language of the universities." Lisp, I was told, is the language of the Artificial Intelligence community...and shouldn't all we computer users be striving toward that?! Serious computerists use FORTRAN and COBOL unthinkingly the way I use Crest.

Ah, me! There I was, crippled and brain-damaged beyond recovery, as E. Dijkstra said, by my early exposure to BASIC.

But then I realized that many of my programming needs were being fulfilled by programs written in BASIC...some of which I had developed myself. These included the mail list/data base program that handles hundreds of magazines for my public rela-

tions clients and also does the rough media scheduling for my advertising clients, as well as the income/expense bookkeeper that tells me how Coudal & Associates, Ltd., my marketing communications firm, is doing.

I felt better.

And the more I thought about it, the more strongly I realized that for my purposes, BASIC is probably all I need. That led me a bit deeper into North Star BASIC, which many say is a superior version of the language. I know that I certainly prefer it to MBASIC and CBASIC, which I have played with a bit, but I admit that is a purely subjective judgement. Bob Stek has written a review of American Planning Corporation BASIC in Compass, Vol. III, no. 3. I also have APCBASIC and agree with Bob that it may be the ultimate BASIC; APC says that its BASIC is rooted deeply in North Star BASIC.

With all that preamble out of the way, let's get down to it. I am going to dissect the program that accompanies this article line-by-line and try to show what each line does, including some elementary concepts and perhaps some more sophisticated ones. Hopefully, there may be a little something for everyone.

The program we'll look at is one written in response to the classical problem posed in many computer programming classes:

Write a program that will determine the minimum number of coins (half-dollars, quarters, dimes, nickels and pennies) necessary to make up any monetary amount given.

First of all, this program is not the **most efficient** or **most elegant** (whatever the hell that means) or **fastest** program that could be written to solve the problem posed above. In fact, it is clumsy in places just to show off some capabilities of North Star BASIC. On the other hand, it does have some nice touches. One is that

the algorithm is fully **recursive**, which is something high level languages have difficulty in achieving. Another nice touch is some good programming techniques...perhaps my only good programming techniques...but you'll see what I mean.

Let's go through it a line a time, so you can see what is going on. I'll just jibber along here with my thoughts.

LINE 100 and LINE 105 Always start your programs with a couple of REM lines telling what you're setting out to do, who you are, and when you did it. There's nothing worse than coming on a fascinating little piece of code you wrote six months ago and not having the foggiest idea what it's supposed to do. REMark every little jot, tidbit and tad of code with a purpose and a date. You'll love me for this suggestion. Also, RENumber everything at least by 10s before you NSAVE or SAVE. That provides space for code you forgot to put in (such as line 105 in this example). That's one example of good programming technique.

LINE 110 In North Star BASIC, you **don't** have to DIMension one-dimensional arrays that will use 10 or fewer indices. North Star BASIC will do it for you, setting aside 10 indices and initializing each of the array variables with a zero. However, it's a good programming habit to DIMension every array, even small one-dimensional arrays. It makes it look like you were planning ahead. In this case, where we'll only be using four indices, it saves space in memory. To prove this to yourself, after you've typed in the program, run the program with and without LINE 110, checking each run with the direct command !FREE(0) and note just how much space you've saved. With memory cheap, it doesn't seem like a big deal, but it can be vitally important in bigger programs. I wrote a Sherlock Holmes adventure in APCBASIC, setting out a number of 10x99 string arrays (you can do that very easily in APCBASIC, which remedies a major failing of North* BASIC). I soon ran out of memory on my 64K Horizon. By going back and looking at the space actually used by the arrays, I was able to reclaim almost 4K of RAM by simply reDIMensioning the arrays to their smaller and accurate sizes. Needless to add, LINE 110 was not put in this program

until after the algorithm had been worked out and I knew what size array I would need.

LINE 120 You don't have to know anything about DEFined functions. You never have to use them. You can get the same effect with a GOSUB to a line of code that performs the calculation and RETURN. You also can just write the whole calculation into the program every time you need it. This particular DEFined FuNction is really simple. All it gives you is the integer value (stripping off the decimal values) of A1 divided by A. I had a lot of trouble with DEFined FuNctions and still, in all honesty, do not really understand the multiple line variety. But there is a certain power in the DEFined FuNction. You are literally altering the "personality" of the machine when you create one of these little gems. You are in effect adding a special and unique function--not unlike SIN or ARC or some of those other exotic functions--to the inherent instruction set of the machine. That's control.

In using DEFined functions, start thinking about them in the middle i.e. **with the equal sign**. To the left goes the stuff you want the computer to recognize as a DEFined FuNction. To the right goes the stuff the computer has to figure out.

Let's look at the left side of the equal sign first. You start a one-line DEFined FuNction with DEF. That's like hitting a mule in the forehead with a board, or offering the blonde in the bar \$100. It gets their full attention. DEF tells the computer that what comes next on the line is a DEFined FuNction. FNA immediately following tells the computer that in the future when it sees FNA, it should scramble around in its memory for whatever is on the right side of the equal sign. Immediately following the FNA come the "arguments" in parentheses. These are the values that FNA will look up and use on the right side of the equal sign when the time comes to calculate. In our case, **(A1,A)** tells the computer that we want it to use the values of A1 and A in computing the overall value of FNA. We hope it ignores B and C and all the other variables that are running around.

Now for the right side of the equal sign. The variable FNA will take the value of whatever calculations take place over

there on the right side of the equal sign. You can make that calculation incredibly complex, or extremely simple, as in our case. The point is that every time FNA is recognized by the computer, it will perform all the calculations on the right side of the equal sign and assign the results of that calculation to FNA.

That's simple. Ask someone else about multi-line DEFs. [See Burdene Orris's article on this subject in Compass, Vol. III, no. 2. -Ed.]

LINE 130 What can one say? The machine has to have something to work with. The input line asks for the monetary amount. By inserting a string value (the constant "AMOUNT?" enclosed in quote marks), you prevent the computer from printing its own question mark. A1 is a variable, or identifier, or whatever you want to call it. It's going to change later in this program, which is the point of "recursion", but for the moment it has whatever numeric value you type in at the keyboard. A1 in fact becomes the "recursive variable" that makes this whole thing work. For purposes of example, let's say you INPUT .99. It could just as easily be .34 or 6.56 or 9078 or 9078.51. But let's use 99 cents for the example, INPUT as .99.

LINE 140 That's just a blank line on your screen. It makes the eventual output neater. It was added after the whole program was written and the first runs of the output showed that everything was squashed together. Making the screen appearance easy to read and even esthetically pleasing is worth trying for, even if it just means adding a blank line. By the way, the symbol "!" and the word "PRINT" mean exactly the same thing. You don't save any memory space by typing a ! instead of PRINT because North Star BASIC converts both the one-character ! and the five-letter word PRINT to a one-byte "token" (respectively 92H and 82H and 146T and 130T) as it interprets the line of BASIC code. You don't have to know all that stuff. North Star BASIC knows it, so why should you? Just use the ! It's faster to type and takes up less space in the program line.

LINE 150 This is another example of good programming practice. This program, which determines the minimum number of coins necessary to make up an input dollar amount, uses this line to discourage the smart aleck who tries to put in a number smaller than 1 cent. Someone will always try a negative number or a 0. When that happens, the program, with line 150, simply returns to the INPUT question. It's patient. It will do this all night if somebody keeps putting in -99 or 0 or .001. Recognizing and trapping illogical or improper input and giving the user a second chance to do things right is the single friendliest thing a programmer can do for the user. It's arrogant to say, when an untrapped program bombs out to the familiar READY, "the user should have known better..."

LINE 160 The INPUT statement allowed the user to put in something like .09 or 5.66, which is normal monetary notation. However, the computer works best with whole numbers...Thus we multiply the input by 100 to get rid of the decimal points. It can be done a number of other ways, but this is straightforward and only takes a few bytes. It makes the later algorithm easier and the final output clean and direct. Wouldn't you trade a line of code for that?

LINE 170 This is the DATA line that sets out the half-dollars (50), quarters (25), dimes and nickels. The three 0s and the 99 are just checks for the program to find out when it has tested all the possible coins. When the program eventually sees that 99, it will stop its computation and go to the output portion where it will display the result. I deliberately left out the Susan B. Anthony dollar coin because I kept mistaking it for a quarter in poker games. I hate it. The penny is treated as a "remainder" at the end of computation as you'll see and thus is not part of the data list.

LINE 180 The main body of the program begins here. This is the "algorithm," which is just a fancy word for formula.

(There is no such thing as an "algorasm" which has been defined as a brief and fleeting moment of intense pleasure experienced by the machine.) A computational loop is set up by the FOR statement. Since we are going to test the total INPUT back there at line 130 for four coins, we ask for four iterations...B=1 to 4. This is also pretty straightforward stuff, but one must realize that iteration--doing things over and over again--is the heart of computer programming. Computers are pretty stupid machines, but they really enjoy totally mindless, dull and repetitive tasks such as repeating computations and they do them very well.

LINE 190 Here, the algorithm reads the data contained in line 170. The variable A gets the first number in the line, which is 50, and the variable C gets the next number, which is 0. Anytime you see a line that starts with DATA, you should find another line somewhere in the program that starts with READ. If you don't, you might as well kill the line that starts with DATA because it's absolutely useless without the READ line. (The converse is not true; you can have READ lines without DATA lines. Treat those apparently isolated READ lines carefully and with great respect. They probably make the program work.)

LINE 200 Now we're getting down to business. The conditional test contained in an IF-THEN statement is a powerful programming tool, and becomes even more so when supplemented by an ELSE. You are telling the computer that IF some condition is met THEN do thus-and-so, otherwise go about your normal business, which is usually executing the next line of code. When you use the IF-THEN-ELSE construction, you are telling it the same thing, except that the ELSE can tell it to go off somewhere other than next line of code and do something. North Star BASIC also gives you the ability to make other IF-THEN statements the target of a "successful" IF, as this line of code does. First, LINE 200 checks to see IF the input total is smaller than the first coin it is testing, the 50-cent piece. Since the .99

assigned to variable A1 and then multiplied by 100 in line 160 is larger than the 50 READ into variable A, the program stops reading this LINE 200 immediately after IF A1<A and "falls through" to LINE 210. However, if the value of A1 had been smaller than A, the algorithm would not try to divide the total (A1) by the coin (A). Instead, the algorithm would go on in LINE 200 to find out what to do since the first IF test (IF A1<A) was "successful." The next IF-THEN in line 200 determines if the program had reached the end of the data...whether it had tested all four coins in the DATA line. If the value of C is 99, indicating all the values had been tested, the program will jump out of the current loop, which was set up in line 180, and go directly to printing the results. That part of the program begins at LINE 250. Finally, if all the data has not been tested (indicated by C being something other than 99), the ELSE in this line directs the machine to go back to the top of the loop (LINE 190, NEXT B) and perform the test again, this time reading the next two values (25 and 0) into A and C respectively. The computer has a "pointer" inside somewhere (I picture it as one of those rubber-tipped hickory sticks that nuns beat me with, though sometimes it acts more like a middle finger extended from a closed fist) that keeps track of where it is in a DATA line. Use of complex IF-THEN-ELSE statements gives the programmer incredible control over the flow of the program and uses the machine's inherent ability to make logical decisions to its fullest.

LINE 210 We're getting to the heart of things, so pay attention. The array that we DIMensioned back in LINE 110 and the DEFined FuNction that we invented in LINE 120 now come into play. The FNA has the value of 1 (99 divided by 50 and then stripped of the decimal part of the answer). That value of FNA (1) is stored in array element A(B). The variable B, which is the FOR-NEXT loop counter, is thus serving two purposes: it is keeping track of the number of iterations we asked for in the loop and it is serving as an indexing device for the array A() where

we are storing the number of individual coins. This is typical and routine use of the FOR-NEXT counter in dealing with indexing of arrays. You could complicate your life by using a separate variable as the array index and adding one to it every time the loop was executed, but what's point of that? Keep it simple.

LINE 220 The T in this line is for Total, the total number of coins identified by the algorithm. The number of half-dollars or quarters or dimes or nickels found by the algorithm is collected in this variable. Incidentally, the form of this simple statement, $T=T+\text{something}$, is the single biggest stumbling block for adults coming into programming for the first time, simply because $T=T+\text{something}$ is blatant nonsense in the world of mathematics. There, $T=T$ and nothing else. The problem is with the "=", which really doesn't mean "equals" but instead means "represents" or something like that. (And no, I don't like alternatives such as "==" that you have to use in other languages). I once questioned a 12-year-old who was being introduced to BASIC about $T=T+\text{something}$ and she thought a minute and said, "Of course it does. Otherwise the program wouldn't do anything."

LINE 230 This is the heart of the recursive nature of the program. In recursive programs, the problem redefines and simplifies itself, an apparent impossibility, until the lowest "level", which is usually the easiest problem to solve, is reached. In the case of this program that "level" will be reached when we redefine the A1 number to a number smaller than the smallest coin (5) that we have in the data line. Then we'll just have pennies left, which will be counted individually instead of being massaged by the DEFINED FuNction. To understand this line, look at the code to the right of the minus sign. We are subtracting from A1 the number of coins found on this trip through the loop, $FNA(A1,A)$, times the value of the coin itself, A. When this amount--the amount to the right of the minus sign--is subtracted from the current value of A1, you have a new A1, a smaller A1, to deal with. That, friends, is recursion.

LINE 240 What's this! A second NEXT B! There was one up there in LINE 200. You have heard over and over that for every MAN there's a WOMAN, that for every DROP OF RAIN that falls a FLOWER grows, and that for every FOR there must be a NEXT. Not so! The truth of the matter is that you must visit FORs and NEXTs that control the same loop alternately, but they don't necessarily have to be the same FOR or NEXT. If you try to visit the same FOR or NEXT twice in a row, without an intervening appearance by its partner, you will get the dreaded CONTROL STACK ERROR (which somehow is very forbidding, but certainly not as threatening as the FATAL ERROR #39 or CATASTROPHIC ERROR that other BASICs present). This concept is a bit tricky and is even considered cheating in some circles, but it works fine. Just watch it closely if you decide to use it. You don't need the B in NEXT B but it's good programming practice to use it so you can trace the loop more easily when de-bugging. NEXT B sends you back to the next iteration of the loop, unless that was indeed the fourth iteration, in which case you will fall through to the next line.

LINES 250-310 We are simply displaying the answer. The total number of halves, quarters, dimes, and nickels were captured in T and whatever is left at the lowest level of the recursion, A1, represents the pennies needed. Thus the total number of coins needed to make \$.99 is $T+A1$, or 8. LINE 260 is simply a blank line like LINE 140 to make the display more attractive and easier to read. LINES 270 to 300 display the contents of the array elements that correspond to 50, 25, 10, and 5. And LINE 310 displays the final recursive value of A1, the pennies. It happens that .99 which we selected uses at least one of every coin available except the nickel. For the nickel, LINE 300 displayed the initialized value of the array element A(4) which is 0, the right number! Amazing! North Star BASIC knew that part of the answer even before the problem was presented to it!

* * *

```

100 REM PROGRAM DETERMINES MINIMUM NUMBER OF COINS TO MAKE A TOTAL
105 REM BY EDGAR F. COUDAL PARK RIDGE, IL 4.9.83
110 DIM A(4)
120 DEF FNA(A1,A)=INT(A1/A)
130 INPUT "AMOUNT? ",A1
140 !
150 IF A1<.01 THEN 130
160 A1=A1*100
170 DATA 50,0,25,0,10,0,5,99
180 FOR B=1 TO 4
190   READ A,C
200   IF A1<A THEN IF C=99 THEN 250 ELSE NEXT B
210     A(F)=FNA(A1,A)
220     T=T+A(B)
230     A1=A1-FNA(A1,A)*A
240 NEXT B
250 !"COINS NEEDED:",T+A1
260 !
270 !"HALF-DOLLARS:",A(1)
280 !"  QUARTERS:",A(2)
290 !"    DIMES:",A(3)
300 !"    NICKELS:",A(4)
310 !"    PENNIES:",A1

```

Please, don't write or call and give me a "better" program for solving the problem. I know there are better programs. This one was written to illustrate some BASIC concepts, especially recursion, DEFINED FUNCTIONS, cheating with FOR-NEXT statements, array handling, and especially some good programming practices. However, if I have muddied things more than clarified them, or if I have misstated something, please let me know and I'll try to get it right in a future article.

* * *

Now for a couple of bad programming practices that were employed in this example. It is rather gauche to jump out of a loop the way we did in LINE 200 with the "THEN 250" part of the line. Gauche, but not really harmful in this case because the loop has been completed anyway if that logic is triggered. Better practice would be to make it THEN EXIT 250. The EXIT warns the machine that the program is finished with the loop and the machine then cleans up all remaining parts of the loop that are lying around in there (sweeps them under the power supply where no one will notice) and releases the memory used for keeping track of the loop for better purposes. Use of the EXIT is absolutely necessary if there is another loop later in the program, or if you are trying to make an early departure from a "nested" loop.

A second bad programming practice in this example is that you have to type in RUN and press return to make it work a second time. You really should have some lines after 310 that ask if you would like another run. If the answer is yes, then the program should initialize all the variables to 0 and leap back to LINE 130, asking for a new number to calculate.

And finally, a few random thoughts on the program in general. You'll note that in this entire program, only one statement per line is used, and the lines are littered with spaces. North Star BASIC lets you pack many statements to a line, separated by backslashes, without any spaces. That really saves memory space. Savings of 30% are common. As a test, this program was compressed and the REMARKS and blanks removed; its size was reduced from the present 528 bytes to 306 bytes, a 42% saving in space, and it runs faster. However, if you have the memory space to write programs one statement to a line, loaded with spaces, they certainly become easier to understand.

Also, the indentations in LINES 190-230 are a way of showing that this is the body of a loop, which also makes understanding the program easier.

Last, I mentioned "de-bugging"...correction of the program you have written so it will run. Do you know of any human endeavor other than computer programming where you fully, confidently, and enjoyably expect to **fail** the first time?

#

BYTE THAT FYLE

By Perry Garst

North Star BASIC can store three types of data in files: numbers, strings, and bytes. Sometimes this byte ability is overlooked but it can be useful for packing files with data in a minimum space. A variable, preceded by an ampersand (&), can be written to or read from a file and can have a value between 0 and 255. This is not too exciting itself but if a second byte is used to represent multiples of 256, the two bytes can have any value between 0 and 65535. This is the number of memory locations that can be directly addressed by a microprocessor with 16 address lines (16=two bytes). Using successive bytes as multipliers results in a number system with a radix (number base) of 256, which gets into big numbers fast.

A North Star BASIC program uses bytes and multipliers to calculate line numbers but for files it uses packed BCD with an extra byte to supply information about the

number. For eight-bit precision this takes five bytes. Fourteen-bit precision takes eight bytes. If the numbers used in a file are integers or positive amounts like dollars and cents, bytes can be used for a substantial saving in file space.

A typical personal finance program might require the following information in a record:

Record Number
Category Code
Amount
Date
Check Number
Description

If a character string of 40 bytes is used for the description and numbers are used for the other inputs, between 67 and 82 bytes are needed for each record. The string needs two overhead bytes and the numbers need from five to eight bytes each. If bytes are used instead of

Fig. 1

```
REM CONVERT NUMBER TO FOUR BYTES
DEFFNA(A)
A1=(A-INT(A)*100)\A2=A/65536\A2=INT(A2)
A5=INT(A)-A2*65536\A3=A5/256\A3=INT(A3)\A4=A5-A3*256
RETURN A1 AND A2 AND A3 AND A4
FNEND

REM CONVERT FOUR BYTES TO NUMBER
DEFFNB(A)
A1=A1/100\A2=A2*65536\A3=A3*256\A=A1+A2+A3+A4\RETURN A
FNEND

REM CONVERT NUMBER TO TWO BYTES
DEFFNC(C)
C1=C/256\C1=INT(C1)\C2=C-C1*256\RETURN C1 AND C2
FNEND

REM CONVERT TWO BYTES TO NUMBER
DEFFND=C1*256+C2
REM INPUTS
INPUT"CATEGORY CODE ",X
INPUT"AMOUNT ",A
INPUT"DATE ",B
INPUT"CHECK NUMBER ",C
INPUT"DESCRIPTION ",D$
REM CONVERT TO BYTES AND WRITE TO FILE
A=FNA(A)\C=FNC(C)
WRITE#0,&X,&A1,&A2,&A3,&A4,&B,&C1,&C2,D$,NOENDMARK
REM READ BACK FROM FILE
READ#0,&X,&A1,&A2,&A3,&A4,&B,&C1,&C2,D$
A=FNB(A)\C=FND(C)
```

numbers, the same information can be put in much less space. The category code can be a single byte. If more than 256 codes are needed it is probably not a "personal" finance program. The date can be a single byte if the records are kept by the month, or coded into two bytes if longer periods are used. Check numbers can be stored in two bytes if they don't exceed 65535. The record number can probably be calculated on the fly and left out of the record, but at most it should not take more than two bytes. The amount would probably need four bytes.

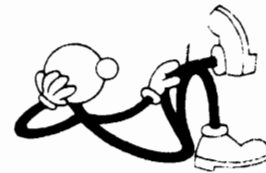
The following table shows the savings:

	8 dig.	14 dig.	bytes
Record no.	5	8	-
Code	5	8	1
Amount	5	8	4
Date	5	8	1
Check. no.	5	8	2
Descript.	42	42	42
Total	67	82	50

An easy way to program bytes into a file is to use user functions for multiple bytes. The program segment in Fig. 1 shows how this can be done.

FNA(A) and FNB(A) are really three-byte functions with the "cents" tacked on the end. It will handle numbers up to 16+ million but with 8 digit precision it truncates the cents for numbers over a million and causes an "OUT OF BOUNDS" error for numbers over 16+ million. Not too restrictive for personal finance.

The byte handling ability of North Star BASIC can be used to keep files to a minimum size. It is also useful if machine-language subroutines are "called" during a BASIC program since the bytes can be sorted and added, which doesn't work too well with packed BCD. It is a feature that should not be overlooked.



THE TROUBLES I'VE SEEN

By James A. Oliver
913 Palmetto
Pasadena, TX 77506
(713) 477-8458

I am a new member of INSUA. In reading Vol. III, no. 1 of Compass, I noticed that some the readers are experiencing what I have already gone through. I will try to cover some problems that had me going for a time.

1. The PHANTOM signal generated by the disk controller board.

On my own North Star plus three other boards that I have examined, the disk controller board is not wired up as the manuals show. This can be easily detected by buying an ohm meter to check for continuity between pin 67 and one of the PH holes. If there is no continuity between either hole and pin 67, then the board has to be modified. This is a simple process and I will modify anyone's board for the price of return postage and

insurance. If a person is good with a meter and a soldering iron, I can tell him on the phone (between 11:00 a.m. and 2:00 p.m. week days or anytime weekends) how to make these modifications.

2. The 2065 Cal Comp. 64K RAM board.

This one threw me for a while but the best that I can figure is that the board is very sensitive to noise on the S-100 bus. The way to get it working on the North Star is careful placement on the bus and a terminator. The terminator that I use is an active terminator made by CompuPro and costs about \$60.00. The placement of the boards is as follows: put the CPU board in slot 1, the 2065 board in slot 3, the disk controller board in slot 5, and the terminator in slot 7. I have used this set-up for a year with no problems. Calling Cal. Comp. is a waste of money for long distance phone calls as they were of no help and the persons that I talked to never heard of a North Star computer.

3. The H-25 or Z-25 printer made by Heath.

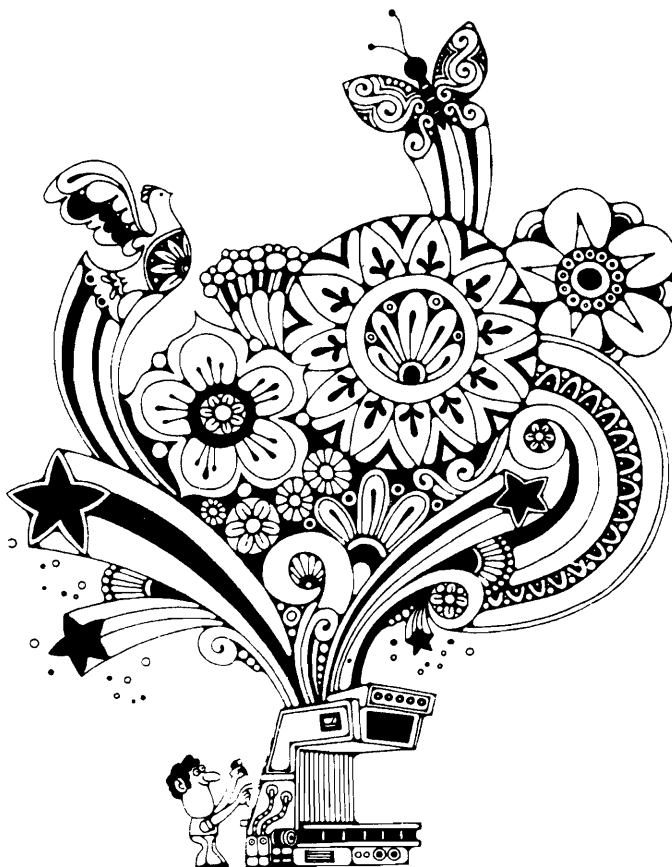
The hardwire handshaking on this printer comes out of pin 11 on their RS-232 connector. Inside the Horizon remove the jumper that goes from pin 1 to pin 12 at location 4D on the motherboard. Install a jumper from pin 11 of the serial connector to pin 1 of the header at location 4D. This will enable the printer to run at 9600 baud.

4. The Compass newsletter.

I have found many articles of interest but the replies to members' questions and explanations sometimes refer back to previous issues of the newsletter. Can anything be done to lessen the use of these back references for new members who do not have these back issues?

* * *

(In reply to comment 4, there is no practical way to avoid reference to prior issues; the alternative would be to repeat explanations anew in each issue. The best solution is to order a complete set of back issues!! --Ed.)



ALIEN EQUIPMENT

by Warren Lambert

I.E.E.E. 696 STATIC MEMORY

PRODUCT: RAM 17, 64K byte CMOS static memory board

VENDOR: CompuPro Division of Godbout Electronics, Oakland Airport, CA 94614

REVIEWER: W. Lambert

Philosophical raving

{begin raving} To me the most important virtue of the N* Horizon is the S-100 bus, which offers a computer that can grow with the most advanced products.

I want a generic computer in a white box labeled, "COMPUTER, 8/16/32-BIT, IEEE696, CP/M, RS232C," with interchangeable parts&programs. {end raving}

Godbout-CompuPro has been a leader in microcomputer design, and their products enjoy an outstanding reputation for advanced engineering. They offer S-100 CPU boards using the following chips: 8085 (6 MegaHertz), Z80B (6 MHz), 8088 (8 MHz), 8086 (10 MHz), iAPX286 (10MHz), M68000 (10MHz), and NS16032 (10MHz).

RAM17 is a 64K byte 8-bit RAM using the new "byte wide" 16K (8 bitX2K) CMOS static RAMS. "CMOS," or complementary metal-oxide semiconductors, offer low

power consumption and good noise tolerance. If you plan any computing outside of the Earth's nice protective atmosphere, you'll definitely need a static RAM board, as a single alpha particle can make your dear dynamic RAM lie to you. A 64K byte RAM17 uses 200 milliamps, less power than many dynamic RAMS. A static RAM, as opposed to a dynamic RAM, isn't fussy about timing in advanced applications, such as DMA (direct memory access).

RAM17 (list \$599 or certified \$699) is sold by cutthroat discounters for \$300-400 with 10 megahertz chips, nearly \$500 for the 12 megahertz version. These prices are slightly lower than a North Star 4 megahertz dynamic RAM board.

The RAM17 has 20 configuration switches able to adapt it to most IEEE696 computers, including those using either 24-bit addressing or the MPM-II and CP/M3 "bank switching with global memory," in which global memory contains basic operating system routines needed by programs in any 64K page of memory.

Godbout also offers two Product User Manuals for \$20-25 which contain well written manuals for all their products; while other companies now offer RAMS similar in design to RAM17, having well written manuals in advance was a strong selling point for me. The manuals are clear, even humorous, as in this passage from the system support manual: "How to configure your System Support 1 in under 5 minutes, without reading the manual ...for those of you who can't wait long enough to read the manual **...WE STRONGLY SUGGEST THAT YOU RELAX, AND READ THE MANUAL!!!**"[emphasis in original] Reading the RAM17 manual enabled me to install the board in about ten minutes with only one false start: A nonIEEE696 feature of

the Horizon is in its use of bus#61 as an "extra ground." The designer should not be criticized for this feature, as he did it in ancient times before even a wild eyed maniac dared to dream of personal computers with a 24 bit address. Nowadays in IEEE696, line #61 is defined as A20, or bit 20 of a optional 24 bit extended address. The morally upright solution that any decent person would use to fix #61 is given on p28 of N*'s HRZ-D-DOC (Revision 1): "Bus lines 61 and 70 are connected to the ground plane on the motherboard to reduce bus noise. ...[take your computer completely apart and] ...each line must be cut from the ground plane on the component side at each of the 12 connector pin locations...." For my little one-page 0-FFFF_H Horizon, I got by with the following dodge:

RAM17 has 8 switches [A16..A23] letting you place the board in any of 256 64K pages of extended memory. Bus#61 is A20, so I set the extended address for [A23..A16=00010000] rather than [A23..A16=00000000], and the board works fine. If I need a true 20 bit address, then I'll have to cut the traces, but if I can live with less than 1024K, the present solution should work.

The switch settings below work on my Horizon (DQ controller, Z80A "ZPB" CPU board by N*).

		1	2	3	4	5	6	7	8	9	10
S1	ON	<	*	*	<	*	*	*	*	*	*
	OFF	*	<	<	*	<	<	<	<	<	<
S2	ON	<	<	<	<	*	<	<	<	<	*
	OFF	*	*	*	*	<	*	*	*	*	<
		1	2	3	4	5	6	7	8	9	10

Fig.1. RAM17 settings for extended addressing mode in unmodified N* Horizon.

For general reviews of RAM17 see Machrone, B. "State-of-the-art S-100 Memory Boards," Micro-systems, January 1983, p70-73, or Anderton, C. "CMOS: Memory with a future," Byte, Jan. 1982, p.416-419.

POSSIBLE SETCPM BUG

There may be a bug in Lifeboat CP/M 2.23a SETCPM utility or the "configuration byte" for certain Horizon 3's. SETCPM, for those having older versions of CP/M, is a menu-oriented utility that patches CP/M options, e.g. single or double sided disks, printer port, etc. It's much easier to SETCPM than to figure out port addresses or configuration bytes.

On my Horizon 3, I have a single sided disk (A: 165K bytes), and two double sided minifloppies (B: and C: 338K bytes). I had trouble getting CP/M to treat C: as double sided.

After some several sympathy-seeking suicide gestures, I found that, even though I told SETCPM I had only 3 drives, that setting the nonexistent drive D: to double sided solved my problem. I sent the bug to Lifeboat around Christmas, but so far, no response.

JRT PASCAL

There are three minor bugs in the excellent bargain Pascal by JRT, a package priced so low that I feel grateful to JRT and inclined to overlook faults.

CUSTOMIZ. One detail concerns the terse explanation of the program CUSTOMIZ, on page 82 of the JRT2.1 manual. This program enables the user to tell the Pascal EXECutive what disk units to search for external procedures and functions. I have a Horizon 3 and I like to use C: as a "LOG" [Library of Garbage].

The first time I ran CUSTOMIZ on A: using the search list **CBA**; later compiling sometimes produced "BDOS ERROR ON J:" [J:! I went broke on C: -- who does it think I am, J:Paul Getty?]. Pascal couldn't find external procedures unless they were on the same drive as the Pascal program being compiled. At first I thought "Waddya want for \$30," but I was wrong! JRT is meticulous: I re-ran CUSTOMIZ with the search list '**CBA**' (i.e. with string quotes), everything ran fine: Pascal no longer searched in vain for disk J:, and external procedures could be found. I could even stay in WordStar, and compile and EXECute files using WordStar's R "run mode."

JRTASM. The assembler in version 2.1 won't work in a 58K computer, a problem confirmed by JRT. They said this bug is fixed in version 2.2 and 3.0 I was very pleased with their candid, helpful information on the phone, and they put some update sheets in the mail the very day I called. I can't however, commend them for shipping orders quickly & reliably. Both times I ordered JRT Pascal (ver 2&3), they lost my order.

TEXT MODE I haven't succeeded reading CP/M files in "text mode," so I always use BINARY, which works fine. It treats files literally, character-by-character, which I prefer. JRT's explanation of text vs. binary mode is negatively helpful, but the little dump program on page 64 (JRT Pascal User's Guide 2.1) is explanation enough. Incidentally, Gilbreath & Gilbreath in "Eratosthenes Revisited" (Byte, 1/83, page 323) said "We could not get the text files in JRT Pascal to work and finally had to resort to binary files and checking for end of file and end of line directly."

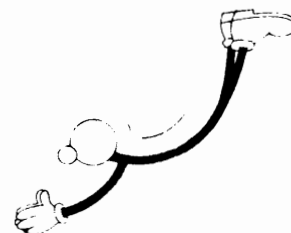
Despite problems in shipping, JRT achieved something very important and admirable in offering a decent version of a major language for only \$30.

P.S. It works fine to rename JRTPAS2.COM as COMPILE.COM and EXEC.COM as R.COM if you're in the habit of using other compile and run commands.

P.S.2. If anyone has successfully used JSTAT or JGRAF in a simple heuristic example, a grateful nation awaits!

HELP WANTED: Compass could use brief equipment reviews of ALIEN EQUIPMENT that worked or failed to work with your N* Horizon or Advantage. Send your notes to Warren Lambert, Lakeshore MHI, 5908 Lyons View Drive, Knoxville TN 37919. CP/M minifloppies for finished notes will be mailed back within a week. If you have something to share but don't want to write a review, call me [615 584-1561 x7724 weekdays, 938-3482 evenings&weekends] and I'll write a note quoting you.

%%



MAIL PROGRAM

By Leonard Morgenstern

Powerful mail list programs are fine for businesses and organizations that send thousands of letters, but are not needed for small lists. The attached program, MAIL, incorporates the names and addresses in DATA statements, and is quick and easy to set up or change.

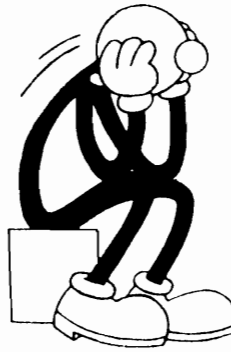


```
10 REM MAIL
20 REM SIMPLE MAIL LIST
30 DIM Q$(30)
40 !"THIS PROGRAM WILL PRINT MAIL LIST FOR ",
50 READ Q$\PRINT Q$!
60 !"TO ABORT, PRESS ^C"
70 INPUT "TO PROCEED, PRESS RETURN ",Q$
80 REM
90 P=2 \REM Set P to printer output device
100 REM
110 LINE#P,80
120 READ L
130 DIM A$(78*L)
140 REM
150 FOR I=1 TO LEN(A$)
160 A$(I,I)=" "
170 NEXT
180 REM
```

```

190 K=0
200 FOR I=0 TO L-1
210 J=I*78+K*26+1
220 READ A$(J)
230 IF A$(J,J)="*" THEN EXIT 270
240 NEXT
250 K=K+1
260 IF K<3 THEN 200
270 FOR I=1 TO LEN(A$) STEP 78
280 PRINT#P,A$(I,I+77)
290 NEXT
300 PRINT#P
310 IF A$(J,J)<>"*" THEN 150
320 !\!"END OF JOB "
330 END
340 REM . . . . .
350 REM The first data item must be a string containing the
360 REM the title of the mail list. The second item must be
370 REM the number of lines in each label. Then follow
380 REM strings containing the data. Blank lines are repre-
390 REM sented by null strings. They must not be omitted.
400 REM The list is terminated by the string "*" .
410 REM . . . . .
420 REM
430 DATA "COMPUTER CLUB",3
440 DATA "Dr. Norbert Wiener","MIT","Cambridge, Mass."
450 DATA "Dr. Donald Knuth","Stanford U.,""Stanford,CA"
460 DATA "*"

```



LEVY ON FORTH

By Saul Levy

Marc Perkel's Forth article (not FORTH since it is not a mnemonic like BASIC) contains the usual chest beating about how great his favorite language is (Pascal programmers sound exactly the same). I am familiar with Forth as I was employed by Kitt Peak National Observatory during the time Forth was being hailed as the answer to all programming needs on Kitt Peak Mountain by the National Radio Astronomy Observatory, where Chuck Moore did the major development of Forth, and later by Kitt Peak's programmers who wholeheartedly adopted Forth for the control of telescopes and instrumentation (a deluge of very complicated programs). A number of the staff became thoroughly addicted to Forth and are still proselytizing the glories of this language.

I was supposed to make myself familiar with Forth as part of my duties, but

couldn't see any reason to spend the large amounts of time needed to really learn this language. It was also hard to keep up with the (partly) Ph.D.-level programmers who were writing code that I was supposed to fix on the spot whenever hardware problems occurred (the hardware was very unreliable for quite some time). This last demand from one of my supervisors was totally impossible as I couldn't cope with doing my own job and also that of a (usually) excellent programmer at the same time.

I was shocked at how one-sided the viewpoints were when people came to Kitt Peak to learn about Forth and whether it was the language to use at their own institutions. Forth, like any other language, is not all flowers and spice (even roses have thorns)! I am for the use of whatever language allows rapid and useful program development. If you have a choice, use whichever language you want

to. Often you get no choice and I'm not sure if this is a big problem, although some programmers complain bitterly about having to use BASIC all of the time!

Kitt Peak was talking about dumping Forth and using FORTRAN instead, allowing most scientists to create these programs instead of a small "elite." The problem of redeveloping all of the Forth programs was just too excessive and so this plan was dropped. Another limitation I recently heard about is that Forth, Inc. is no longer in business.

Since Forth is a stack-oriented language, all is not as easy as the examples indicate. The code I have seen has been full of DUPs, SWAPs, OVERs, ROTs, DROPs, 2DROPs, 2DUPs, etc. I am amazed that any programmer can make heads or tails out of anything written by someone else in this language!

"Floating point (arithmetic) is rarely needed." This may be true if you're playing games, but the real world says otherwise! Even handling dollars and cents is ridiculous with integer arithmetic. You should try to program with only integers so you can see just how limiting this statement really is. Most Forth implementations (that you can afford!) do not include floating point arithmetic, so beware!

Mr. Perkel's examples will be quite obtuse to beginning programmers (the "." word is not explained at all, for instance). Forth seems to have the peculiar problem that it is hard to explain (by any of the people that I have heard try). Forth is a language which really demands that you sit down at a terminal and keep at it until you begin to understand it. I haven't looked at any of the books available, so can't say if this problem occurs in book form also!

The author's BASIC example is a cheap shot that didn't go very far. The use of GOSUBs in BASIC causes a lot of confusion which can be eliminated by simply not using them! I have had to write a whole package using GOSUBs and hated every minute I wasted looking up where they were and what they do. If memory is limited or you can more easily keep track of what you are doing, then go ahead and use GOSUBs. Otherwise, it is much easier to just write "in-line" code which is much easier to debug and runs faster.

The rest of line 6730 doesn't prove the author's contention either because it is out of context and meaningless. Any good BASIC programmer would not have any problem understanding what this line means if it were embedded within a program (my apologies to beginners!) Also, the Forth Disk I/O section doesn't even come close to North Star's BASIC which allows eight files to be accessed at one time. Nothing I have ever written even comes close to needing that many open files. A larger disk buffer would be welcome, but even so, I have found that the new quad drives allow rapid file access. I have also found North Star's BASIC to run very rapidly and thus to defuse this point in all of the harangues I have read against BASIC. Speed for speed's sake is often not needed in many applications. I add assembly routines if speed is really important (like in sorting). It also helps to compare BASIC code from a knowledgeable programmer rather than the (mostly) beginner-written code you see so much of. Note that I do not consider it a sin to be a beginner!

Most Forth versions are not standardized as far as I can tell. The standards groups try, but most implementations can hardly be considered as following the standard. Another point to beware of!

Mr. Perkel's version of Forth does sound interesting (as Forths go). The price is right, but is a language which demands an expert to even get it going, a language for the masses? BASIC certainly demands no such thing. fig-Forth also demands a heck of a lot out of anyone trying to get it to run on a disk-based system. I have found Forth to be a language which appeals to a limited group of programmers. I hardly see this changing in the future. If you like it, then use it. Just keep the addicts from bending my ears so hard!



OVER THE HORIZON?

Greetings INSUA members,

Like you I bought my Northstar a number of years ago, when it was the very best machine on the market. It has only been during the last year that I have regretted the purchase. Why? Well, not because my Horizon is beginning to fail, for I have only had two failures in the last 5 years, both memory chips. The reason I regret the purchase is that there exist machines today that can access 1 megabyte of storage and have an entire world of software to chose from. The Horizon **will** become obsolete there is no way we can prevent this, but we can extend the life and fun of the machine for a number of years if we make a concentrated effort to co-operate.

This club numbers in the hundreds. If each of us contributed only 1 piece of software each year we would prolong the life of our machines. How many of us have written utilities, games, teaching aids, puzzles etc. for the Horizon and have then filed them away and forgotten about them? I have at least 10 such programmes, would you be interested in sharing your creations? There is no need to restrict yourself to North Star BASIC: the programmes could be in FORTH, FORTRAN, MBASIC, CBASIC.

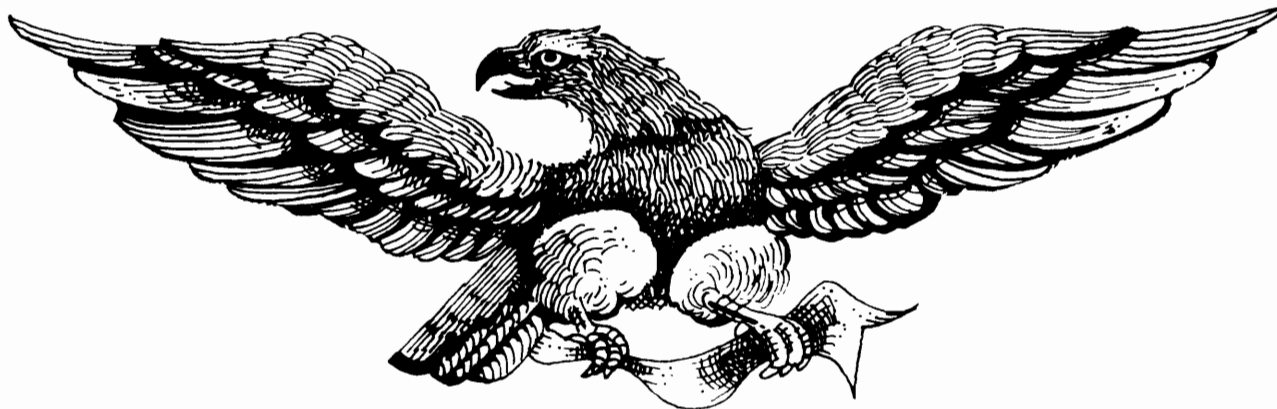
My machine, affectionately named THE BEAST, will last me for the next 10 years because I cannot afford to replace it each year with the latest and greatest. I think it a pity that the NorthStar Horizon, will most likely become obsolete years before its MTBF. As long as my machine works, I will make use of it.

Yours Truly,
Peter de Jager
Willowdale, Ont., Canada

P.S. Has anyone written a Basic text processor? or a full screen editor for a cursor addressable screen? If so, will you share it with your club?

* * *

(Horizons are still rolling off the assembly line at North Star; moreover, 16-bit boards will keep your Horizon up to date. The old workhorse has a lot of life in it yet! -- Ed.)



THE GAME OF KAYLES

By Peter de Jaeger

THE GAME OF KAYLES WAS FIRST INTRODUCED TO THE GAMING COMMUNITY BY DUDNEY. SINCE THEN IT HAS BEEN A SOURCE OF FRUSTRATION AND JOY, TO BOTH MATHEMATICIANS AND GAMERS.

THE MOST RECENT REFERENCE TO THE GAME THAT I AM AWARE OF IS IN A BOOK ENTITLED ON NUMBERS AND GAMES BY J. H. CONWAY (LONDON: ACADEMIC PRESS, 1976). THIS IS A DELIGHTFUL EXCURSION INTO MATHEMATICAL GAMES AND NUMBERS THE LIKE OF WHICH YOU HAVE NEVER SEEN. FOR THOSE OF YOU WHO ENJOY STRAINING YOUR BRAIN WRITING VERY UNUSUAL COMPUTER PROGRAMMES, THEN THEN THIS COULD BE CONSIDERED REQUIRED READING.

BEFORE WE WALK THROUGH THE PROGRAMME LET US DEFINE/DESCRIBE THE GAME.

KAYLES IS A GAME OF SKILL BETWEEN TWO PLAYERS. THE EQUIPMENT CONSISTS OF A NUMBER OF BOWLING (SKITTLES?) PINS AND A BALL. THE GAME PROCEEDS AS FOLLOWS.

1) THE PINS ARE SET UP IN A NUMBER OF ROWS, FACING THE TWO PLAYERS.

2) THE PLAYERS ALTERNATE KNOCKING DOWN A SINGLE PIN OR 2 ADJACENT PINS. (WE ASSUME THAT THE PLAYERS ARE ACCURATE AND MAKE NO ERRORS)

3) THE LAST PLAYER TO KNOCK DOWN A PIN (OR PINS) IS THE WINNER

AT FIRST GLANCE THIS WOULD APPEAR TO BE JUST ANOTHER VARIATION ON NIM. HOWEVER, AS I SOON FOUND OUT, IT IS A MITE MORE COMPLICATED THAN THE ORIGINAL NIM. THOSE OF YOU WHO BELIEVE THAT THIS IS NOT SO ARE INVITED TO STEP OUTSIDE THE CLASSROOM AND WRITE A COMPUTER PROGRAMME TO PLAY A PERFECT GAME OF KAYLES.

WELCOME BACK. LET US NOW PROCEED TO THE PROGRAMME LISTING, WHERE WE WILL EXPLAIN HOW THE PROGRAMME IS PUT TOGETHER. THE THEORY BEHIND THE GAME WILL REMAIN A MYSTERY TO THOSE WHO CHOOSE ONLY TO COPY THE PROGRAMME WITHOUT TRYING TO FOLLOW THE MANY COMMENTS (REM'S).

LINES 10-110 WE START WITH A PROGRAMME LABEL AS DOCUMENTATION

LINES 120-140 NEXT WE DEFINE SOME OF THE ARRAYS BEING USED

LINE 150 DEFINES THE MAXIMUM NUMBER OF PINS WE WILL ALLOW SET AT 19. THIS WILL REQUIRE 72 CHARS. ON YOUR SCREEN.

LINES 190-400 WE SET UP THE NIMSUM TABLE. THIS SHOULD BE FAMILAR TO WRITERS OF NIM GAMES.

LINES 450-550 WE SET UP THE TABLE OF GRUNDY NUMBERS

LINES 590-610 WE BEGIN THE HUMAN/COMPUTER INTERFACE WITH THE
 OPTION OF READING A PROGRAMME EXPLANATION. NOTICE THE '!'
 THIS IS A SHORT FORM FOR PRINT IN NORTH STAR BASIC.

LINE 650 SETUP THE PINS. P\$ IS A VECTOR OF CHARACTERS OR A
 CHARACTER STRING. THIS STRING CONTAINS THE STATUS OF EACH OF
 THE PINS. AT THE START ALL THE PINS ARE CONSIDERED TO BE DOWN.

LINE 660 WE START A NEW SERIES OF RANDOM NUMBERS. RND(-1)
 PERFORMS THE SAME FUNCTION AS RANDOMIZE DOES IN OTHER BASICS.

LINES 670-690 WE NOW SET UP A RANDOM NUMBER OF PINS. A MAXIMUM
 OF 'P'

LINES 730-750 WE GIVE THE POOR HUMAN A CHANCE TO GO FIRST; AFTER
 ALL, HE NEEDS ALL THE HELP HE CAN GET.

LINES 810-860 THIS IS THE CONTROL SEQUENCE

+START+ PRINT OUT STATUS OF PINS
 PLAYERS TURN
 END OF GAME? YES-GOTO END
 NO-CONTINUE
 COMPUTERS TURN
 END OF GAME? YES-GOTO END
 NO-GOTO START
 +END+ PRINT WINNER.

LINES 880-910 END GAME AND REQUEST IF HUMAN WANTS TO BE BEATEN
 AGAIN.

LINES 950-1110 RULES FOR KAYLES.

LINES 1130-1360 THIS IS THE PRINT ROUTINE. THIS COULD BE
 CHANGED TO SUIT ANY TERMINAL.

LINES 1400-1650 THIS SUBROUTINE TAKES IN THE HUMAN RESPONSE AND
 MAKES SURE THAT IT IS VALID.

LINES 1680-1760 CHECK THE GAME STATUS.

LINES 1820-2950 THIS IS THE 'BRAINS' OF THE PROGRAMME. IT IS
 HERE THAT THE COMPUTER'S TURN IS DECIDED.

LINES 1800-1950 RESET VARIABLES TO BE USED. WE ALSO EXPLAIN
 WHAT THE VARIABLES ARE USED FOR

LINES 1990-2190 IN THIS 'LOOP' WE 'READ' THE PIN CONFIGURATION
 TO SEE HOW WE ARE DOING.

LINES 2230-2250 HERE WE 'NIMSUM THE GRUNDY NUMBERS' TO FIND THE
 VALUE OF THE POSITION.

LINE 2300 IF THIS VALUE IS ZERO THEN THE HUMAN IS WINNING. THIS IS OF COURSE HIGHLY UNLIKELY.....

** NOTE ** IF YOU WISH YOU COULD HAVE THE COMPUTER MAKE MISTAKES. I.E. MAKE A RANDOM MOVE WHEN HE IS WINNING. THIS SUGGESTION IS FOR THOSE OF YOU WHO DO NOT LIKE BEING BEATEN ALL THE TIME.

LINES 2340-2830 THE COMPUTER SEARCHES HERE, TRYING GALLANTLY TO GET HIS NIMSUM TO ZERO. TO THOSE OF YOU INTERESTED IN THE THEORY BEHIND KAYLES, STUDY THIS SECTION CAREFULLY.

LINES 2870-2950 HERE THE COMPUTER RANDOMLY PICKS A PIN. THE ONLY HOPE THE COMPUTER HAS IS THAT THE HUMAN MAKES AN ERROR AND GIVES UP THE WINNING POSITION.

THIS THEN IS THE PROGRAMME AND AN EXPLANATION; IF ANY BUGS ARE FOUND OR IF YOU HAVE COMMENTS TO MAKE, MY ADDRESS IS 150 CACTYS AVE. #38, WILLOWDALE, ONTARIO, CANADA M2R 2V3

```
10 REM .....
20 REM ..
30 REM ..          KAYLES. AN EXTENDED NIM VARIATION          ..
40 REM ..
50 REM .. AUTHOR\ PETER DE JAGER                               ..
60 REM .. REFERENCE\ 'ON NUMBERS AND GAMES'                   ..
70 REM ..          J. H. CONWAY (ACADEMIC PRESS 1976)         ..
80 REM ..
90 REM .. COMPUTER\ NORTH STAR          LANGUAGE\ NS BASIC    ..
100 REM .. STORAGE SIZE = 39 BLOCKS    RUN SIZE = 21.3K      ..
110 REM .....
120 DIM P(10,2)      \REM THIS IS USED FOR 'READING' A CONFIGURATION OF PINS
130 DIM B(19)        \REM THIS VARIABLE IS USED WHEN WE MAKE A RANDOM CHOICE
140 DIM P$(19)       \REM THIS IS WHERE THE PIN POSITIONS ARE KEPT
150 LET P=19         \REM THIS IS NUMBER OF PINS POSSIBLE
160 REM
170 REM THIS IS THE NIM-SUM TABLE FOR ALL NUMBERS 0-15
180 REM
190 DIM N(15,15)
200 FOR I=0 TO 15
210   FOR J=0 TO 15
220     READ N(I,J)
230     NEXT J
240   NEXT I
250 DATA 0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15
260 DATA 1,0,3,2,5,4,7,6,9,8,11,10,13,12,15,14
270 DATA 2,3,0,1,6,7,4,5,10,11,8,9,14,15,12,13
280 DATA 3,2,1,0,7,6,5,4,11,10,9,8,15,14,13,12
290 DATA 4,5,6,7,0,1,2,3,12,13,14,15,8,9,10,11
300 DATA 5,4,7,6,1,0,3,2,13,12,15,14,9,8,11,10
310 DATA 6,7,4,5,2,3,0,1,14,15,12,13,10,11,8,9
320 DATA 7,6,5,4,3,2,1,0,15,14,13,12,11,10,9,8
330 DATA 8,9,10,11,12,13,14,15,0,1,2,3,4,5,6,7
```

```

340 DATA 9,8,11,10,13,12,15,14,1,0,3,2,5,4,7,6
350 DATA 10,11,8,9,14,15,12,13,2,3,0,1,6,7,4,5
360 DATA 11,10,9,8,15,14,13,12,3,2,1,0,7,6,5,4
370 DATA 12,13,14,15,8,9,10,11,4,5,6,7,0,1,2,3
380 DATA 13,12,15,14,9,8,11,10,5,4,7,6,1,0,3,2
390 DATA 14,15,12,13,10,11,8,9,6,7,4,5,2,3,0,1
400 DATA 15,14,13,12,11,10,9,8,7,6,5,4,3,2,1,0
410 REM
420 REM AND THE FOLLOWING ARE THE GRUNDY NUMBERS FOR KAYLES
430 REM FROM N=0 TO 83
440 REM
450 DIM G(84)
460 FOR I=0 TO 83
470     READ G(I)
480     NEXT I
490 DATA 0,1,2,3,1,4,3,2,1,4,2,6
500 DATA 4,1,2,7,1,4,3,2,1,4,6,7
510 DATA 4,1,2,8,5,4,7,2,1,8,6,7
520 DATA 4,1,2,3,1,4,7,2,1,8,2,7
530 DATA 4,1,2,8,1,4,7,2,1,4,2,7
540 DATA 4,1,2,8,1,4,7,2,1,8,6,7
550 DATA 4,1,2,8,1,4,7,2,1,8,2,7
560 REM
570 REM
580 REM
590 !"DO YOU KNOW OF THE GAME OF KAYLES ", \ INPUT A$
600 IF A$="" THEN 590
610 IF A$(1,1)="N" THEN GOSUB 920
620 REM
630 REM HERE WE SET UP A RANDOM COLLECTION OF PINS
640 REM
650     P$="00000000000000000000" \REM SET ALL PINS DOWN
660     B(0)=RND(-1) \REM THIS IS THE SAME AS RANDOMIZE IN OTHER BASICS'
670     FOR I=1 TO P \REM HERE WE ALLOW FOR A MAX OF P PINS
680         IF RND(0)<.6 THEN P$(I,I)="1" \REM THE HIGHER THE .6 THE MORE PINS
690     NEXT I
700 REM
710 REM CHOOSE TO GO FIRST OR SECOND
720 REM
730 !"DO YOU WISH TO PLAY FIRST ", \INPUT A$
740 IF A$="" THEN 730
750 IF A$(1,1)="Y" THEN 810
760 GOSUB 1120
770 GOTO 840
780 REM
790 REM THE FOLLOWING IS THE BASIC PROGRAMME FLOW
800 REM
810 GOSUB 1120 \REM PRINT OUT PINS
820 GOSUB 1370 \REM ASK FOR AND CHECK PLAYERS INPUT
830 GOSUB 1670 \ IF F=1 THEN 880\REM CHECK FOR END OF GAME
840 GOSUB 1770 \REM COMPUTER MAKES HIS MOVE
850 GOSUB 1670 \ IF F=1 THEN 890\REM CHECK FOR END OF GAME
860     GOTO 810 \REM START OVER AGAIN.....
870 END
880 !" YOU HAVE WON THE GAME.....LUCKY..." \ GOTO 900
890 !" I HAVE BEATEN YOU ....ONCE MORE ....." \ GOTO 900
900 !"DO YOU WISH TO PLAY AGAIN ", \ INPUT A$
910 IF A$(1,1)="Y" THEN 650 \ END
920 REM

```

```

930 REM RULES FOR KAYLES
940 REM
950 !" THE GAME OF KAYLES WAS INTRODUCED BY DUDENEY. IT IS PLAYED BY SKILFUL"
960 !" PLAYERS WITH A NUMBER OF ROWS OF PINS. SEE BELOW."
970 !
980 !"  [] [] [] []      [] [] []      [] []      [] []"
990 !"  [ ][ ][ ][ ]  [ ][ ][ ]  [ ][ ]  [ ][ ]"
1000 !"  [01][02][03][04]  [06][07][08]  [10][11]  [13][14]"
1010 !"  [ ][ ][ ][ ]  [ ][ ][ ]  [ ][ ]  [ ][ ]"
1020 !
1030 !" FOR OUR PURPOSES WE SHALL ASSUME THAT THE PLAYERS ARE SO SKILFUL"
1040 !" THAT THEY NEVER MAKE A MISTAKE. IE THAT THEY ALWAYS DO WHAT THEY"
1050 !" INTENDED TO. IT IS POSSIBLE TO KNOCK DOWN A SINGLE PIN OR TWO PINS"
1060 !" THAT ARE ADJACENT TO EACH OTHER. IT IS IMPOSSIBLE TO KNOCK DOWN"
1070 !" MORE THAN TWO PINS WITH A SINGLE SHOT."
1080 !" THE OBJECT OF THE GAME IS TO KNOCK DOWN ALL THE PINS , THE LAST"
1090 !" PERSON ABLE TO KNOCK DOWN A PIN WINS."
1100 !" THE PLAYERS ALTERNATE KNOCKING DOWN PINS."
1110 RETURN
1120 !
1130 REM THIS IS THE PRINT ROUTINE
1140 REM
1150 FOR I=1 TO P
1160 IF P$(I,I)="1" THEN !" [ ] ", \ IF P$(I,I)="0" THEN !"   ",
1170 NEXT
1180 !
1190 FOR I=1 TO P
1200 IF P$(I,I)="1" THEN !" [ ]", \ IF P$(I,I)="0" THEN !"   ",
1210 NEXT
1220 !
1230 FOR I=1 TO P
1240 IF P$(I,I)="0" THEN 1290
1250 A$=STR$(I) \REM NS BASIC NUMBER TO A BLANK + STRING
1260 IF I<10 THEN A$(1,1)="0" ELSE A$=A$(2)
1270 !"[" ,A$,"]",
1280 GOTO 1300
1290 !"   ",
1300 NEXT
1310 !
1320 FOR I=1 TO P
1330 IF P$(I,I)="1" THEN !" [ ]", \ IF P$(I,I)="0" THEN !"   ",
1340 NEXT
1350 ! \ !
1360 RETURN
1370 REM
1380 REM PLAYER INPUT AND CHECKING ROUTINE
1390 REM
1400 !" WILL YOU KNOCK DOWN ONE OR TWO PINS ", \ INPUT A$
1410 IF A$="ONE" THEN 1560 \ IF A$="1" THEN 1560
1420 REM
1430 REM CHOSEN TO KNOCK DOWN 2 PINS. INPUT 0,0 TO CHANGE YOUR MIND
1440 REM
1450 !" WHICH PINS WILL YOU KNOCK DOWN ", \INPUT P1,P2 \IF P1*P2=0 THEN 1400
1460 P1=INT(P1) \ IF P1<1 THEN 1400 \ IF P1>P THEN 1400
1470 P2=INT(P2) \ IF P2<1 THEN 1400 \ IF P2>P THEN 1400
1480 IF ABS(P1-P2)>1 THEN 1400
1490 IF VAL( P$(P1,P1) )<>0 AND VAL( P$(P2,P2) )<>0 THEN 1520
1500 !" TRY AND CHEAT A POOR COMPUTER .... YOU'RE A BAAAAAD .....BODY..."

```

```

1510 GOTO 1400
1520 P$(P1,P1)="0"
1530 P$(P2,P2)="0"
1540 GOSUB 1120
1550 RETURN
1560 REM
1570 REM PLAYER HAS CHOSEN TO KNOCK DOWN ONE PIN
1580 REM
1590 !" WHICH PIN WILL YOU KNOCK DOWN ", \ INPUT P1
1600 P1=INT(P1) \ IF P1<1 THEN 1400 \ IF P1>P THEN 1400
1610 IF P$(P1,P1)<>"0" THEN 1640
1620 !" TRY AND CHEAT A POOR COMPUTER.....TSK...TSK."
1630 GOTO 1400
1640 P$(P1,P1)="0"
1650 GOSUB 1120
1660 RETURN
1670 REM
1680 REM END OF GAME CHECK . IF GAME IS NOT FINISHED SET F=0
1690 REM
1700 F=1
1710 FOR I=1 TO P
1720 IF P$(I,I)="1" THEN EXIT 1750
1730 NEXT
1740 RETURN
1750 F=0
1760 RETURN
1770 REM
1780 REM THE COMPUTER MAKES HIS MOVE
1790 REM
1800 REM .....
1810 REM
1820 REM WE WILL FIRST ZERO OUT ALL VARIABLES THAT WE USE HERE
1830 REM
1840 FOR I=1 TO INT(P/2)+1 \REM MAX NUMBER OF PIN GROUPS
1850 P(I,0)=0 \REM GRUNDY NUMBER FOR THIS GROUP
1860 P(I,1)=0 \REM START LOCATION OF THIS GROUP
1870 P(I,2)=0 \REM LENGHT OF THIS GROUP
1880 NEXT
1890 T=0 \REM TARGET FOR GRUNDY NIM-SUM
1900 Z1=0 \REM ARE WE READING A GROUP ? 0=NO !
1910 Z2=0 \REM HOW MANY GROUPS HAVE WE GOT
1920 Z3=0 \REM HOW MANY PINS DOES COMPUTER HIT
1930 Z4=0 \REM TOTAL NUMBER OF PINS LEFT
1940 Z5=0 \REM THE RANDOM PIN THAT COMPUTER CHOOSES
1950 REM .....
1960 REM
1970 REM WE WILL NOW 'READ' THE PIN CONFIGURATION
1980 REM
1990 FOR I=1 TO P
2000 IF P$(I,I)="1" THEN 2080
2010 IF Z1=0 THEN 2130 \REM NOT WORKING ON A GROUP WE CONTINUE
2020 REM
2030 REM WE HAVE FINISHED A GROUP. WE FINISH GETTING INFORMATION
2040 REM
2050 P(Z2,0)=G(P(Z2,2)) \REM GRUNDY NUMBER FOR THAT GROUP
2060 Z1=0 \REM WE ARE FINISHED WITH A GROUP ,SET SWITCH=0
2070 GOTO 2130
2080 IF Z1=1 THEN 2120 \REM WE ARE ALREADY WORKING ON A GROUP

```

```

2090     Z1=1                \REM HERE WE HAVE STARTED A GROUP
2100     Z2=Z2+1            \REM INCREASE THE NUMBER OF GROUPS WE HAVE
2110     P(Z2,1)=I         \REM START LOCATION OF THIS GROUP
2120     P(Z2,2)=P(Z2,2)+1 \REM INCREASE LENGTH OF GROUP BY ONE
2130     NEXT I
2140     REM
2150     REM MAKE SURE WE HAVE TIED UP LOOSE ENDS
2160     REM
2170     IF Z1=0 THEN 2200   \REM NOT WORKING ON GROUP. WE CONTINUE
2180     P(Z2,0)=G( P(Z2,2) ) \REM GRUNDY NUMBER FOR THAT GROUP
2190     Z1=0
2200     REM
2210     REM WHAT IS THE GRUNDY NUMBER FOR THIS SET OF PINS
2220     REM
2230     FOR I=1 TO Z2
2240     T=N(T,P(I,0))      \REM NIMSUM ALL THE GRUNDY NUMBERS FOR CONFIGURATION
2250     NEXT I
2260     REM
2270     REM IF T=0 THEN THE OTHER PLAYER IS WINNING, CHOSE A RANDOM PIN, FROM
2280     REM THE PINS AVAILABLE.
2290     REM
2300     IF T=0 THEN 2840
2310     REM
2320     REM WE MUST NOW DECIDE WHICH PIN OR PINS TO HIT
2330     REM
2340     FOR I=1 TO Z2      \REM WE HAVE Z2 GROUPS . WE CHOOSE ONE AS A
2350                         REM POSSIBLE GROUP TO HIT.
2360     T=0                \REM SET TARGET GRUNDY NIM SUM TO ZERO
2370     FOR J=1 TO Z2      \REM WHAT DO OTHER GROUPS NIM SUM TO ??
2380     IF I=J THEN 2400   \REM BYPASS 'POSSIBLE GROUP'
2390     T=N(T,P(J,0))      \REM NIM SUM
2400     NEXT J
2410     REM
2420     REM T IS NOW OUR TARGET GRUNDY NIM-SUM , WE PROCEED
2430     REM TO LOOK FOR A 'STRIKE' OR HIT THAT WILL FORM A NEW CONFIGURATION
2440     REM THAT WILL NIM SUM TO ZERO.
2450     REM FIRST WE SHALL LOOK A SINGLE HITS.
2460     REM
2470     FOR J=1 TO INT( ( P(I,2)+1)/2 ) \REM WE NEED ONLY CHECK HALF-WAY
2480     IF T=N(G(J-1),G(P(I,2)-J)) THEN EXIT 2650
2490     REM IF THE ABOVE IS TRUE THEN WE HAVE A SOLUTION
2500     REM THE J'TH PIN IN THE I'TH GROUP
2510     NEXT J
2520     REM
2530     REM NO LUCK WITH THE SINGLE PINS , CHECK DOUBLES
2540     REM
2550     IF P(I,2)=1 THEN 2690
2560     FOR J=1 TO INT( ( P(I,2)+1)/2 )
2570     IF T=N(G(J-1),G(P(I,2)-J-1)) THEN EXIT 2670
2580     REM IF THE ABOVE IS TRUE THEN WE HAVE A SOLUTION
2590     REM THE J'TH & J+1'TH PINS IN THE I'TH GROUP
2600     NEXT J
2610     REM
2620     REM NOT FOUND AS YET , SO WE CONTINUE TO SEARCH
2630     REM
2640     GOTO 2690
2650     Z3=1
2660     EXIT 2700

```

```

2670 Z3=2
2680 EXIT 2700
2690 NEXT I
2700 REM
2710 REM WE ARE OUT THE LOOP IF Z3=0 THEN WE HAVE NOT FOUND A SOLUTION.
2720 REM Z3=1 A SINGLE SOLUTION
2730 REM Z3=2 A DOUBLE SOLUTION
2740 REM
2750 IF Z3=0 THEN 2840
2760 IF Z3=2 THEN 2800
2770 !" I WILL HIT PIN ",P(I,1)+J-1
2780 P$(P(I,1)+J-1,P(I,1)+J-1)="0"
2790 RETURN
2800 !" I WILL HIT PIN ",P(I,1)+J, " AND PIN ",P(I,1)+J-1
2810 P$(P(I,1)+J,P(I,1)+J)="0"
2820 P$(P(I,1)+J-1,P(I,1)+J-1)="0"
2830 RETURN
2840 REM
2850 REM WE CHOSE A RANDOM PIN
2860 REM
2870 FOR I=1 TO P
2880 IF P$(I,I)="0" THEN 2910
2890 Z4=Z4+1
2900 B(Z4)=I
2910 NEXT I
2920 Z5=INT( RND(0)*Z4+1 )
2930 !" I WILL HIT PIN ",B(Z5)
2940 P$(B(Z5),B(Z5))="0"
2950 RETURN

```

LOST WORDSTAR TEXT?

HERE'S AN ANSWER!

Henry M. Spelman III

Have you ever had a system crash while working on a WordStar file and had to reboot? Or done something silly like resetting the computer before saving your file? Then you probably cursed, muttered or cried as you tried to reconstruct the missing text, or the editing that you had done since your last save. Be of good cheer, there may be a way out.

If you have not had a power failure, nor turned the computer off, nor started a new edit with WordStar, nor done anything else affecting memory, then the "missing" file is probably still in memory and/or in temporary disk files. A reset to get back to CP/M does not affect anything in the transient program area (TPA) where the

file you were working on resides. Your task is to locate the file in memory, save it to disk, and strip off any garbage that you may have picked up along with the file. Then you may have to combine this file with other files already on the disk.

FIRST use the REN command to rename the backup file so that it won't accidentally be erased. If the file you were working on was TEST.TEX, then TEST.BAK is the latest complete version that you saved on the disk. Change it to TEX.SAV so that you won't lose it. This is a base from which to proceed if all else fails. Protect it as though it were pure gold.

I use two tools for recovering from such crashes. First, that wonderful set of utilities for CP/M called "POWER!" (Version 3.03) (from Computing, 2519 Greenwich St., San Francisco, CA 94123) and second, a Texas Instruments "LCD

Programmer" calculator (which converts numbers from hex to decimal at the push of a key). If you don't have the calculator, you can use POWER! to convert numbers from one base to the other. You could accomplish the same thing with DDT and pencil and paper, but it isn't as easy. For one thing, you can't save a file with DDT unless you first move the file to 0100H.

Now for the recovery. I will assume that you have POWER! up and running, and that you have some familiarity with it. You need to remember a key word that you used in the file in question, preferably near the start. In the following instructions, I will use the word "KEY" where you should use your key word (or, if your organic memory is a blank, any word that you are quite sure is in the file -- even "the"). Use upper and lower case just as you did in the file. Do not include the last letter of the key word in your key. WordStar may have set a bit on that letter in the file so that the search will not recognize it. In the following dialog, the underlined> items are what you type in, the rest are the computer's replies.

Load POWER!, turn on your printer, hit control-P (to send all output to the printer, so that you'll have a record of the hex numbers you'll need later) and proceed.

A=LOG

You will get information about defaults and, at the bottom of the screen, a line that gives the lower and upper limits of the TPA (the constants are for my 58K system -- yours may be different):

TPA 3D00H - BDFFH 258 sectors

This gives you the area to search for the missing file. Go on with:

A=SEARCH 3D00 BDFH "KEY"

(The quotation marks are necessary to indicate that the search key consists of ASCII characters. No punctuation is allowed between the lower and upper search limits.)

The computer will give a list of the memory locations of each occurrence of the key word in the TPA area of RAM. It will look like this:

```
7BD7 "KEY"  
7CBA "KEY"  
7E9E "KEY"  
B80E "KEY"  
BB12 "KEY"  
BB59 "KEY"
```

Some of these will probably be false alarms, but if your key word is actually in your missing file, one or more of the addresses returned by the computer will be in the file. If you get no output, try another key word.

Now type control-P to turn off the printer and scan memory to see where your file begins and ends. Start a little before the first location of KEY. In this case, type:

A=DUMPA 7800 BDFH

The computer will reply with an ASCII dump of the contents of RAM starting at 7800. This may show you the start of the desired file, put you right in the middle of it, or give you a screenful of irrelevant matter. In the latter case, hit RETURN until you reach something familiar. Don't be dismayed if the last letter of each word seems to be missing. They are really there, but WordStar has flagged them, and in DUMPA they will print as spaces. In this particular case, I found that the text I was looking for started at 7850H, which showed up on the first screenful of text displayed. Then I hit RETURN repeatedly to display more memory, and found that the text ended at 821FH. The later occurrences of KEY were leftovers from other phases of the edit, or artifacts from other programs. If 7800H is in the middle of the file, try again with a lower starting address until you find the beginning of the file.

Now for some hex math. (I use the TI calculator to switch bases; the DS command of POWER! will also do so for you.) 821FH - 784FH = 09D0H = 2512D. So the file starts at 7850H and is 2512

bytes long. Divide 2512 by 128 to get 19.625 sectors. Decide on a name for the recovered file, say "FOUND.TXT" and type:

A=SAVE FOUND.TXT 7850 20

(Save 20 sectors starting at 7850H to FOUND.TXT.)

If you were editing a small file, that's all there is to it. You have recovered the file. All you have to do is call it up for editing under WordStar, examine the beginning and end to see if there is any extra material, or anything missing. Extra characters should be deleted and the file saved. If there are missing characters, double check the addresses you used for the SAVE command to be sure that you included everything. Hex math can be confusing! If the rest of the file is not in RAM, it may be in one or more of several disk files. A long file is only partly in RAM at any time. The file that you have saved was that part of the file that was in RAM when the system crashed. The rest of the file is probably already on the disk. There may be duplication of some parts of the file, one version unedited, the other the latest version.

If you were editing TEST.TXT, then you could have any or all of the following files on the disk:

TEST.TXT (the partly edited version of the file) TEST.SAV (the previous version of the file, as it was the last time you saved it. You changed its name from TEST.BAK as the first step of the recovery effort)

TEST.*** (a temporary file created by WordStar to hold part of the edited file)

EDBACKUP.*** (another temporary file)

FOUND.TXT (the file you have just recovered)

Use the TYPE command to look at the .*** files. If either one is empty, it may be erased. If either has data, rename the file TEST.TMP or EDBACKUP.TMP. Then WordStar can't automatically erase them on you.

Use WordStar to examine TEST.TEX, TEST.TMP, EDBACKUP.TMP and FOUND.TXT to see what is where, and to put the puzzle back together. I suggest editing TEST.TEX, using the ^KR command to read in the other three files as needed, and deleting the unedited versions of any duplicated text.

All this isn't as hard as it sounds, and is well worth the effort if you have been entering complicated text or doing extensive editing.

If the effort fails, or putting the pieces back together is too complicated, you can always erase TEST.TEX, TEST.TMP, EDBACKUP.TMP AND FOUND.TXT, RENAME TEST.SAV to TEST.TEX and edit that. You'll have had at least a chance to exercise POWER!

* * *

At Compass, we use a simpler (though sloppier) form of this same procedure. Note: (1) Our WordStar file seems to start at 7849H under all conditions. (2) A save of **more** than the "lost file" from memory won't hurt. We therefore bring up POWER! and (without further ado) type

save z 7849 80

"80" is a wild guess, chosen just because it's a large number of blocks. The command

typex

will allow an inspection of the file saved to disk (here, "z"). If it isn't right, try a different routine (memory won't have been touched where the file resides). Back in WordStar, if necessary the authentic part of the file can be blocked with ^KB and ^KK from the beginning to where the junk (if any) begins, then written to a new file with ^KW. -Ed.]

See for Yourself

(JUSTIFIABLY) IRATE MEMBERS' COLUMN

Dear Compass Editor:

Of the three major elements of North Star equipment, two are very hard to best: quality and price. Matter of fact my partner and I haven't come across anything any better when those two elements are used together.

Unfortunately, the third, service, is unbelievably bad to the point where it is virtually nonexistent.

It took over four months to get North Word, General Ledger, Order Entry, TSS/A and InfoManager II. Paid for in advance of course. Just a typical example

We pay list price for all our equipment from our dealer. For this extra we expect service. (This has been verbally agreed between us.) If we don't keep calling (long distance!), we don't get answers. And when we get them, it takes a loooooong time. Like months! (He is getting better. We just sold two hard drive Advantages for him.)

We ask North Star for help when the dealer can't answer the questions. That is the deal, right? After a month or so we send a registered return receipt letter to the President of North Star requesting assistance in getting answers to our problems. It takes over a month to get any reply to that letter. Meanwhile, the problems haven't been solved. Maybe a month later an answer to one or two questions comes--not all the answers, mind you, just one or two. Apparently the balance of the questions are so difficult that a special team has been assigned and is working night and day for the solutions. North Star tells us how interested they are in the end users and how much they want to help us--but the questions don't seem like they should present such difficulties to North Star.

(In Sears, Roebuck and Co., it is felt that if a customer's problem is so serious that a letter to the president is necessary, quick action is the key. The unit manager gets the letter from the customer and another from the president. The unit manager has 24 hours to solve the problem to the customer's satisfaction and to report back to the president's office.)

So far the list includes everyone that we can pay for help from, or rather, that we HAVE paid for help from but rarely get. That is, the dealer, distributor, and manufacturer. That leaves one more avenue, a users' group.

That's you!

Attached is some correspondence that is old and self-explanatory.

The responses to our pleas for assistance have been: from the dealer, "Ask North Star"; the distributor was not asked; nothing from three letters to North Star; Mr. Michael King can and will solve it; and nothing substantial from INSUA.

The only organization that we haven't sent money to is Mr. King's, and that is the only one that answered with positive information.

The assurances of GOOD service we have received in exchange for our money have basically been lies in our documented experience. We have offered more money (\$40.00 dues to INSUA as opposed to \$20.00, just to send info). To the others we have pleaded on the phone, cursed on the phone, written letters and mailed them virtually every way the USPS offers. The result--lousy service.

My personal suggestion to INSUA's excuse of "we're all volunteers" is either don't offer to help with problems (so why join up?) or raise the dues and hire some people to help out us poor stupid owners. We have \$25,000 invested in mostly North Star equipment and are trying to make a living with it. The equipment works just fine. We don't work nearly as well as the equipment and therefore need help in using the equipment.

We've paid for this help from our dealer, the distributor, North Star and INSUA. What we want to know is, WHEN THE HELL ARE WE GOING TO START GETTING THIS HELP IN A REASONABLE AMOUNT OF TIME?

Sincerely, pleadingly, begging, hopefully (and more synonyms ad infinitum)

Warren L. Loschky
Bethel, Alaska
Membership Number BA838E

P.S. I can't help being curious about what the response would be if this letter were printed in INSUA?

* * *

Open letter to Warren L. Loschky

Dear Mr. Loschky:

Thank you for your letter of 19 May, to which I am responding several months late--giving you added justifiable cause for dissatisfaction.

Let me say first that we at INSUA will probably not be able to answer the specific questions you have asked, though we will print them in Compass in hopes that some reader may be able to help. The principal reason is not that we are unwilling, but that we do not have access to the programs you mention, and even if we did, we have neither the time nor the money nor (probably) the knowhow to analyze the problem and offer a solution.

When I tell you that we at INSUA are volunteers, this is first of all true in a strictly literal sense: as members of the board, we can accept no money from INSUA except as reimbursement for documented expenses (postage etc.) Even when we do pay outsiders, these people are not employees of INSUA: thus we can pay for services rendered, but we cannot penalize or fire anyone for services not rendered.

INSUA's contractual obligation to its dues-paying members is principally fulfilled by the publication and delivery of the Compass newsletter. The other principal expense is the mailbox operation, which handles routine correspondence but can only deal with technical inquiries by forwarding them to members of the board. We try to deal with these, but it should be understood that INSUA cannot be considered under a contractual obligation to provide answers to technical questions.

INSUA board members as a rule have access only to programs they have paid for personally. Occasionally INSUA is offered software for review; more frequently, members review software or hardware which they own individually.

No one at INSUA, so far as I know, owns any of the North Star applications programs you mention in your letter. North Star has no reason to offer them to us free of charge, and we have no reason to purchase them from North Star--not because they are not good programs, but because we have no practical need for them.

INSUA is under a general obligation to dispose of its income each year; this is generally accomplished through free offers to members, for example free disks or a special issue of Compass. We have tried and will continue to try to provide an additional service of answering technical inquiries; however, this cannot be interpreted as a contractual obligation incurred in return for the annual dues.

Having defended INSUA against the implication of contractual malfeasance, I hasten to add that we as individuals and as members of the INSUA board accept the general proposition that we have a general responsibility for addressing ourselves to technical inquiries.

Even here, however, there are limits. The 80/20 "rule" states, roughly put, that eighty percent of inquiries on any subject can be answered with a twenty percent investment of time and money; conversely, the last twenty percent of inquiries can only be answered with an eighty percent investment of time and money. As far as the present constitution of INSUA is concerned, you have asked "one percent" questions, which would require a ninety-nine percent investment of time and money to answer. I'm not certain, but I suspect that at North Star your questions would be counted at something like five percent; additionally, you are asking them to rewrite part of their software, which could require an enormous cost and effort, including obsolescence and upgrades.

Presumably you would be satisfied ultimately with a patch; and if Michael King of Island Data Computer Center in Oak Harbor, Washington, can supply you with one, I am happy to publicize his name in Compass!

By the way, individual members of the INSUA board have let North Star know the value we place on end-user support. If I

had my way, North Star would hire one person just to receive calls on **any** subject--with a clear understanding that advice over the phone would be attempted but could not be guaranteed.

If you don't renew your membership in INSUA next year, I will understand; I hope, however, to see your name on the books again.

Yours sincerely,
Alan H. Nelson, Editor

* * *

INSUA:

We are experiencing a couple of problems with North Star software and thought that maybe you could help us.

Inventory Control and Analysis Program

The key report is the sales analysis report. It has a column for unit sales but they only show up for each item, not for line,

category, and store. Those subtotals are VERY necessary. See the report example in the manual.

General Ledger

The program provides for departments. This is a very desirable feature. However, if one uses this feature, the department totals will print on the G/L listing BUT the account is not totaled.

If any of your members own these programs, have they added these subtotals and totals to the programs? If they have, can we acquire a floppy copy? We will pay a reasonable amount for this. DD or quad drive.

Sincerely,

Warren L. Loschky
BASIC
P.O. Box 1009
Bethel, Alaska 99555

#

MENU FOR 26

By Robert (Bob) Applegate

HERE IS A SHORT PROGRAM FOR BASIC USERS, CHANGING OVER FROM NUMBERS TO LETTERS SO THAT I CAN MENU 26 PROGRAMS INSTEAD OF ONLY 9. EACH TIME A PROGRAM IS ADDED, LINES 200 AND 230 WILL HAVE TO BE EDITED. IT WAS SET UP FOR 13 PROGRAMS DOWN EACH SIDE OF THE SCREEN, WHICH WORKS NICE.

```
100 REM *** GAMES
110 DIM A$(40)
120 D1$=CHR$(126)+CHR$(31)\REM - HIGHLIGHT (FOREGROUND)
130 D2$=CHR$(126)+CHR$(25)\REM - NORMAL (BACKGROUND)
140 CLS\!@(1,29),"Master Games Menu"
150 FOR T=1 TO 80\!"- ",\NEXT T\C=65
160 READ A\FOR N=1 TO A\READ A$,B$\IF N>13 THEN 180
170 !@(N+4,1),CHR$(C),@(N+4,3),A$\C=C+1\NEXT N\GOTO 190
180 B=N-13\!@(B+4,41),CHR$(C),(B+4,43),A$\C=C+1\NEXT N
190 !D1$\!@(3,1),"Enter Option Letter >"
200 !@(3,21),\G$=INCHAR$(0)\IF ASC(G$)<65 OR ASC(G$)>78 THEN 200
210 G=ASC(G$)-64\RESTORE 240\!D2$
220 FOR N=1 TO G\READ A$,B$\NEXT N\CHAIN B$
230 DATA 14\REM - NUMBER OF ITEMS
240 DATA "= Name of game","PROGRAM"
    |   |           |           |
360 DATA "= Return to Main Menu",MENU"
370 DATA "= Exit to CP/M","CPM"
380 END
```

NORTH STAR NOTES

By Edgar F. Coudal
CANSUG EDITOR

New members have responded to the Compass mention of Polaris Notes, the newsletter of the Chicago Area North Star Users Group (CANSUG), by sending in their \$5 and joining--from Louisiana, New York, Colorado, and even San Francisco. But they're not making the meetings! For the \$5, you get the monthly (more or less) newsletter and official membership. If you'd like to be on the mailing list, send \$5 to me at 627 S. Crescent ave., Park Ridge, IL, 60068.

FOR THOSE looking to add fast massive memory, the 5-Meg hard-disk and controller offered by North Star is a good answer. Your editor just hung one on his Serial #00012 and it came up flying immediately. No changes in existing BASIC programs and no changes in any GO files that load **above** 2600H. All the space below that is taken up by HDCOM, buffers, Diskette DOS, Line Editor, Command Processor, File Manager, I/O routines, Dispatch Table and so on. In other words, your 5.2 which loads at 0E00 is useless. However, the BASIC 5.4 that comes with the 5-Meg controller is exactly the same as 5.2 and runs every existing program just as though it were 5.2. The Hard Disk DOS is expanded immensely over the familiar old DOS. For one thing, it incorporates the entire MONITOR from the floppy system, with all the familiar MONITOR commands such as DH and DS and DA. Also incorporated into the Hard Disk DOS are the old Copy Disk and Copy File routines, which used to be separate utilities. That makes using CF and CD much faster.

Among the new and nice wrinkles (some demanded by North Star DOS users almost from the beginning) in the Hard Disk DOS are:

A RENAME command to change the name of a file simply by typing from DOS: RN oldname TO newname

A Multiple Copy routine that lets you copy all or any number of **selected** files from disk to disk with one command.

An OD (for Other Device) that turns on your printer while the terminal stays on, thus changing your system into the biggest, clumsiest typewriter known to man.

Various ACCOUNT management commands that in effect divide the huge Hard Disk into any number of separate smaller Hard Disks, each of which can duplicate file names used on the other "little" Hard Disks. This is very useful for applications where a number of clients are involved and their data files must be kept separate, yet use the same programs.

Other new commands show you the status of free work space in RAM and free storage space on the disk. It is startling when you check the first time and find that 9,000-plus sectors are free. N* DOS files and CP/M files can be stored on the same Hard Disk. It doesn't care, as long as you allocate space for each operating system's files when initializing the disk. You can also run up to four floppy drives (mixed, double and single sided) with the Hard Disk. But I do miss the little red lights on my drives and the whirring and clunking as those slow-stepping heads blindly stagger out over the floppies.

TECHNICAL MEETINGS: Member Bob Strand put together a direct, concise and lucid explanation of FORTH at the June technical group meeting. At CANSUG's July meeting, Treasurer Joe Alonso demonstrated the APCBASIC reviewed in Compass, Vol. III, no. 2 by Bob Stek. The August Technical Meeting of the group was held at Kaltronics, a Suburban Chicago North Star Distributor, who is opening his facility on a Sunday to demo NorthNet, Advantage 8/16, Graphics, etc. The September meeting may be held at a trade association where a Horizon is used in conjunction with video disk recording equipment to create training modules.

CONVERTING N* DOS FILES TO CP/M

is a breeze with a new common domain program offered by Sigi Kluger of El Paso. The program was to have been described in the August issue of MicroSystems along with a listing, but Kluger made it available ahead of publication on the El Paso RCPM (1-915-598-1668). We picked off a copy via Modem7, and it does everything Sigi claims. After booting CP/M, you load the program called (logically!) NSTOCPM. The program asks which drive your DOS disk is in, then asks which drive your CP/M disk is in. The program then displays a 3-item menu:

D = Display the DOS directory in DOS format.

C = Copy a file from DOS disk to CP/M disk.

Q = Quit and warm-boot CP/M.

If you select C, the program asks for the name of the file you want to transfer from the DOS disk. After you provide a legal name (no more than 8 letters) it asks for a name to give the file on the CP/M disk (no more than 12 characters; TOOLONGNAME becomes TOOLONGNAME). The DOS file is copied to the CP/M file 512 bytes at a time. It then returns to the 3-item menu for another file copy, directory listing, or termination of the program through warm-boot.

Now for the restrictions:

1. You must have two drives. Obviously, you cannot have DOS and CPM formats resident on one floppy, though you can on a harddisk.

2. It doesn't work with single density, so you must be DD.

3. UPPER CASE only for the file names (unless the DOS file name somehow collected lower case characters), including the CP/M file name. (The command letters can be upper or lower case.)

4. Z80 only. It won't work with an 8080 CPU.

5. You must have North Star's version of CP/M, because, according to the DOC file, "Special features of North Star CP/M make this program independent of the disk controller address-NSTOCPM checks the OEM code of CP/M to see if it is running on a North Star implementation...If you do not have CP/M as delivered by North Star, NSTOCPM *MAY* work. NSTOCPM will inform you if it can't run in your CP/M environment."

NORTH STAR NEWS: Pricing is now out on the 16-bit Horizon CPU board. Suggested retail is \$799 which includes the 8088 CPU and 128K of onboard RAM. The TurboDOS operating system, for use with it and which runs up to 8 users in both 8 and 16-bit CP/M applications, is offered separately at \$549.

WE HAVE HAD ACCESS to a draft of the official documentation for the CP/M version of North Star BASIC. It offers no great surprises, and seems to be a very straightforward implementation. For you NSDOS BASIC users, there are a few things to note:

1. Control-C will act just the way it does in DOS BASIC, as a "panic stop". A new "RESET" command in BASIC mimics the CP/M control-C (telling the operating system you've changed disks in a drive).

2. Input/output are performed through the CP/M BIOS. Device 1 is the LST device (printer); device 2 is the PUNCH/READER device. Device 0 and devices 3 through 7 = the console terminal.

3. There are four file type indicators in the CP/M BASIC. Type 2 indicates files in BASIC format (the familiar old type 2) and in ASCII-stored BASIC (not available in DOS BASIC, but available now so the BASIC program can be edited with any of those stupid CP/M editors such as WordStar). Type 3 is the same old Data file type 3. Type 0 denotes a file of type other than the three already mentioned. And Type -1 indicates that the file you are trying to access does not exist.

4. CP/M conventions for drive names are used, i.e. A: through P: rather than the familiar drive numbers.

5. DESTROY arguments must contain the extension, i.e. DESTROY "PROGRAM.NSB"

6. CREATE parameters include the ability to specify a file of 0 length, which I thought you always could do under NSDOS.

7. NSAVE has been eliminated. SAVE performs the function of saving newly written programs, and old ones after editing. (Shades of Ver 4.0!)

8. Dynamic allocation of file space. When saving a file which has been expanded beyond its previous size, the program automatically re-sizes the program on disk, expanding it as necessary to hold the file.

9. A utility is included which automatically converts DOS files to CP/M format.

That's a first glance at the differences.

TROUBLE converting programs from one BASIC to another? Latest issue of Access, a most interesting publication aimed at scientific applications for micros, offers some hints, to wit:

1. Syntax errors in copied lines may be generated if there are no spaces between elements of the line. 100IFX>YTHEN300 will run in North Star and TRS I and III BASICS but will give a SYNTAX error in MBASIC and others.

2. TRS BASICs do not require the THEN in an IF-THEN construction. A line such as 200 IF X=Y PRINT "YAHOO!" is acceptable in TRS BASICs, but must be rewritten 200 IF X=Y THEN PRINT "EUREKA!" in North Star BASIC. (Just the exclamatory matter in North Star examples shows the class of Horizon users.)

3. The old bugaboo: multiple statements on a line. Take this line in MBASIC: 100 IF 1=1 THEN 200 : PRINT "WILLY!" You are

never going to see WILLY! In North Star BASIC, statement execution continues on the same line no matter what the logical result of the IF-THE^N.

4. The INKEY\$ statement in MBASIC is not the same as the INCHAR\$(0) statement in NSBASIC. INKEY\$, when encountered by an MBASIC program, simply looks around to see if a key has been pressed on the keyboard. If no key has been pressed, the program continues as if the statement wasn't there at all. In NSBASIC, the program stops at INCHAR\$(0), drops everything and won't continue until you've satisfied it.

#####



LEVY ON NORTH STAR

(& COMPASS)

By Saul Levy

Congratulations on a fine Vol. III, no. 1 issue of Compass! It really hit the spot. And so soon after the previous issue, too. You and your "staff" must be working overtime to get all of this material ready to publish.

Bob Cowart's answers (pick a title) are potentially very useful. I have to strongly protest the glaring conflict of interest in his reply to the second letter. One of the Board members should have written on the value of North Star's computers in today's marketplace, NOT an employee of North Star Computers, Inc.! Please prevent such conflicts of interest in the future by allowing the editor to make such comments. Otherwise you are opening a can of worms which can't taste very good! Please clean up this problem as it leaves a black mark on INSUA!

Even so, I was happy to see Mr. Cowart's comments on this topic since I wholeheartedly agree with the views he expressed. I just don't like his being allowed to say such things. He can still tell us what North Star has up its corporate sleeves. I find that prognostications about any microcomputer company being around in one year (yet alone five) to be meaningless in this trade (even without the lousy economy).

Could I also ask that full names and addresses be given by Mr. Cowart since all of the letters he uses are to INSUA, aren't they? This policy should be used throughout the Compass to further communication among our members. In this way the six month or so delay in seeing a reply can be avoided by sending a copy of our comments to the person involved (they want help now).

I think it is clear that North Star doesn't really compete as this term is normally used. It doesn't have a major share of the marketplace which is ruled by Sinclair, Commodore, Apple, Texas Instruments, IBM, Atari, and Tandy (1982 sales in decreasing numerical order totalling 1.5405 million out of the 1.9 million

systems sold. These figures were tabulated by International Data Corporation: see InfoWorld, April 18, 1983, p.24). North Star has sold about 40,000 Horizons in five years (I haven't seen any figures for the Advantage). Compare this with the 97,500 systems that "little" Tandy sold last year alone!

All of us have heard total ignorance about North Star's computers when we mention that we own one. Horizons are everywhere, but you have to dig them out of the woodwork! If North Star's products were better known, I am sure it could sell lots more. North Star's hardware and software make most of the products from the above named companies look sick (the Horizon was designed in 1977)! This is a lot more important to me than the claims of the other manufacturers, but the cutback in advertising by North Star leads me to the opinion that it doesn't want to be one of the biggies in the marketplace with its moderately-expensive computers (IBM has managed to do this though, so size counts for a lot!).

The Horizon continues to sell well even though North Star President Charles Grant thought that the Advantage was going to replace it (see InfoWorld, November 15, 1982, p.30)! This is quite hard to understand since the Advantage isn't an S-100 bus computer! I don't think he really meant this, as it would be a fatal error for North Star to dump the S-100 bus.

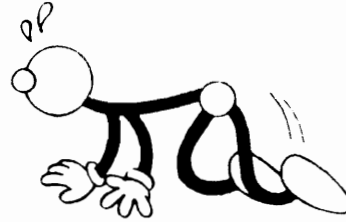
Since North Star hasn't even had a major share of the mass marketplace since maybe 1978 (which is giving them the benefit of the doubt!), I also have to disagree with one of the comments from my friend Burt Andrews. I feel that the mass marketplace supplies us with Sinclair, Apple, and TRaSh-80 products, not North Star's. There are many small companies in this field that have been selling to a limited market. Few of these companies have been as viable for such a long time as North Star!

* * *

The Editor of Compass believes conflict of interest is avoided since Bob Cowart was **identified clearly** as an employee of

North Star when he wrote the columns in question. Readers may discount or credit his opinions as they wish.

INSUA is hesitant to publish full addresses except when the writer asks for replies to his question from the general membership, or when he implies this. In other cases the INSUA board feels it has an obligation to protect the privacy of INSUA members. - Ed.]



RECLAIMING FILESPACE

Paul T. Brady

I wish to share a technique I developed to allocate file space more efficiently under North Star DOS.

The letter from Saul Levy in *Compass*, Vol. II, no. 4, renewed my faith in programmers. I, too, actually like and use North Star DOS, and also do not like complicated editors. Hence, I was getting weary of an endless stream of articles telling me how to install WORDSTAR on CP/M with exotic kinds of CRT terminals.

North Star DOS (henceforth DOS) has two fine attributes: it uses a very simple file structure, and it uses very little memory, leaving memory for what I really need, namely, application programs. I wrote my own editor, which is used by several people in a business, and is not perfect but, like DOS, is simple in its structure and usage.

However, DOS has one horrible attribute which, had I not figured how to correct it, would have caused us to scrap it. DOS always creates new files at the **free area** at the end of the disk, and does not reclaim vacant space in between other files. That is, if the disk had files A, B, C, and D all compacted and then you erased file B, DOS would create new file E immediately following D, even though the file would fit in the gap left by B.

For notation, I will refer to the beginning of the free area at the end of the disk as "the end of the disk". (In the above example, "the end of the disk" would be immediately after file D.)

Most of our programs use only one data file that remains on the disk. However,

the text editor and a program called LABEL keep churning files. These two programs, in normal DOS use, would fill up the disk in only a few weeks, requiring frequent disk compactations. To avoid this, I fixed these two programs to create files in gaps on the disk. We almost never have to do a compaction now.

I will outline the method used and give some tips on getting over the rough spots. The full listings would be nearly useless to COMPASS readers, since they are all tangled up with machine language routines that do other special tasks. This technique is not for the timid--it requires understanding of Z80 assembly language, and will take an experienced programmer a few days to install it. If you have one or two programs that do churn the disk, it is worth your while to examine this method.

Overview. The "reclaiming technique" to use existing gaps for new files is as follows: (1) Have a string variable (S\$ in this example) dimensioned 2048; this will be used to hold the file directory. (2) Create your new file of the type you want, using the standard BASIC CREATE statement, but **create it with size of zero blocks**. (3) Read the directory into S\$ (see below for how to do this). (4) Sort the directory by disk address, using assembly language. (5) Using BASIC statements, go thru the sorted S\$, looking for gaps. Keep track of the smallest gap that will hold your new file. (If there is no gap big enough, you're done; erase your zero length file and CREATE again with the proper file size; the file must be put at the end of the disk.) (6) Having found

the disk address for the gap of the right size (this will be a number from 0 to 699 for double sided disks), CALL an assembly language routine that does a DLOOK for your file name. You'll get the zero length entry you just created. Still in assembly language, change the disk location and length to what you want and do a DWGIT.

This technique is done at the **application program level** and involves no modification of DOS. Hence, any application program you apply this to will reclaim disk space, and any normal usage of DOS will continue to work as before.

This is a "best fit", as opposed to a "first fit" technique. That is, we look for the smallest gap that will hold our new file, and to do this, we have to examine **every** gap. This brings us to:

Timing. Before jumping into details, here is how long this method takes using our editor. After the user requests a "write" of the file, (1) the directory read-in and assembly language sort takes about 2 seconds; (2) the BASIC search for the best gap takes about 7 seconds (depends on how many files there already are on the disk); and (3) the DLOOK and DWGIT take another 2 seconds, after which the real business of writing records begins. Thus, there is a delay of about 10 to 12 seconds before records are written. This has been quite acceptable in several years' usage, but it could be shortened if the gap search were also done in assembly language.

Here are some pointers to get over the rough spots.

Entering assembly language programs.

Do this in BASIC at the start of your program by coding your (short) assembly language routines in decimal. You'll need a small block of memory reserved for these routines (256 bytes should be ample). Here is an example of BASIC code that will load your routines. This assumes your assembly block begins at decimal 57089:

```
FOR I=57089 TO 57321 READ X
  FILL I,X NEXT
DATA 123,106,195,37,32, (etc)
```

After this is done, 123 (decimal) will be in byte 57089, 106 in 57090, and so on.

Thus, you could code in Z80, translate this into a sequence of decimal numbers, and enter this way. The routines are short, so it's not too bad. If you have a better way, fine.

Locating the string S\$. Your assembly language routines must be able to reference and work on S\$. Whenever BASIC defines a new variable, there is an internal 16-bit location that points to one location beyond the variable. In DOS 5.1, this pointer is in 59F0, and in 5.2, it is at offset 2DF9 from the start of BASIC. (For example, if your BASIC starts at 100 hex, then the pointer is at 2EF9.) So, after you dimension S\$ as 2048, you could have $X = \text{EXAM}(2EF9) + 256 * \text{EXAM}(2EFA)$. (Of course, you must use decimal values for EXAM; I used hex for clarity in the example.) One **VERY** important point--be sure X was used prior to the DIM S\$ statement, else 2EF9 will point to one beyond X rather than S\$, as BASIC would have to define X when it executed the "X=..." statement, and the pointer would be changed before you were able to read it.

Now, X points to just beyond the end of S\$. You're almost there. (X-2048) points to the beginning of the string, and (X-2048-2, ie, X-2050) points to the 2-byte location that stores the **length** of S\$. You'll need both pointers stored somewhere. Using the FILL statements, store these two values in two 2-byte locations in the block you reserved (for example, FF00 points to length S\$, FF02 points to S\$).

Read the directory into S\$. After creating your new file of length zero with the standard CREATE command, enter an assembly language routine that does the following. Set HL to point to the length of S\$ (load HL with FF00 using the above example). Then LD (HL),0; INC HL; LD(HL),8; INC HL. You have forced the length of S\$ to be 800 hex, or 2048. HL now points to the start of S\$ (the two INC HL did that). EX DE,HL and load HL with 0000 (the disk starting address), and BC with the constant 0x0181, to specify "read high density disk #1" (refer to the DOS manual for the disk calls, you could also read disk #2, or could even pass a number

identifying the disk as part of the CALL parameter). Finally, LD A,4, where "4" is the number of track-sector blocks (each 512 bytes) to be read. Now jump to DCOM at offset 0x22 (use JP) in DOS (again, consult DOS directions). DOS will do its own RET to BASIC. You have just loaded the 2048-byte disk directory into S\$.

Sorting the directory in S\$. This is the famous "left as an exercise to the reader" section. The directory, now in S\$, consists of 128 16-byte strings. Each has 8 bytes for file name, and then 8 bytes of data. The first two of the 8 data bytes are the sort key, which contain the disk track-sector address (from 0 to 699, each indicating a 512-byte block). When you're done sorting, the first 16-byte entry in S\$ should have the lowest value for the byte-pair 9 and 10; the next 16-byte entry the second lowest, etc. WARNING!--Byte 10 has the most significant bits, 9, the least. So, first compare the pair of bytes 10, then 9.

How to sort? The simplest is the bubble sort. First compare the first two 16-byte entries; if out of order, reverse, etc, thru pair 127-128. Then do again thru pair 126-127 (after pass one, the highest is in 128 already). Isn't that notoriously inefficient? Yes, but with only 128 items it is still fast. In assembly language, it takes less than one second. In BASIC, trying to work with 16-byte segments of S\$, it would probably take about 3 minutes, so forget it. I did this in 65 bytes, you can accept this as a personal challenge.

It probably occurred to you that some of the directory entries are blank; that is, you have less than 128 files on your disk. At this point, ignore this and sort the blanks right along with the real ones. It doesn't matter if the blanks have garbage in bytes 9 and 10 and get put at random locations in your S\$ string. All that matters is that the **real** directory entries are in disk order.

Finding the smallest gap to hold your file. Your next step is in BASIC (though, as mentioned earlier, it would be noticeably faster in assembly language). You must also know that bytes 11 and 12 (lo, hi) of each entry contain the length in

512-byte blocks. Hence, the disk address (bytes 9, 10) plus the length point to the block just beyond a given file. So, for every real entry, you want to find the point just beyond the file, and for the next real entry, you can subtract its address from the prior "just beyond" point and get the gap size. You can also simultaneously keep track of the location. You must ignore two kinds of files: blank entries, identified by starting with a space " " in the file name (byte 1), and entries of zero length. (Occasionally, zero length files occur on disks. They're normally harmless, but spell disaster for the gap finding algorithm, which could be misled into thinking there is a gap where a file really exists.)

Fig. 1 is the BASIC code I use. Here, "T" is the **new file size in 256-byte blocks** (i.e., the regular BASIC block size. Note that if T is odd (e.g., 31 blocks), we'll need the next step up (32 blocks). Hence, file size will be $\text{INT}((T+1)/2)$. U\$ is a 2-byte null string; it is used to test for zero length files and ignore them. T3 is used to set the pointer to the latest gap. When done, T3 will be the disk location.

Note that in 733, we ignore the record if it begins with a space (a blank directory entry); likewise, ignore if it has zero length in 734. In line 736, we immediately exit to 740 if we have a gap exactly big enough, as there is no need to continue the search. The "THEN 765" is reached if we have a gap big enough, and this gap is smaller than the smallest gap found thus far.

Change the directory entry. Line 740 references "SUB1" and "SUB2", both assembly-language routines. SUB1 takes the file size in **512-byte blocks** and puts it into a convenient memory location, say, FF06 and FF07 (two bytes are required, FF06 is low, FF07, high). Recognizing that on entry, DE contains the CALL parameter, here is SUB1:

```
LD (FF06),DE Store the file
                size in 512-byte blocks.
RET
```

Fig. 1

NOTE: YOU HAVE ALREADY CREATED YOUR NEW FILE, HERE CALLED "F\$", OF ZERO LENGTH AND YOU READ THE DIRECTORY INTO S\$.

```

731 T1=9999\U$=CHR$(0)+CHR$(0)\T3=0
733 FORV=1 TO 128\S1$=S$(16*V-15,16*V-4)\IFS1$(1,1)=" "THEN738
734 IFS1$(11)=U$THEN738\REM FILE HAS ZERO LENGTH
735 I=ASC(S1$(9))+256*ASC(S1$(10))\T2=2*(I-T3)
736 IFT2>=TTHENIFT2<=T+1THENEXIT740ELSEIFT2<T1THEN765
737 T3=I+ASC(S1$(11))+256*ASC(S1$(12))\REMPPOINT TO BEYOND FILE
738 NEXT
739 IFT1=9999THEN741\T3=W\REM IF 9999, THEN NO GAP BIG ENOUGH
740 S$=F$+" "\J=CALL(SUB1,INT((T+1)/2))\J=CALL(SUB2,T3)\GOTO743
741 DESTROY F$\REM GAP SEARCH FAILED, CREATE IN NORMAL WAY
742 CREATE F$,T,X
743 OPEN #0,F$,J --- ETC, WRITE RECORDS, RESUME NORMAL PROGRAM

765 T1=T2\W=T3\GOTO737\REM SMALLEST BIG ENOUGH GAP THUS FAR

```

SUB2 (Fig. 2) is more complex. It first takes the new disk location in T3 and stores it in the two earlier locations, namely FF04 (low) and FF05 (high). The order is important--FF04 thru FF07 have disk address and size. Then, SUB2 looks up the file name. Where is the name? In S\$; you put it there in 740, and we know where S\$ is stored, since FF02 points to it. The string must be terminated with a space, as done in line 740 with S\$=F\$+" ".

After this assembly routine, you're finished; you have your file where you want it, and can OPEN it and write. Note: the "RET C" in the above routine is a precaution. DLOOK sets the carry if the file entry is not found. It should be there,

of course, since you just created it, but if it isn't there, issuing the DWRIT after moving random data into the directory with the LDIR would be a disaster.

Is it worth it? This looks pretty complicated, and it is. However, computers don't mind doing complicated things over and over. You, on the other hand, would have to code this only once; that might take a week or so of spare time. In our application, the computer gets used nearly every day, creating perhaps 5 or more files a day. We almost never have to do a disk compression. I feel the time I spent almost two years ago to figure this out was well worth the effort.

Fig. 2

LD (FF04),DE	Store address
LD HL,(FF02)	HL now points to S\$
LD A,1	Required for DLOOK
CALL DLOOK	This is offset 0x1C in DOS
RET C	A precaution--see note below
EX DE,HL	Now DE points to Byte 9 of entry
LD HL,FF04	HL points to block location & size
LD BC,0004	A counter
LDIR	Set the new location & size
JP DWRIT	DOS offset 0x1F. Note: DOS will do a a RET to BASIC

If you are adventuresome, try your hand at this. BUT--be sure to use a scratch disk when you try it out, as you might clobber the disk directory (as I did) if you coded it wrong. As you must know, this is a fatal error of the worst kind!

As for the North Star company, I asked them about this early after we got the computer. I wrote letters, made phone calls, to no avail. I feel that their disk storage algorithm is very poor, and the ability to reclaim gaps should have been part of DOS.

Wanted. Does anybody have a program that will copy a disk file-by-file, and not bomb out when an unreadable file occurs? The DOS copy disk routine is great until a bad block is reached, when it just dies. In this case, I'd like some way our staff can try a different program to copy the disk.

Thanks for the opportunity to share this, and let's see more on North Star DOS and BASIC.

COMMENTARY

By Larry Anuta

There are a few topics I would like to discuss, which have been brought up in the past issues of Compass.

FORMAT: I prefer the old 8-1/2 x 11 format, because I can keep the back issues in a three ring binder. There is no neat way to store back issues in newsprint format. By the way, the punched holes are not really necessary because most people can punch their own holes, and this would save some expense.

N* Advantage: Since I own a Horizon, I would prefer to see more articles on the Horizon than the Advantage; however I don't mind seeing some Advantage articles, but not at the expense of the Horizon. It may be true that the Advantage has graphics and a 8088 co-processor option, but the Horizon has the S-100 bus, which makes the option possibilities endless.

CP/M: I like the idea of CP/M articles, and the addition of CP/M disks to the library. I hardly ever use N* DOS unless I have to, and since N* has officially endorsed CP/M, I think its time has come. However, I think all such articles should be directly related to CP/M's implementation on the Horizon or Advantage.

I use S.A.I.L. CP/M, by Mike Boland of S.A.I.L. software, which is especially configured for the North Star Horizon. It can be easily configured to use the second serial port or the parallel port (with 7 or 8 bits), and increase your disk capacity to use all 40 tracks, if you have Tandon drives. Address: 1136 S. Grand St., Mesa AZ 85202 (602) 246-3154.

JRT Pascal: After waiting several months for the 3.0 upgrade, I have finally received it, only to find that it won't compile in 56K of RAM. They claim that some versions of CP/M take up too much space, and under these conditions JRT requires 58K to compile. S.A.I.L. CP/M supports 58K of RAM but my hardware doesn't. They said they will give me a refund if I return the JRT package.

POWER ON/OFF: When I first received my Horizon, about three years ago, it was one of the first Horizons with Tandon quad drives. I managed to zap two write-protected N* DOS system disks by turning the power off with the disks in the drives.

I talked to a N* technician on the hotline, a man who has helped me on several occasions, and seems to know what he is talking about. He told me that when the power shuts down, the N* controller goes out of control and may try to write

on a write protected disk. In my experience this will happen about 50% of the time.

N* ANSWER MAN: I am really tired of hearing about the lack of support from mail order dealers. In the past three years I have spent over \$10,000 on computer hardware and software which has been purchased by mail. One of these firms even burns in the equipment to ensure reliability. Try getting this kind of support from a local dealer! Two local dealers I considered buying from have since gone out of business. Two other local dealers gave me misleading advice on what to buy. We have two Computerland dealers who are so busy pushing Apples and IBMs, they have forgotten they even sold N*'s. In fact, last week they no longer had N*'s on display.

I highly resent the suggestion that we rat on our friendly and supportive N* mail order dealers to the N* Gestapo. The only reason certain computer companies don't want mail-order dealers, is they're afraid they might have to actually support their equipment themselves.

If this is North Star's new attitude toward support, when friends ask "What computer should I buy?" I'll have to suggest the IBM PC.

If I'm going to pay list price for equipment at a local store, I don't want the store salesman learning how to support it on my time. I don't want them wasting my time calling N* for the answers; they had better already know the answers!

If N* wants to operate this way, they had better initiate a training program for N* dealers.

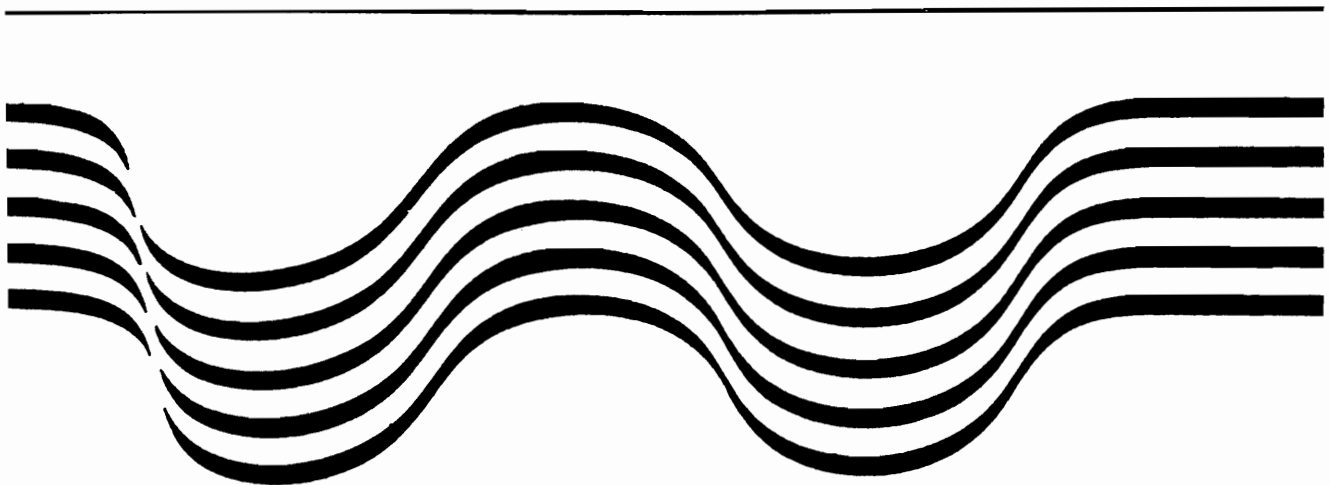
WORDSTAR: I don't really want to see any more WordStar articles. The program hasn't kept up with the state-of-the-art in word-processing.

I lost interest in MicroPro products when they stopped offering phone support and shunned mail order dealers. The program is overpriced, is not easily user-configurable, and does not have adequate printer support.

I would suggest looking into The Final Word by Mark of the Unicorn. It lists for \$300, is completely and easily user-configurable, supports most attributes of the Epson printer, most other printers, and most attributes on most terminals. In fact they even offer good support.

* * *

(Consumer Reports, September 1983, p 485, states: "Interestingly, our respondents had remarkably little trouble when they ordered by mail, as many had done. In fact, they were as satisfied with the assistance they got in setting up their computers as were people who bought locally. Reports from a sampling of government consumer-affairs offices and Better Business Bureaus around the country tend to confirm that consumers are registering almost no complaints either about retailers or mail-order suppliers of computers." --Ed.)



ON AGAIN OFF AGAIN

By James C. Matthews

Isn't it strange how often, just after you put your foot in your mouth (or typewriter), some genius will come by and tell you that it is there.

Just after my recent raving to Compass, in which I mentioned that I thought the warnings about turning systems on and off with disks in was hogwash, I found different.

At the last meeting of the C³M&BFA (Capitol City Chip Monkeys and Bit Flippers Association), I was advised of a technique which will offer a fairly good chance to clobber diskettes on power up/down.

This technique requires a system with at least one drive in the main system, and at least one separately powered drive externally. Also, the line terminator must be located in the external drive.

Now, if the external drive's power supply comes up slower than the main one,

or if you just forgot to turn on the terminating drive, all the other drives will see a low (active!!) on the terminated lines. Therefore they think that they are selected, motor-started, and write-gated. With just a little cooperation from bus noise, etc., you could be able to clobber diskettes regularly.

This was supposed to have been discussed some time ago on the HUGBB (Heath users group on MicroNet).

Hope this solves some problems for somebody!

Added thought: A blown fuse in the external drive supply would cause the same effect. Also, an open filter cap in that supply could make for some very interesting effects, since, with a common 3-terminal regulator, it would produce a stream of negative-going pulses at 120 cycles. Neat, ay wot??

COMMENTARY

By Robert Clyne

First let me make a few comments on items which have appeared in past issues of the Compass.

1. Power up/down. I generally power my Horizon up with a disk in the A: drive due to the inconvenient location of the reset switch. I usually but not always remove the disks before powering the computer down. The actual power switching is done from a power bar. I have not had any problems with this procedure.

2. The readers who complained about the coverage of CP/M and WordStar: CP/M is not the ideal operating system (nor is any other yet invented) but is good, and a lot

better than DOS, so keep up the coverage of CP/M. WordStar is also not ideal but it is a good word processor and one of the most widely used, so again, keep up the coverage.

3. To the reader with the IMSAI MIO board trying to get Lifeboat CP/M running (Compass, vol. 2, no. 4, p.17): I don't know which revision of CP/M you have but when I purchased my copy of CP/M 2.2 for North Star from Lifeboat, the manual on bringing it up indicated that there was a byte in the CONFIG.COM file which could be patched for various I/O setups and one of those listed was an IMSAI MIO with the status port at 43H and the data port at 42H. Since I have a Horizon I have never tried this to see if it works but in theory

it should be relatively easy. The copy of this manual which I have was copyrighted in 1979 by Lifeboat. I hope this is of some help.

4. To Mr. Saul Levy (Compass, Vol. 2, no. 4, p 27): I have a Tandon TM-100-2 and two Tandon TM-100-4 drives in my system and have seen several other Horizons with Tandon drives and they all shut off a few seconds after the last access, so I suggest he check his system to find the problem if his drives continue to run indefinitely when they are not being accessed. This is not a generic property of Tandon drives. There is, according to the documentation that came with my disk controller, a modification to the controller to cause the drives to run for different periods of time after the last access, but none of the options listed would cause them to run indefinitely.

5. RAM TEST FROM CP/M (Compass, Vol. III, no. 1): The code starting at MVLP to just before the final jump can be replaced by a Z 80 LDDR instruction. If you are running a Z-80 processor but are using an 8080 assembler the instruction can be faked by inserting a DB 0EDH, 0B8H at MVLP. The assembler thinks this is just data and is therefore happy but the processor executes it as a LDDR instruction. The last instruction in the listing should be a JMP instruction in Intel mnemonics rather than a JP as shown.

* * *

I have been running my system for about a year now with 63K of RAM using the prom address decoding hardware on the North Star processor board to drive the phantom line which phantoms out the 16K North Star RAM board whenever the disk controller PROM is being accessed. This system has been working well for me even though North Star is opposed to it.

In case anyone is unaware of it, North Star CP/M Release 1.1.0 will wrap itself around the prom and give more TPA than the earlier versions. The PRM still takes up 1K of memory but CP/M does not all have to go below the PROM; instead, much of it will go in the memory from EC00H to FFFFH, provided you have memory there of course.

* * *

And now for my own questions:

1. Does anyone know of a modification for North Star 16K RAM boards, circa 1977, to allow them to work with IEEE 696 disk controller, processor, etc. boards? Any help would be appreciated.

2. Does anyone have any patches for WordStar to achieve microspacing for justification with a dot matrix printer? I need it for an IDS Prism, but if anyone has done it on an Epson or other dot matrix printer I can probably convert it to the Prism.

3. I see from message #02546 on Proxima that North Star has come out with a new version 2.1.0 of DOS. Will INSUA be distributing this? (Yes--this was offered free earlier this year, and is henceforth available as INSUA Disk Library #1024 (\$10.00) --Ed.)

Messages left on Technical CBBS in Dearborn, Michigan, phone (313) 846-6127 will reach me.



PEKIN FOR CHURCHES

by

Robert (Bob) Applegate
3133 Lower Mt. Road
Sanborn, NY 14132

I have written a program called PEKIN, for church treasurer work. This consists of one main program and a number of secondary programs. For this program to be used by another church, a lot of deleting and filling in would be required. I have added REM statements to make it easier to understand. It now runs under NORTH* but in January 1984, it will be running under baZic on CP/M.

The Pekin program is set to run with two disks. Drive 2 contains all the files, and a new disk is used each year. Because of the ERRSET procedure which checks months (on drive 2), a program

called CR.FILES must be run first from the main program. This procedure creates the more than 50 files on drive 2. After the initial setup, everything will run from the PEKIN program. The cursor control was designed around the Hazeltine 1500.

I am sure that though you won't be able to use the program as it is, you will be able to get some ideas that could help you with church work.

Fig. 1 shows the main program. Even though this is of no use without the others, it gives an idea of how PEKIN works.

I will be happy to mail a copy of the complete program to anybody who will send two blank diskettes and a self-addressed stamped envelope to me at the above address. (This offer is good until the end of Spring 1984; thereafter, please send a written inquiry before sending blank diskettes.)

```
100 REM *** PEKIN           as of August   8, 1982
105 REM                   updated December 22, 1982
110 REM
115 CO$=CHR$(126)+CHR$(28)\REM CLEAR SCREEN
120 EO$=CHR$(126)+CHR$(25)\REM NORMAL MODE (SCREEN)
125 E1$=CHR$(126)+CHR$(31)\REM HIGHLIGHT MODE (SCREEN)
130 REM REINSTALL - ERRSET 620,Q,Q1 - AFTER STARTING NEW YEAR
135 OPEN #1,"MONTH,2"\READ#1,A4\CLOSE #1\REM MONTH OF REPORT
140 Z$="MABETPSF"\N=40\REM REINSTALL - ERRSET - AFTER STARTING NEW YEAR
145 DATA 1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22
150 DATA 23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40
155 DIM A(N),B(N),C(N),D(N),A$(20)
160 DEF FNC$(X$,Y$)=CHR$(126)+CHR$(17)+CHR$(C+1)+CHR$(R-1)\REM CURSOR CONTROL
165 FOR I=1 TO N\READ A(I)\NEXT\!CO$\!
170 C=18\R=3\!FNC$(X$,Y$), "*****"
175 R=4\!FNC$(X$,Y$), "*" *"
180 R=5\!FNC$(X$,Y$), "*" *"
185 !E1$\C=33\!FNC$(X$,Y$), "PEKIN CHURCH"\!EO$\C=18
190 R=6\!FNC$(X$,Y$), "*" *"
195 R=7\!FNC$(X$,Y$), "*" *"
200 R=8\!FNC$(X$,Y$), "*" Enter in Which Month ..... *"
205 C=52\!E1$\!FNC$(X$,Y$), "M"\!EO$\C=18
210 R=9\!FNC$(X$,Y$), "*" Month is No. *"
215 C=36\!E1$\!FNC$(X$,Y$), A4\!EO$\C=18
220 R=10\!FNC$(X$,Y$), "*" Enter in New Amounts ..... *"
225 C=52\!E1$\!FNC$(X$,Y$), "A"\!EO$\C=18
230 R=11\!FNC$(X$,Y$), "*" Enter in New Balance *"
235 R=12\!FNC$(X$,Y$), "*" and Monthly Receipts .. *"
```



```

240 C=52\!E1$\!FNC$(X$,Y$),"B"\!EO$\C=18
245 R=13\!FNC$(X$,Y$),"*      Edit Entries .....          *"
250 C=52\!E1$\!FNC$(X$,Y$),"E"\!EO$\C=18
255 R=14\!FNC$(X$,Y$),"*      Test Print-out .....          *"
260 C=52\!E1$\!FNC$(X$,Y$),"T"\!EO$\C=18
265 R=15\!FNC$(X$,Y$),"*      Print Treasurer's Report .    *"
270 C=52\!E1$\!FNC$(X$,Y$),"P"\!EO$\C=18
275 R=16\!FNC$(X$,Y$),"*      Start of New Year .....          *"
280 C=52\!E1$\!FNC$(X$,Y$),"S"\!EO$\C=18
285 R=17\!FNC$(X$,Y$),"*      Finished Program .....          *"
290 C=52\!E1$\!FNC$(X$,Y$),"F"\!EO$\C=18
295 R=18\!FNC$(X$,Y$),"*                                           *"
300 R=19\!FNC$(X$,Y$),"*                                           *"
305 C=25\!E1$\!FNC$(X$,Y$),"input >"\!EO$\C=18
310 R=20\!FNC$(X$,Y$),"*                                           *"
315 R=21\!FNC$(X$,Y$),"*****"
320 !CHR$(7)\C=31\R=20\!FNC$(X$,Y$),
325 O$=INCHAR$(0)\FOR O=1 TO 8\IF O$=Z$(O,0) THEN EXIT 330\NEXT\GOTO 325
330 ON O GOTO 400,415,335,350,340,345,360,395
335 CHAIN "BALANCE"
340 CHAIN "TEST"
345 CHAIN "TREASURE"
350 OPEN #1,"EDIT1,2"\A5=1\WRITE #1,A5
355 CHAIN "EDIT"
360 !CO$\OPEN #1,"CHUR2-12,2"
365 READ #1,B0,B1,B2,B3,B4,B5,B6,B7,B8,A,B,C,D,E,F,G,H\CLOSE #1
370 OPEN #1,"HOLD"\WRITE #1,F\CLOSE #1
375 C=15\R=12
380 !FNC$(X$,Y$),"Put New Disk in drive #2 and then press <RETURN> >"
385 C=64\!FNC$(X$,Y$),\L$=INCHAR$(0)
390 CHAIN "CR.FILES"
395 !CO$\END
400 !CO$\R=12\C=29\!FNC$(X$,Y$),\INPUT"Enter month of report ",A4
405 OPEN #1,"MONTH,2"\WRITE #1,A4\CLOSE #1
410 CHAIN "PEKIN"
415 !CO$\IF A4=1 THEN GOSUB 590
420 IF A4=1 THEN 465
425 A4=A4-1\GOSUB 595
430 OPEN #1,"CHUR1-"+A$+",2"
435 A4=A4+1\GOSUB 595
440 OPEN #0,"CHUR1-"+A$+",2"
445 FOR X=1 TO N
450 READ #1%(X-1)*15,B
455 WRITE#0%(X-1)*15,B,NOENDMARK
460 NEXT\CLOSE #1
465 INPUT"ACCT. NR. AND EXPENDITURE ",X,Y
470 IF X=0 THEN 515
475 FOR I=1 TO N
480 IF X=A(I) THEN EXIT 495\NEXT
485 !"ACCT. NR. IS IN ERROR, TRY AGAIN"\!
490 GOTO 465
495 READ #0%(X-1)*15,B\READ #0%(X-1)*15+10,D
500 WRITE #0%(X-1)*15,B+Y,NOENDMARK
505 WRITE #0%(X-1)*15+10,D+Y,NOENDMARK
510 GOTO 465
515 T4=0\T5=0
520 FOR I=1 TO N

```

```

525 READ #0%(I-1)*15,B(I)
530 T4=T4+B(I)\NEXT
535 FOR I=1 TO N
540 READ #0%(I-1)*15+10,D(I)
545 T5=T5+D(I)\NEXT
550 IF A4=1 THEN 575
555 A4=A4-1\GOSUB 595
560 OPEN #1,"CHUR1-"+A$+",2"
565 READ #1,O,P\T4=T4+P\CLOSE #1
570 A4=A4+1\GOSUB 595
575 OPEN #1,"CHUR3-"+A$+",2"
580 WRITE #1,T4,T5\CLOSE #1
585 CHAIN "PEKIN"
590 GOSUB 595\OPEN #0,"CHUR1-"+A$+",2"\RETURN
595 IF A4=1 THEN A$="1"\IF A4=2 THEN A$="2"\IF A4=3 THEN A$="3"
600 IF A4=4 THEN A$="4"\IF A4=5 THEN A$="5"\IF A4=6 THEN A$="6"
605 IF A4=7 THEN A$="7"\IF A4=8 THEN A$="8"\IF A4=9 THEN A$="9"
610 IF A4=10 THEN A$="10"\IF A4=11 THEN A$="11"\IF A4=12 THEN A$="12"
615 RETURN
620 !CO$\R=12\C=12!\FNC$(X$,Y$),"Put Pekin Files in drive #2 ",
625 !"and then press <RETURN> key. ",\U$=INCHAR$(0)\CHAIN "PEKIN"
630 END

```

NORTH STAR ANSWERS

by Bob Cowart

(Bob Cowart, formerly an employee of North Star, has now taken up a career as a free-lance writer. -Ed.)

Q: Is there a simple, or even not so simple way to use DOS assembly language programs written for the Horizon on an Advantage? -A.P.S. Virginia

A: That all depends on the program. If your programs are hardware dependent, that is if they expect to address I/O (input and output) ports, the disk controller, printer, etc. in specific logical locations, then you haven't got a chance. The layout of the Advantage is very different from that of the Horizon in this way. In addition, if your program relies on "system calls" to the DOS, you will also run into trouble since the system function locations are located differently in Advantage GDOS than in Horizon DOS.

On the lighter side, if you have an assembly language program which is independent of either of these limitations you should be safe. Since both machines use a Z-80 microprocessor, the actual assembly language code will execute in either one. You may just want to try running a Horizon DOS program in the Advantage to see what happens -- it may work.

But beware. Occasionally, running a program not designed for your machine can have catastrophic results. For example, I have seen all the data on an Advantage hard disk system lost from running a CP/M public domain program called STATUS. his program polls all possible input/output ports to determine which ones are active. Apparently the hard disk controller, which is I/O mapped, responded ungraciously to the poll, wiping out the hard disk directory.

Q: Is it possible to set up a Horizon multi-user system (with 18 Megabyte hard disk)

where each user has simultaneous access both to programs running under HDOS and also to ASP programs like Northword? -J.F. California

A: Sorry about that. Can't be done because the TSS/C operating system allows only CP/M and HDOS to work together. The TSS/A operating system allows multiuser operation of the ASP software only.



Q: When running the RECOVER program (to recover files from diskette to hard disk under the HDOS backup scheme) if I give the pathname of an existing file on the hard disk as the destination, I get an error message 021, "Can't create new file". To make it work I have to invent a non-existent file name to recover to, and then use CF (copy file) to copy the recovered file into the correctly named file. Is this a bug? Is there an update?

A: This is not a bug. This was done intentionally to prevent accidentally recovering into an existing file which you might not want to clobber. Although this method is a little more cumbersome, it at least prevents catastrophes such as wiping out a huge database or document file. A faster technique is to create a new account, recover your files into the new account, then do an account delete (the original account), and an account rename (new name to old name). This, of course, only works if you don't mind deleting the entire old account.

Q: I'd be very interested in hearing your thoughts or North Star's corporate thoughts about the Horizon's future. For those of us who are unlikely to be using Z-80 CP/M 2.2 in 1985, a little systems integration would go a long way. If the Horizon is now a dead end from a commercial point of view (I realize North Star has to make money to survive) why not develop a 16/32 bit machine compatible with old N* so much as possible? I'd like to keep as much of my Horizon as possible through 1990 if IEEE-696 modifications don't destroy the

machine and new state of the art products can be incorporated into rather than replacing the Horizon (especially CP/M 3, 16/32 bit CPU, graphics). W.L.- Tennessee.

A: For the 40,000+ users of Horizons, good news. In their continued support of the old reliable Horizon, North Star has come out with the Horizon 8/16 upgrade. This allows either single or multiuser (up to 8 users) to run CP/M 80 and/or CPM/86 software on existing Horizons. Each user has his/her own microprocessor board (either Z-80 or 8088) and memory. This means the system will be much faster than the current TSS/C, TSS/A arrangements. In addition, the operating system is actually TurboDos resulting in an overall increase in operational speed as compared to MP/M and MP/M, 86. The upgrade boards available are as follows: (prices as of June 15, 1983)

8 bit CPU with 64K RAM
\$599
16 bit CPU with 128K RAM
\$799
348K RAM for 16 bit CPU
\$1199

TurboDos for multiuser 8/16
\$549

Some Horizons (older ones) may need the power supply upgraded to support the addition of more than a few boards, since these boards have lots of circuitry on them. Any Horizon serial number of 40,000 and up needs no modification, and may support up to 8 additional boards total. Check with your dealer or North Star for more details.

STRING FORMATTING

By Henry M. Spelman III

NorthStar BASIC has a very simple and easy to use print formatting scheme. The following notes may help clarify its use.

The largest number that can be used in a format string is 32. A larger number (!%33I for example) will give a "FORMAT ERROR" message. This can be a tough error to find and correct, as one normally looks everywhere else on the line.

The manual says that including a "#" in the format string makes that format the default format. There are no examples, and it took me a while to figure out how to use this feature.

The string "!%#2I," (as a separate statement) will force the printing of all following numbers as two digit integers, right justified. Longer integers or real numbers will generate a "FORMAT ERROR" message. Similarly "!%#C10F2," (as a separate statement) will print all following numbers as right-justified real numbers with dollar sign, commas and two figures to the right of the decimal point. The comma suppresses the line feed that would otherwise follow the print statement. If you are using the format string with an argument (!%#2I, N), you would use a comma after the "N" if a linefeed was not desired. To return to the default free format, just include "!%#," in the program (as a separate statement) when you are through with your special format.

The format feature can be used to force numbers entered into a string with the "STR\$" function to fall exactly where you want them in the string. It will also prevent a dangerous error that can occur when translating a number to a string.

Under the normal free format, "N\$=STR\$(N)" will input a space as the first character in the string, and the digits of the number as successive characters. If N is too big to fit in the string, it will be truncated at the right without warning. If LEN(N\$)=5 and N=50 then N\$= " 50". If N=50000, then N\$= " 5000". Program 1 will

illustrate what happens with free format. Running the program will give:

```
5
50
500
5000
5000
5000
5000
READY
```

Note that there is a space before each number, the numbers are left-justified, the last item in the data statement is truncated at the right when the number has five or more digits, and there is no error message.

Now, to create Program 2, we add statement 25 to change the default format and statement 75 to return to the free format. Running the revised program gives us:

```
5
50
500
5000
50000
FORMAT ERROR
READY
```

The numbers are now right-justified in the string, so that when you print the numbers from the string, they will fall in a predictable place in the printout. The error is flagged, so that it won't be a time bomb waiting to give an unexpected and rather subtle bad result.

I use formatting like this when I enter a customer number, which is generated by the program, to a name and address file. When a mailing label is printed, the customer number, which might be A\$(21,25), will appear in a predictable format no matter how many digits it contains.

Program 1.

```
10 DIM N$(5) \ REM Five characters in the string
20 DATA 5, 50, 500, 5000, 50000, 500000 \ REM numbers to put in the string
30 FOR M=1 TO 6
40     READ N
50     N$=STR$(N) \ REM fill string with number "N" changed to a string
60     ! N$ \ REM Print the string
70 NEXT M
80 END
```

Program 2.

```
10 DIM N$(5) \ REM Five characters in the string
20 DATA 5, 50, 500, 5000, 50000, 500000 \ REM numbers to put in the string
25 !%#5I, \ REM Default format is an integer of 5 digits
30 FOR M=1 TO 6
40     READ N
50     N$=STR$(N) \ REM fill string with number "N"
60     ! N$ \ REM Print the string
70 NEXT M
75 !%#, \ REM Return to free format
80 END
```

☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆☆

VENDORS' COLUMN

In this column we cite material sent us unsolicited by various vendors. Informational only--statements and claims are the vendors'.

NORTHSTAND

NORTHSTAND is a sturdy steel floor stand designed to support your North Star Horizon on its side. Save valuable desk space by mounting your Horizon on Northstand. Insert floppy disks into your drives with the label side up. You furnish the Horizon; we furnish the stand, wheels, glides, and bolts. 150.00 plus tax and freight. JKS & Associates, 2011 Key Blvd., Suite 3, El Cerrito, CA 94530.

CLEO

There's a new source of employment information as technically advanced as the job openings it offers. It's CLEO. That's short for Computer Listings of Employment Opportunities. Talk to CLEO with your

personal computer. Describe the job categories, companies, or the geographic areas that interest you ... CLEO guides you at every step with explicit online instructions. And best of all it costs you nothing to use except the price of a phone call. CLEO access: (213) 618-8800. Telephone number for assistance: (213) 618-1525.

CompCare

CompCare Computer Systems is a North Star dealer and also an ADS licensee with North Star. We have developed comprehensive software for the veterinary field and are now seeking to increase our retail hardware sales. We have many possible applications presented to us and are very interested in finding users who have done various different hardware applications successfully.

Fred Kaun
CompCare Computer Systems
815 Witzel Avenue
Oshkosh, Wisconsin 54901

Micro Design Associates

New S-100 Disk Controller. Runs 5-1/4 inch and 8 inch drives simultaneously. All signals compatible with IEEE 696 (C-100). Totally compatible with North Star controller. Reads and writes formats of the following systems: IBM 3740, IBM 37, IBM PC, Cromemco, Televideo, Morrow, Osborne, Kaypro, Lobo, Xerox, TRS-80 models 1-4, 12, 16, and most other soft-sectored systems.

When used in combination with North Star controller, your computer will read 5-1/4 inch hard-sector North Star in addition to 5-1/4 inch softsector and 8 inch softsector drives. 8 or 16 bit data bus. Comes with free BIOS and instructions for your CP/M. Can be used with CP/M 2.2 or 3.0. Comes with a format program and other utilities. Price \$299.00. Assembled and Tested.

Micro Design Associates
25 S. 8th St.
Columbia, Mo. 65201
(314) 874-2777

Horizons

North Star Horizon Computers. Preowned and thoroughly checked before shipping--guaranteed to work perfectly on receipt.

32K HRAM
MDS-A-D Floppy Disk System
2 SSDD Drives
100,000 day real time clock

Quantity available: 14 units
Delivery: 10 days
Price: \$1,275 each
Contact:

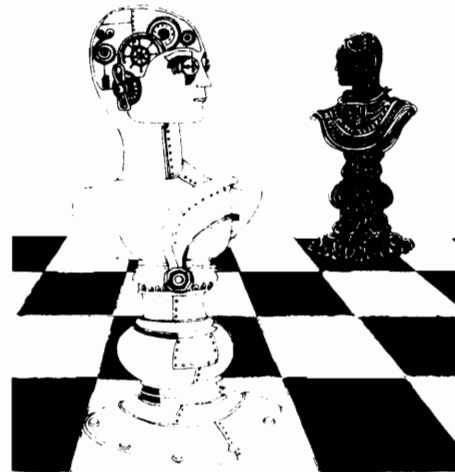
R. Kurt Geist
6770 Cayton
Houston, TX 77061
(713) 645-4427

PH Associates

We can provide you with quality Winchester disk drives for less. These are compatible with both the Advantage and

the Horizon, and are available now in 5, 10, and 15 megabyte sizes. Also available are the large capacity MARK series drives of 20, 33, and 46 megabytes.

PH Associates
8720 Old Courthouse Road
Vienna, VA 22180
(703) 281-5762



ANSWER

The ANSWER is a low-cost device that enables the use of any 300 baud modem (acoustic or direct connect) in the auto-answer mode. We anticipate that our product will be particularly useful for club members wishing to participate in bulletin boards or electronic mail. The ANSWER is also intended for anyone who wishes to access their home computers while at work or away from home.

The price of the ANSWER is \$69.95 with a \$2 charge per unit for shipping and handling. Special discounts for quantity orders: 5-9 for \$64.95, 10 and more for \$59.95

Conestoga Data, Inc.
8403 - 121st Place North
Largo, FL 33543
(813) 531-8517

CLUBS

The following list of North Star Users Groups was published in Vol. II, no. 2, of North Star Notes. The last two are new groups.

Philadelphia Area

North Star Users of Delaware County, c/o G. Holdsworth, Box 298 Newton Square, PA 19073, (215) 356-8550, (215) 566-2696 (Evenings).

Chicago

Cache, Doug Robson, 2452 West Grace, Chicago, IL 60618, (312) 463-1833.

Chicago Area North Star Users Group, Ed Coudal, 627 S. Crescent Avenue, Park Ridge, IL 60068.

Boston

The Boston Computer Society, North Star Users Group, Three Center Plaza, Boston, MA 02108, (617) 367-8080.

Ashtabula

Ashtabula Area Computer Club, c/o Grant Johnson, 2322 West 13th Street, Ashtabula, OH 44004, (216) 964-3212.

New York, NY

CP/M Users Group, Editor: Bonita E. Taylor, 1651 Third Avenue, New York, NY 10028.

New York City North Star Users Group, Dr. Jeremy Shapiro, 302 West 86th Street, New York, NY 10024, (212) 496-6050.

Seattle

North Star Computer Society, Inc., P.O. Box 311, Seattle, WA 98111, Chuck Langenberg, Pres. (206) 285-0259.

Raleigh

Triangle North Star User Group, c/o Ted H. Emigh, Box 5487, Raleigh, NC 27650.

Amarillo

North Star Users, Wanda Johnson, Route 5, Box 91, Amarillo, TX 79118.

East Syracuse

CHIPS, Wyn Babcock, 7260 Colamer Road, East Syracuse, NY 13057, (315) 656-3111.

Netherlands

HCC—North Star Users Group, Secretary: Rob Verdoorn, Ysdans 16, 2907 AZ Capelle A.D. Yssel, The Netherlands.

Minneapolis

Computer People Managing, 7720 Bush Lake Road, Minneapolis, MN 55435.

Houston

Our company would like to organize a North Star Users Association in Houston, Texas. I have been unable to locate an active organization of North Star users in this area. Please publish a notice in the next issue of Compass that anyone interested in a North Star Users Association should contact:

Ed Hall
APPLIED METEOROLOGY, INC
(AMI SYSTEMS)
9000 Southwest Freeway, #326
Houston, TX 77074
(713) 777-0106

Contra Costa County, California

I am in the process of forming a North Star Users Group in the Contra Costa County area. This will be an active users' group, meeting once every three to four weeks. We would like to hear from any member who would be interested. Please call me.

Gary Leung
2406 Talavera Drive
San Ramon, CA 94583
(415) 828-7269

LETTERS TO THE EDITOR

INSUA:

I have a North Star Horizon operating under TSS/C ... Do you know of anyone in the Association who has implemented CP/M 3.0 on his/her machine?

Stan Fuller-Bey
Toronto, Ontario
Canada

INSUA:

About Compass Vol. III, no. 2, "Multi-line User-Defined Functions": At my work, I am using similar functions. From my experience, there is a thing to add to it.

When using cursor control function, you may experience that the cursor is out of control. This symptom is caused by the BASIC program's automatic CR insertion.--There is a column-number counter in the BASIC program. When any character is PRINTed the counter is incremented even though CHR\$(13)=CR. When this counter gets the value of line length (normally 80), the BASIC program outputs a CR by itself. This counter is initialized when PRINT statement not terminated with "," is executed.

To prevent this, you just insert a statement "LINE 165,0" in the beginning of the program. The second argument "0" will disable the automatic CR insertion. This process is mentioned in the explanation of the LINE statement.

I hope this can help somebody!

Fumihide Nakamura
Anaheim, CA

INSUA:

I was pleased to receive the latest issue of Compass. Another good job--

My particular interest at present is in interfacing my system (IMSAI & MDS) to control motor positions and A/D converter inputs. Do any of your members do much interfacing with scientific instruments for measuring (say) temperature, volts, frequency, or other parameters? The

computer can calculate certain functions and control experiments, etc. I would like to find someone close with similar interests.

Bob Melvin
Melvin Sales, Inc.
P.O. Box 5283
San Mateo, CA 94402

(We have run several articles and notes on A/D in previous issues of Compass. On-line Instrument Systems, Route 2, Jefferson, Ga. 30549--(404) 367-9191, specializes in instrumentation and scientific application for the Horizon. Any further comments from INSUA members? - Ed.)

INSUA:

INSUA members may be interested in an article appearing in Microsystems, Vol. IV, no. 5 (May 1983) by Richard Feldman. Feldman explains clearly how to upgrade a North Star 32K RAM to a 64K board (though you must lose 4K for the disk controller). I have looked over his changes and don't see any problems, but I have not yet had the time to try them. From reading the article, you also see how far memory production has advanced in the last 5 years.

Keep up the good work.

Tom Belliveau
Montreal, PQ, Canada

INSUA:

Re Compass, Vol. III, no. 1, pp 24-5, "40 Tracks":

I have tested all the patches on p. 25, and they work just fine. I have also discovered that SA 400 have only 35 tracks and that SA 450 DD/DS have 40 tracks on each side and run at the same speed as the SA 400.

The problem I had was in transferring the files from my DD/DS-35 tracks format to the new DD/DS 40 tracks format. Here is how I did it:

- Using my 35 tracks DOS and CF program I transferred all the files from my DS disquettes to my SS disquettes (from DA 450 to SA 400) thus making sure that no file exceeded the 350 blocks of my single-sided disquettes.

- Then using my new 40 tracks DOS and CF program, I initialized new 40-track disquettes and transferred all the files from my single-sided 35-track disquettes to my double-sided 40-track disquettes (this was necessary because after 35 tracks the old system would start its way back toward track 0 and with the new system this would happen only after 40 tracks.

I hope this might be of some use to others who might be faced with the same problem.

Daignault and Associes
Drummondville, Quebeque, Canada

INSUA:

I am running an MX-80 printer that has a lot more functions than WordStar can drive. If you know of any publication telling how to patch in multiple additional functions, I would surely appreciate knowing of it. I already have used up the four user-definable codes in the WordStar program.

Edward Cook
Portland, Oregon

INSUA:

How do I send all eight bits to my Epson FX-80 printer through the North Star Horizon's parallel port? My Horizon now sends 7 bits and uses the eighth bit to strobe the printer.

Dominick Andrisani
West Lafayette, IN

(Both of these problems are easily solved--see articles on installation of Epson and WordStar in Compass, Vol. II, no. 3. See also Oliver C. Stokes Jr's "Run the Parallel MX-80 with North Star 5.2 DQ" in October 1983 Microsystems. --Ed.)

INSUA:

I was very pleased to read J. Burdeane Orris's article on Multi-Line User-Defined Functions in Compass, Vol. III, no. 2. These exact capabilities of N* BASIC and the speed of N* DOS are the reasons we have remained N* dealers and develop our turnkey systems around N* products. We use functions to format screens, input and edit data, search files, issue error messages, etc. In fact, we do so many things with functions that many of our programs are simple skeletons of logic that make many function calls. Mr. Orris has made an excellent presentation to INSUA members and I encourage them to practice the use of functions.

Please add my concurrence to James Matthews commentary. Let's devote as much space as possible to "pure" North Star topics.

How about some articles on North Star's HDOS and TSS/C. I realize that many members don't have hard disks yet, but the price is now very attractive and many of us use them extensively in business and other applications. There are certainly many members more qualified than myself, but I would contribute ideas and "program quickies" for a periodic column on these products. I certainly would like to communicate with other users of these products and welcome mail ideas or questions.

Sincerely,
Terry W. Hough
Datagroup, Inc.
P.O. Box 13309
Greensboro, N.C. 27405

INSUA:

In Compass, Vol. III, no. 2, a question was asked of Robert Cowart by P.W.W. of Manchester, Mo. regarding FONT changing programs for an Advantage. I purchased such a program from my local North Star dealer: it was entitled SOFTFONT. It is written and optimized for the dual-quad Advantage and works beautifully. The dealer is:

MICRO WORLD COMPUTER STORE, INC.
1901 Vestal Parkway East
Vestal, NY 13850

Sincerely,
James Baumgartner
Conklin, NY

INSUA:

The letter from Larry Feigen in Compass, Vol. III, no. 2, prompted me to offer some other solutions to the unavailability of RAM chips for the 32K RAM card.

I had a similar problem with my 32K card. The -16 volt line on my Horizon motherboard blew out, and took 14 RAM chips on the 32K card with it. After a futile effort to find the correct replacements for the blown chips, I eventually arrived at a solution which was both easier and cheaper. The 8K chips used on this card are really 16K chips on which one or more bad bits are ignored by tying an address line either on or off and using only half the chip. 16K RAM chips (4116's) work just fine as replacements, and the price and availability are fantastic, since they are used in TRS 80's, Apples, and most other micro's of two to three year old vintage.

If you need a quick fix for a problem you may want to borrow chips from unused portions of your card. If you don't have any unused addresses on the card to steal from as Mr. Feigen did, you can also borrow the parity chips and move them into other positions. This will cause a constant parity error on the card but most of my software doesn't use the parity feature anyway.

Martin Brown
Pleasant Hill, CA

INSUA:

1. Re "ON-OFF" controversy: I'm a dedicated "leave-'em-in-er." This never caused me any problems, from MDS-DOS rell v2 & CP/M 1.4 through every

revision, then a Horizon and recently an Advantage as well. Unfortunately, only one !@#% !! thing around here behind the crashes and lost programs and data: Confounded Pilot Error! Except:

2. Re HD18 timing disc "bug": For several months this past Spring I've been getting mystery CRC errors on the 18M Marksman disk. Finally, after spending many futile and frustrating hours trying to recover files and data (without much luck), I hit upon the idea of contacting my dealer, found out about the timing disc bug, and ultimately lucked out on the lot. Whew!

Here's my quibble: Not only has North Star known about this problem for some time; they considered it serious enough to redesign the 18M cooling fan arrangement to include dust filters and have made a kit available to their dealers to update older drives! But they chose not to advise registered 18M owners of the problem, and how to deal with it successfully. Which to my way of thinking is a shoddy piece of work. What do you think?

And that's why I'm very glad INSUA's around, and off, it seems, to a fine start.

David Anderson
Buffalo, NY

INSUA:

I would very much appreciate additional information on various hard disc systems that have been implemented on N* Horizons including the Morrow, Corvus (Constellation and Mirror backup), and, of course, N*. Please include information on good points as well as problems and serviceability.

With regard to Mark Perkel's request for input on modem programs, I would like one which implements all the bells and whistles of the D.C. Hayes Smartmodem including Morse code. If the program could also keep track of time of day for auto-dialling, that would be better yet.

If anyone has implemented the H510-4 board for additional devices and has the software I/O routines running and

debugged, I would be willing to pay for copying.

One final question: I have an 88-SPM Clock Module for S-100 bus computers by International Data Systems which works fine in my IMSAI when I was running single-density North Star. Now, in my Horizon, this board seems to lock up the bus and the computer won't function. Any suggestions?

Ivan L. Fillmore
Hanover, PA

INSUA:

We use a North Star Horizon running CP/M to keep records and manage correspondence for a small mail-order book business. In future newsletter articles, I would naturally be most interested in topics concerning the Horizon and CP/M, but the other material I've seen is also interesting.

By the way, has anybody been able to completely relocate the disk controller 4K higher in memory to allow a 62K CP/M system on the Horizon? Some programs would work much better with a larger TPA.

Philip E. Bond
Spring Valley, NY

(For information on moving the disk controller, see Compass, Vol. I, no. 3, or contact Bob Hogg, Micro Computer Devices, 25651 Minos St., Mission Viejo, CA 92691 Tel: (714) 770-2168; also, see Robert Clyne's "Commentary" in the present issue. Finally, October 1983 Microsystems has an article by John H. Gillespie, "A North Star Improvement," on constructing a 61K CP/M system and increasing available RAM on the North Star Horizon. Our thanks to Dan Konigsbach for calling this to our attention --Ed.)

INSUA:

In a letter to Compass a while back, I wistfully asked for useful articles pitched more to us "middle-school" users.

Wow. Just finished Volume III, no. 2, and between the "Phantom Null" note which clears up a long-standing mystery

for me, and Mr. Orris's tremendously useful article on using user-defined functions, this issue alone was worth the price of a year's membership in INSUA.

In addition, the Chicago User Group's programming quickies on choosing "faster" operations let to a massive rewriting of a long BASIC program I am using, with visible beneficial result.

Finally, the APC BASIC review by Bob Stek led to my ordering this "Mercedes" of BASICs--it is just what I have been looking for. And "Tiny" Matthews's commentary on holding down the CP/M content rates a hearty "hurrah!"

This paean of praise is not intended particularly for publication--use the space for something better. I just wanted you to know that the Newsletter's usefulness to me has increased by an order of magnitude, and I am grateful.

Sincerely,
Bill Harmon
San Francisco, CA

(C'mon--you didn't seriously expect us not to print this letter, did you? -Ed.)

INSUA:

I think it would be a **great advantage to the members** if our membership list were sold. This would be an automatic screening process to receive North Star oriented advertising. At least we wouldn't be beaten to death with IBM and Apple (Wormy). Besides, we are already on hundreds of mailing lists. One more won't make any difference.

Regards,
Thomas M. Brown
Nashville, TN

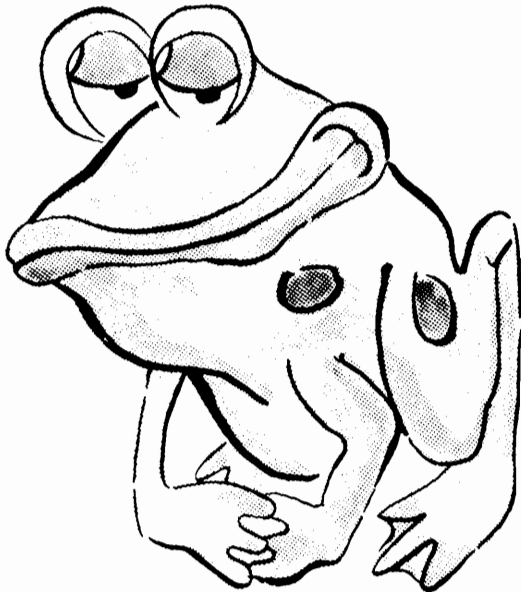
(The Board of Directors has consistently taken the position that the membership list with names and full addresses constitutes a body of information which must be kept inhouse in order to guarantee the privacy of members who might want the information protected. The Board is considering a move to solicit advertising from dealers in hardware and software compatible with North Star products. More on this soon. -Ed.)

INSUA:

People often complain about the North Star reset switch being inconveniently located on the back of the cabinet, and those who are handy with wiring may move the switch to the front. My solution is to turn the box 90 degrees so that the left side is at the back. This brings the reset switch to a much more convenient location and also makes it easier to get at the cables.

I have the computer box to the right side of my screen and keyboard. This arrangement leads to inserting disks with the left hand, which I was already accustomed to doing, as I originally had the box to the left of the screen and keyboard. The left side, now at the back, can be flush against the wall if desired.

Sincerely,
Nora M. Taylor
Bethesda, MD



Dear **INSUA:**

This is written using **FANCY FONT** and a N* Advantage and an Epson FX-80. The text-editor is Word-Star.

The address for the publisher is **SoftCraft**

8726 S. Sepulveda Blvd. Suite 1641, Los Angeles, CA 90045

I have not fully explored all the program's capabilities, but I find the type very attractive in some of my business correspondence using the dot-matrix printer.

Question for your technical expert re: Morrow DMA controller who said earlier that it can control both 5.25 and 8". Is the 5.25 drive the N* (10sector) format?

I am a new member, and am interested in balance of articles between Horizon, Advantage, CPM and DOS.

Sincerely,

Eugene Dong MD,JD
786 Holly Oak Dr
Palo Alto, CA 94309

INSUA:

Since North Star seems to be going after the business market, S-100 articles of substance in other journals seem to be drying up. I hope Compass will fill the gap with specific information on North Star operating systems, quirks, utilities, and other software goodies.

I am not interested in re-hashes of hardware press releases, excessive space devoted to the new kid (Advantage), which I don't own, or excessive allegiance to CP/M.

Yours faithfully,
Ian G. Ellis
Fairdale KY

INSUA:

I have an immediate need for some specialized software. Can you help me find a software package designed especially for churches that will run under CP/M or North Star DOS? It should include membership lists, tithe and offering bookkeeping, and things like that. Help!

Sincerely,
G.L. Hooks
Hooks Electronics
16 Berkeley Road
Framingham, MA 01701

[See Bob Applegate's PEKIN in this issue. Any other readers to the rescue? --Ed.]

INSUA:

I am trying to understand exactly what the North Star does after its POJ to E800. ... If anyone has a listing of the code at E800, knows how it works, remembers seeing an article about it, I would appreciate it if you could share that information with me. I would, of course, be willing to share with the club any information given me (as long as the supplier agrees).

Further, if there are any members who know anything about the internals of UCSD Pascal in general (or North Star UCSD Pascal in particular), I would appreciate help. I suspect that North Star did not write a full p-code interpreter, but just tacked a North Star BIOS onto a Z-80 p-code interpreter. I hope that the "BIOS hooks" required by that Z-80 p-code interpreter are documented somewhere. Better still, I hope the Z-80 p-code interpreter is in the public domain (as it might be if it was developed at the UCSD campus), and might be available if I just knew where to ask. Any assistance or information would be greatly appreciated.

Thanks for any and all help,
David P. Holberton
13229 Nassau Drive
Dale City, VA 22193

INSUA:

Almost a year ago I became a dealer for Advantage Computers in my country. I

have a very good opinion about the Advantage hardware and also about the TBS software. Nevertheless I have been maimed by name recognition. Advantage who? TBS who?

With the new trend to bundle hardware and software initiated by Osborne, followed by Kaypro and now by Morrow, the days of North Star, Advantage and so on are numbered. What a sorry situation! Unless they do THE SAME. Why? Because their hardware (even though good) is not that outstanding to create a niche for them alone in the marketplace.

Cordially yours,
Felipe S. Loaiza, Manager
SURTIELECTRONIC

[We note that IBM PC is selling an unbundled system--like hotcakes! And Osborne is in trouble. We trust that North Star will just go on churning out Horizons and Advantages--and finding buyers! --Ed.]

INSUA:

I have a standard Horizon with ... two 8" DS drives (Morrow Discus DJ2) and SMS (Software Music Synthesis System for North Star DOS by Jon Bokelman).

I want assistance to have a version of SMS for CP/M, 8" SSSD diskette and a best music board.

I want assistance for the purchase of one 256K memory board for CP/M, CP/M3, MP/M, and North Star with 8 or 16 bit CPU, SD Systems ExpandoRam, Macrotech International Corporation, or another company.

Thank you,

M. Marc Lamarre
2825 ave Livernois #10
Quebec (QUEBEC) G1L 4S8
Canada

[We suggest you first contact Jon Bokelman about a possible CP/M version of SMS; perhaps readers can assist M. Lamarre with other matters. M. Lamarre's letter was accompanied by an advertisement for a 256K board, 8088 microprocessor board and CP/M-86 board for the North Star Horizon from Greenbriar

Marketing International, Inc., 509 So. 48th St., Suite 105, Tempe AZ 85281. --Ed.]

INSUA:

The DOS Monitor TM program test for memory takes about 15 minutes for 16k. There is no indication if it is running or crashed unless there is a memory error. If you could publish the source code for the Monitor or give an assembly patch to print an asterisk after each loop it would help. In fact the source code would allow us to modify the Monitor or put it on a prom.

Also, if there is a program in public domain to dump the variables in a program to show which are being used, I'd like to see it in Compass (in N* BASIC preferably).

Yours,
M. Gilbert
W. Hempsted NY

(Readers--help! -Ed.)

INSUA:

I have a copy of DOS 1.0.0 and GBASIC and BASIC version 5.3 but need a full copy of documentation. Can any reader help?

J. Baumgartner
Box 7D RD2
Conklin NY 13748

INSUA:

I need an assembler to protect source code on N* Advantage using BASIC or GBASIC.

Need instruction on programming the PF keys on Advantage (we have the KEYMASTER CP/M disk from Computer Shoppe) under BASIC and GBASIC.

Looking for a driver under GBASIC for Okidata. Am trying to change the driver in the Advantage Demo Disk--having trouble. Have tried Measurement Masters but have received no response.

Tim Deaton
733 1st Street
Shelbyville, IN 46176

+++++

INSUA DISK LIBRARY

JRT PASCAL CANCELLED: see #1012

All disks are \$15.00 unless noted otherwise. Specify **disk number** and **Single** or **Double density** (SD or DD). Enclose check and send to INSUA, P.O. Box 2910, Fairfield, CA 94533.

NORTHSTAR FORMAT

- #1001 NS DOS and BASIC 5.2 with Ramtest and Monitors. SD \$10.00
- #1002 DD \$10.00
- #1003 "TELSTAR" MODEM PROGRAM to run under Northstar DOS.

- #1004 "THE COMPASS" Volume 1, no. 1.
- #1005 "THE COMPASS" Volume 1, no. 2.
- #1006 DISASSEMBLER by Lance Rose. SHORTHAND CODER.
- #1007 "THE COMPASS" Volume 1, nos. 3 and 4.
- #1008 MICRO-COUNT II VERSION 2.0. An accounting package. SD (two disks) \$20.00; DD (one disk) \$15.00
- #1010 "THE COMPASS" Volume 2, no. 1.

CP/M FORMAT

- #1011 CP/M Utilities Disk #1. Includes Squeeze routines and FLS-11.COM, which acts like a UNIX pipeline, and directory sort routines.
- #1012 JRT PASCAL for CP/M. **CANCELLED**. May be ordered direct from JRT Systems, 1891-23rd Avenue, San Francisco, CA 94112.
- #1013 MODEM7 for CP/M; version of program by Ward Christensen, et al. Currently DD only.
- #1014 CP/M Utilities Disk #2. SUBGEN (performs submit functions), SAP (Sort and Pack directory), SWEEP

(directory housekeeping; SWEEP14 (for CP/M 1.4), and RESOURCE (8080) / ZESOURCE (Z80) disassemblers.

NORTHSTAR FORMAT

- #1021 "THE COMPASS" Volume 2, nos. 2 and 3.
- #1022 "THE COMPASS": Special Issue; Volume 2, no. 4; and Vol. 3, no. 1.
- #1023 North Star DOS/BASIC 2.1.1
- #1024 Mover/Equus for DOS 2.1.1. Includes routines for Sol owners.

Insua

*Publishers of the Compass Newsletter
International NorthStar Users Asociation
PO Box 2910 • Fairfield, CA 94533*

*Bulk Rate
U.S. Postage
PAID
Walnut Creek
Permit No. 203*

TATE 2766
JAMES TATE
23914 SPRING DAY LN.
SPRING, TX 77373

THE COMPASS NEWSLETTER

INTERNATIONAL NORTH STAR USERS ASSOCIATION

VOLUME III, NO. 4



THERAPY FOR BURNT-OUT HACKERS

FROM THE CHAIRMAN

The West Coast Computer Faire provides an exciting mix of new technology, new ideas, and many, many, many people seeking and sharing information. As such, it is a perfect opportunity for INSUA members from all over the world to come together.

Announcement of Annual Meeting Of INSUA

As chairman of INSUA's Board of Directors, it is my pleasure to announce that our third annual membership meeting will take place at the West Coast Computer Faire on Saturday, March 24, 1984 at 1:30 pm. The meeting room provided courtesy of the Faire management, will be posted at the INSUA booth # 1139C in Brooks Hall, San Francisco Civic Auditorium.

Call for Nominations

Among other business to take place at the annual meeting, there will be an election of seven members to the board for the coming year. At this time we are soliciting nominations of candidates for directors of INSUA for 1984. Below you will find information on how to submit nominations and elect directors of your choice.

Join Us!

For those of you who are planning on attending the Faire, be sure to drop by the booth, introduce yourself, sign up for 1984 (we're going to make it worth your while right off the bat), ask questions of the person on your right, answer questions for the person on your left, pick up the current Compass newsletter, and generally join in the commotion, hubbub, and spirit of things.

If, alas, you are unable to attend the Faire, you are not going to be left out. Sign up for your membership in what promises to be the most exciting year for INSUA yet, including a new look and more frequent Compass newsletter, new dynamite software offerings from our expanding N* and CP/M software library, more user tips than ever for HORIZON, ADVANTAGE,

and DIMENSION computers, and always a special place in INSUA's heart for vintage brands (yes we know you love your SOL computer).

If, by lucky chance, you are at the Faire on Saturday afternoon, please join us in the meeting room, hear the learned speakers, cast your ballot, and say your piece. And oh yes, refreshments will be served.

Election of Directors

Deadlines with respect to the election of Directors of INSUA for 1984 are as follows.

February 22, 1984. Submission of nominations to appear on the printed ballot are closed. Ballots are printed and mailed to members. Nominations will be accepted by mail up to March 20, 1984, although the names of these individuals will not appear on the printed ballot sent to members.

March 20, 1984. Ballots received at the INSUA Mailbox by this date will be counted at the March 24 Annual Meeting. Ballots may be submitted in person at the March 24 meeting.

The powers of the corporation are vested in the Board of Directors, which may delegate the performance of duties to officers and agents of the Corporation. The term of office of a Director is one year, and the Director must be a member in good standing during the year served.

The Board of Directors meets monthly, with these meetings generally occurring in the San Francisco area. Travel to and from these meetings is the responsibility of the Director, with mileage reimbursement made at the rate of \$ 0.20 per mile to a maximum of \$20.00 per meeting. Directors unable to attend these meetings may elect to substitute phone expenses for the travel allotment for the purpose of being represented at the meetings.

Any member can submit one or more nominations and may include oneself as a nominee. However, the name of a member

should not be submitted until the proposed nominee has been contacted by the nominating member and agrees to the nomination.

Nominations submitted in writing by members not in attendance at the meeting must be received at the INSUA office by March 20, 1984. Nominations received prior to February 22, 1984 will be included in the printed ballot sent to members.

Members can be nominated by voice at the meeting on March 24, 1984. The names of such individuals will not, of course, be on the printed ballot previously submitted to the members for the purpose of absentee voting.

It is suggested that a short statement be included with each nomination as to the qualifications of the nominee for the position of Director.

Robert Beaver, Chairman

!!! SEE LEAF BEFORE BACK COVER FOR RENEWAL AND NOMINATION FORMS !!!

INSUA BOARD OF DIRECTORS 1983

Bob Beaver, President
Clyde Steiner, Vice President
Sarah Wasserman, Secretary-
Treasurer.
George Riddle, Membership
Mark Sheppard, Disk Librarian
Larry Hamelin, At Large
Alan Nelson, Editor, Compass

DISCLAIMER

Programs printed in Compass and/or distributed through the INSUA disk library are offered to INSUA members in good faith. INSUA, however, is unable to guarantee the operation of any of these programs or to guarantee support. Users are advised to test the programs thoroughly for themselves in conditions under which they are to be used. Users who employ such programs in serious business or financial applications must do so at their own risk.

Facts or opinions published about manufacturers and dealers, and all opinions expressed in articles and letters, are the responsibility of the authors, and not of INSUA or the Editor of Compass. INSUA offers the right of reply to members and non-members alike.

INSUA's Address:

P.O. Box 2901
Fairfield, CA 94533

THIS ISSUE:

Most of the present issue of Compass was printed on a Diablo 630 printer using MagicBind software. (See review). Most glitches in the printing are the result of inexperience on the part of the Editor. Since Compass is considering a change in style and/or format, the Editor solicits members' opinions on whether changes are needed, and if so, of what kind?



NORTH STAR PLANS 16-BIT IBM-COMPATIBLE SYSTEM

By Deidre Depke
Associate Editor
Micro Market World

[This article is reprinted from ISO WORLD (November 14, 1983), p. 11, with the permission of the editor. ISO WORLD has recently changed its name to Micro Market World. --Ed.]

SAN LEANDRO, Calif.--Financially troubled North Star Computer Inc. has jumped into the 16-bit market with an IBM PC-XT-compatible system that, if marketed properly, could save the company from ruin, according to the company's president and industry analysts.

The multi-user system, called Dimension, will be unveiled at the Comdex/Fall '83 show.

North Star has long been criticized for its failure to develop a 16-bit system.

"I think this route of finding a niche in the IBM world with a unique architecture is the correct strategy for long-term success for a company such as North Star," said North Star President Charles Grant.

"The product is enough to keep them afloat," said Ellen Levin, analyst with International Data Group (IDC) of Framingham, Mass. "It's a nice product at a super price. There's a real need for a good multi-user system," said Levin.

North Star will market the Dimension through OEMs and dealers. The company is targeting large OEM contracts to expand its distribution in the Fortune 1000 area and evaluating its base of dealers to select those who will be able to sell the system to small businesses, the firm said.

Grant said dealer response to the preliminary information about the computer has been very good. "Our problem is selecting those who are interested," he noted. "We're actually having to trim our dealer list."

The list price for the Dimension is \$7000 for the complete system with a 15-Mbyte hard disk, a floppy disk, two workstations, and the operating system. A 30-Mbyte version with the same configuration lists for \$8000. Each additional workstation costs \$1,500.

Analysts said marketing the system may be a problem for North Star.

"They're going to have a problem finding the distribution channels," said Ralph Gilman, an analyst with InfoCorp of Cupertino, Calif. The problem, he said, will be selecting dealers who won't be intimidated by offering a system that directly competes with IBM, and who will have sufficient knowledge to sell multi-user systems.

IDC's Levin agreed that finding the right channels will be difficult.

"They're going to have to be real choosy about channels," she said. "They'll have to find someone who knows multi-users. It will be good for specialty stores."

However, Levin does not see a problem with dealer intimidation. "Not everybody wants a multi-user personal computer. So they'll buy some IBM, some North Star. It's really kind of a complementary sales strategy. Anybody who tries to overpower IBM loses; North Star knows that."

In December 1977, when North Star shipped its first computer, the 8-bit Horizon, the company had a good lead as one of only a handful of companies in the multi-user marketplace. By the end of 1981, more than 30,000 Horizon systems were installed, and North Star had a leading position in its field. That year the company introduced another 8-bit system, the Advantage.

Since then, however, the company's sales have slipped and analysts said the company--in order to remain in business--needed a 16-bit computer to offer its OEMs, systems houses, and dealers.

The Dimension is North Star's answer.

In preparation for marketing the 16-bit system, the company hired Dharam Ahuja as senior vice president of marketing and sales to work on broadening the North Star's distribution structure. Ahuja plans a strategy for the Dimension based on its design and economics.

North Star is banking on multi-user workstations that will enhance teamwork within the company and can utilize IBM PC-XT software to attract consumers.

The firm also wants users to recognize that the Dimension can save them money. North Star literature says a five-user Dimension system will cost 50% less than five IBM PCs or PC-XTs.

The Dimension can support up to 12 workstations, each running a different PC-XT business application.

The heart of the system is a single-board computer based on the Intel 80186 processor and a 13-slot IBM bus. The CPU for the workstation board is an 8088-2.

The operating system is compatible with IBM PC-DOS 2.0 and has built-in electronic mail capability for up to 12 users. It provides each user with the equivalent of a PC-XT with networking and shared access to resources such as fixed disks, printers, and communications devices.

#

NORTH STAR NEWS RELEASE

[The following bulletin was released by North Star Computers for 23 December 1983. -Ed.]

The North Star HORIZON 8/16 new capability for a classic micro

North Star Computers, Inc., announced today that it will offer its new HORIZON 8/16 microcomputer preconfigured for two users. As a result of this offer, customers will be able to purchase the new multi-user, multi-processor computer with a five megabyte hard disk for two users for as little as \$2,600 per user. ¶We assume this means \$5,200 for the package. - Ed.]

North Star has found that typical HORIZON customers start off with a two-user system. This was the basis for the two-user preconfigurations, according to North Star's HORIZON Product Marketing Manager Karl Sterne.

"This should assist dealers because they won't have to configure the system for typical initial sales," explained Sterne.

Up to eight users can be supported by the HORIZON 8/16 system--for even less cost per user than the initial two-user bundles. For example, a 15 megabyte HORIZON 8/16 for eight users costs less than \$1,300 per user.

The original HORIZON product line, first introduced in 1978, has become a leading S-100 business microcomputer system. In 1981 it became a multi-user system for up to five users sharing one central processor. The new HORIZON 8/16 offers either 8-bit or 16-bit processor boards that are dedicated for each of up to eight users. With the NorthStar TurboDOS operating system, the result is a

microcomputer that can perform some applications as much as three times as fast as the older HORIZON.

TurboDOS is a high-speed operating system that has gained popularity in Europe during the last three years. It is a superset of CP/M, with extra commands and a friendlier interface. NorthStar TurboDOS is the first available 1.3 version, offering 16-bit CP/M-86 capability and networking features such as electronic mail. It is a true multi-user system containing the record locking features of MP/M II and MP/M-86.

North Star is also now shipping a new 5 1/4 inch, 30 megabyte hard disk with the HORIZON 8/16. The new 30 megabyte hard disk HORIZON 8/16, with a retail price of \$6,999, offers almost twice the storage capacity of the company's eight inch 18 megabyte disk subsystem. The new 30 megabyte disk drive will come integrated into the HORIZON 8/16, and can be used with North Star's present operating system and applications without modification. A 30 megabyte Hard Disk Upgrade Kit for current HORIZONS is available for a retail price of \$4,999.

A two-user HORIZON 8/16 configuration is available with a 30 megabyte hard disk for \$7,699. A 15 megabyte two-user configuration is also available for \$6,699.

#

NEWS FROM CANSUG



By Edgar F. Coudal

TECH MEETINGS: Kaltronics, North Star distributor in Northbrook, IL, opened its doors to our group on Sunday, August 7, for the technical meeting. Paul Werner, N* Midwestern regional sales rep, was on hand and presented a hands-on demo of the NorthNet LAN, then a slide show about the Big Blue Star. There was no technical meeting in September because the editor had to spend Sundays at various statistical and probability research facilities, most notably Arlington Park racetrack. Technical meeting in October centered on communications problems and software, with a step-through and demonstration of the public domain MODEM7 program.

NORTH STAR COMPANY NOTES: In keeping with its recent aggressive marketing efforts, the company is offering a new program called "Flexi-Bundle". Despite the name's awkwardness, it accurately describes the program. When a user purchases a new Horizon or Advantage, he gets to choose up to \$2,000 in North Star software to go along with the machine...FREE. The selection list looks like one of those Chinese menus where you pick one from A and two from B. On the list are operating systems, languages, word processors, spreadsheets, data base managers and all the rest of the N* proprietary stuff. It should be a useful marketing effort. The company claims that Flexi-Bundle, used to its fullest, effectively reduces the price of a system by 36%. Now, if they'd only do something about their stupid ads!

PROGRAMMING TIPS. One of the single smartest things you can do as a programmer is presume you and other users are really dumb. Portable Computer magazine says that the most common user input error is hitting a RETURN when the program expects a number or string. To avoid various problems this can cause, always add a line such as 105 in this kind of construction:

```
100 INPUT "First name: ",F$
105 IF F$ = "" THEN 100
110 !"HELLO, ",F$
```

If the user enters a RETURN (null string or "") the program kicks him back to line 100 and repeats the question. In North Star BASIC, the problem has been solved internally for numbers, because the interpreter simply says, somewhat grouchy but with a certain air of resignation: "INPUT ERROR-RETYPE"

Another good protection is in being thoughtful when designing lines to accept answers to YES/NO questions. Take this program:

```
50 !"THIS IS THE BEGINNING."
100 INPUT "RETURN TO BEGINNING? ",X$
110 IF X$ = "YES" THEN 50
120 END
```

If the user types in Y or YEP or YOU BETCHA, BOSS, the program ENDS. To avoid that, use this line:

```
110 IF X$(1,1)="Y" THEN 50
```

That accepts Y and YEP and anything else that begins with the letter Y.

If you can anticipate that the common response to the question will be YES, then here's an even better solution:

```
110 IF X$(1,1)<>"N" then 50
```

Now, the user has to make the very conscious effort of typing NO or N or NOT ON YOUR LIFE, BUSTER in order to end the program. Anything that does not begin with an "N" will return him to the beginning of the program, such as Y or YES or CAROL LYNLEY or CUBS or even 7 or especially a RETURN. To complete this discussion, the fastest and best way doesn't

even use the INPUT statement. Instead of the final version we came up with immediately above, use this program:

```
50 !"THIS IS THE BEGINNING."  
100 !"RETURN TO THE BEGINNING?"  
110 X$=INCHR$(0)  
120 IF X$ <> "N" THEN 50
```

The INCHR\$(0) statement waits for only one character from the keyboard before roaring on to the next line.

MORE ON converting programs written in other BASICs to N*:

A subtle problem may occur if you exit a FOR/NEXT loop improperly before the loop is completed, as in this program:

```
100 FOR I = 1 TO 10  
110 IF I = 3 THEN 200  
120 NEXT I  
200 REM MORE PROGRAM CODE....
```

North Star BASIC will execute that without complaint, but it may not clean up the pointer that is still keeping track of the I loop, which would be stuck at 3. In some cases, when another FOR/NEXT loop is encountered in the program, you may get a NEXT WITHOUT FOR error, which will drive you bats if you properly constructed the second loop. To avoid this rare problem, rewrite line 110 to read: IF I = 3 THEN EXIT 200. The EXIT tells the interpreter to clean up all the old loop overhead, such as pointers.

Still another FOR/NEXT problem with conversions is solved with equal ease. In TRS BASIC, you can end nested loops with a single NEXT, and in MBASIC, you can end them with one NEXT and the names of the variables, i.e.:

TRS-80

```
10 FOR I = 1 TO 10  
20 FOR J = 1 TO 5  
30 NEXT
```

MBASIC

```
10 FOR I = 1 TO 10  
20 FOR J = 1 TO 5  
30 NEXT J, I
```

North Star BASIC will crash on both constructions, but will respond to either of these:

```
10 FOR I = 1 TO 10  
20 FOR J = 1 TO 5  
30 NEXT  
40 NEXT
```

```
10 FOR I = 1 TO 10  
20 FOR J = 1 TO 5  
30 NEXT J  
40 NEXT I
```

(The above was originally printed in ACCESS magazine.)

HARD DISK HINTS: With more of our number adding hard disks, we'd like to pass along some hints and tips.

1. If your hard disk is mounted in a separate cabinet, so that you have to turn it on separately from the computer itself, be wary of the order in which you flip switches. Think of the order as a window, with the Hard Disk inside. Turn ON the terminal, computer and other stuff first, then turn on the Hard Disk. Then allow at least 10 seconds for a 5-meg hard disk to come up to speed before loading the operating system. When turning OFF the machines, turn off the Hard Disk first and allow at least 10 seconds for the capacitance to drain. Then turn off the other things. The reason for this is that the computer itself may spray wild electrons around at startup and shutdown, and if the Hard Disk is running, you may contaminate data, or worse yet, the directory. I know North Star denies this. I also know that I've had to reformat my 5-meg hard disk twice because of this...

2. For those with older 18-meg hard disks, N* suggests cleaning the timing wheel under the sealed assembly once a month.

3. For those who are running HDOS 2.0 or earlier, and who want to move up to the new HDOS 2.1.0, pay very close attention to the caveats in the manual. There is an especially tricky part that asks the question, "Is This What you want?" If you type in YES, which is the logical response, you will wipe out the disk completely. If you type in NO, the program goes ahead and executes perfectly. Switching from earlier HDOS to HDOS 2.1.0 is one case

where you **MUST** read and understand the doc before starting...

4. Since HDOS does not offer a RAMTEST, simply use the old ones that came with 5.2. They load at 3000 and 5000, out of the way of all the HDOS stuff that runs from 0 to 2600. However, you can't reboot with control-C since it doesn't know where to find the HDOS. Neither do I. The origin tables in the HDOS manual are absolutely incomprehensible.

5. The HDOS directory of utilities that come with the system lists a tantalizing program called CPM.FIX and indicates that it is 140 sectors long and is a CP/M file. But it is impossible to access and worse, is not mentioned anywhere in the documentation. Viewing it through DED and DDT sheds no light on what it might be or do. If anyone knows, I'll pass the information along in the next issue.

WANT TO BE AN AUTHOR? Remember the dearly beloved Whole Earth Catalog of the late '60s and early '70s? The publisher, Stewart Brand, is now putting together the Whole Earth Software Catalog. Same idea as WEC.

Doubleday will print the first edition of WESC in fall, 1984, and Stewart is looking for reviews of any software for any computer. He wants two to six paragraphs, and will pay \$20 if the review is printed, \$10 if you're the first to suggest it and someone else reviews it. Will also publish articles, with payment of from \$100 to \$300 each. Send the reviews to Richard Dalton, Editor, Whole Earth Software Catalog, PO Box 428, Sausalito, CA 94966.

InfoWorld magazine reviewed the Advantage in its Sept. 1 issue and gave it extremely high marks. Reviewer Doug Topham summed it all up by saying: "The Advantage is an outstanding buy. It is compact, easy to use, powerful and highly reliable. I can honestly say that I could not find a single defect--or even minor annoyance--in the unit I tested. Everything works just the way you would expect it to work...For a single-user CP/M system with high-resolution graphics, the Advantage is just about unbeatable. It's an excellent choice for anyone who wants to use a computer for word processing, data-base applications, accounting or business graphics."

#

HEBREW AND GREEK

By George Eldredge
2501 Carquinez Ave
El Cerrito CA 94530
(415) 233-0069

I have programs that transliterate Hebrew and Greek from BASIC or WordStar on my Horizon to my Epson 84 dot-matrix printer. There is unlikely to be enough interest to list the programs here, but I will assist anyone who is interested. The system is straightforward. If the following description does not give you enough ideas, I can help you out.

A program CODE.COM must be run before going into Wordstar. It has booster code in 100H and boosts the decoder language and codes from 200H to high memory above my 54K CP/M.

A program DECODE.HEX has its origin in 0D920H. When called there from a BASIC program or from a patch in my CP/M BIOS, it transliterates the ASCII code in register DE using the code in DAOH + (DE * 8). Register

D is 0 for Hebrew and 40H for Greek. This is the program boosted from 200H by CODE.COM.

The Patch in my BIOS looks at Wordstar text for \H or \G followed by a stream of characters ending in a /. Non-alphabetic characters, such as punctuation, are not decoded. This patch could be put in Wordstar if it is not wanted to decode output from other sources.

The character codes and descriptions are in a Northstar BASIC program. When run it prints large-scale letters for inspection and stores the codes in the high memory.

The Greek has upper and lower case. Upper case English goes to final Hebrew letters for letters that have final forms.

This letter will end with the characters \HMolw/, which will be transliterated as Sholom.

□17W

A REQUESTED MONITOR *fix*

By

Saul G. Levy
2555 E. Irvington Rd., Lot 47
Tucson, Arizona 85714
(602) 889-7753

M. Gilbert in the Compass (Vol.III, No. 3, p. 58) wanted a disassembled source code listing of the North Star Monitor and a modification to the Test Memory (TM) routine which will show that it is working (similar to that in RAMTEST3 and RAMTEST5). I have disassembled this program and here is the patch he wanted. If for any reason you have not used the Monitor before, the patch will give you a good reason to start using this very useful program.

The RAMTEST3 and RAMTEST5 programs that come with releases 5.2 and 2.1.1 do not always find memory errors. The Monitor memory test will sometimes find these errors.

Testing memory with a delay of 0 seconds runs fast, but it will not show most of the memory errors which exist. You need to test with a delay of 1 or 2 seconds to catch all of them. The memory block is first written with a running cycle of hex bytes. Then after the delay period, a read cycle occurs which compares what was just read to what should still be there from the write cycle.

Using a long delay on a large block of memory will make the computer act as if it has gone off in the weeds until the routine prints "PASS COMPLETE" to show it is still there and working. The modification below will print an asterisk (*) after each of the 256 subpasses which end at the "PASS COMPLETE" message. Every byte in the memory block will be set to all 256 hex values during each full pass (from 0 to FF).

The patch instructions are for the M0E00 Monitor that came with Release 5.2DQ. This table covers all of the releases that I have access to. Start your changes at the following addresses (or use the same offset with another version from each release):

Release No.	Origin (ORG)	Offset from ORG	Patch addr.	Starting addr. of new code
4.0	2A00H	0404H 023BH	2E04H	2C3BH
5.1	2D00H	043DH 025FH	313DH	2F5FH
5.2	0E00H	043DH 025FH	123DH	105FH
2.1.1	1000H	043DH 025FH	143DH	125FH

Releases 5.1 and 5.2 apparently use the same source code. 2.1.1 is very similar, but 4.0 is very different. This patch should work with all of these releases. My modifications were tested on only the 5.2 0E00H version. Use great care when determining the relevant addresses to change. Hex addresses are easily confused.

You can add the patch to a running version of the Monitor, but I suggest you load the file at the normal address (when in the DOS: LF M0E00 E00), then use another Monitor version (at a different address) to make the changes. This saves much confusion about addresses, and your Search Memory (SM) commands will also appear at the end in the command buffer to cause a second match! If you only have one Monitor, just load the copy to be changed with an offset of 1000H (LF M0E00 1E00, etc.). Check your work very carefully if you use the offset!

The first change is to replace the instruction at the patch address. This is where the patch will be "added" so that we will jump to the patch's actual starting address. The original instruction at the patch address will be moved to the end of the new instructions.

The new instructions will fit at the end of the command table so we will put them there. This saves having to reassemble the Monitor! Type one of the following lines to find the instruction we have to change (it happens to be a jump already!):

```
DH 1230,10          Display Hex bytes around the C3 (jump)
                    in 123CH
or
```

```
SM E00-15FF 1,1,0,C3  The C3 jump is at the same address as
                        above (make sure you look at only the
                        Monitor you wish to change!). DH as
                        above at the addresses shown
```

Only one C3 jump should appear within 16 bytes of the one we want. The assembly instructions are:

Address	Bytes	Instruction	Notes
1239	01 01 00	LXI B,0000H	The only one in the program
123C	C3 47 12	JMP 1247H	Write this jump address down!
123F	60	MOV H,B	

This code is in all four releases. Type the following three lines to change the jump address to jump to the new code (DS is Display and Substitute):

```
DS 123D
47= type 5F then space bar for next one (printed on this line)
12= type 10 then RETURN to end
```

Check your changes by typing: DH 1230,10 again! You should see: C3 5F 10. If not, it is far safer to reload and start over!

The command table ends with 17 free bytes in all versions. The last valid command must be followed by a zero (00) byte! Add this patch only AFTER the first of these zeroes! If you find less than 17 bytes, you have a modified Monitor which includes extra commands and will have to find a copy that has not been changed. Type one of the following lines to find the DS command at the end of the command table (note that this is not the command buffer mentioned above):

```
DA 1050,20          Display ASCII values at end of command
                    table
or
SM 0E00-15FF "D","S"  The DS command ends the command table.
                    DA as above at the addresses shown
```

After DS you will see the two-byte address (with the low-order byte first) where the DS routine is located in the Monitor, then comes 17 zero bytes (105EH to 106EH). Leave the first one a zero and change the next 14 bytes as you did above:

```
DS 105F
00=C5 00=F5 00=3E 00=0 00=6 00=2A 00=CD 00=3
00=E 00=F1 00=C1 00=C3 00=47 00=12 then RETURN to end
```

The last two bytes you changed are the jump address you wrote down above. You should have two more zeros left at the end of the command table! Check this by typing: DA 1050,20 and also check that you have entered these bytes in the correct order!

This ends the modifications. The new assembly instructions are:

Address	Bytes	Instruction	Notes
105F	C5	PUSH B	Save register B
1060	F5	PUSH PSW	Save accumulator and flags
1061	3E 00	MVI A,0	Output device no. in accumulator
(terminal)			
1063	06 2A	MVI B,42	ASCII asterisk in B, 42 is in decimal (change to suit your taste)
1065	CD 03 0E	CALL 0E03H	Character-output-routine jump within in Monitor which jumps to DOS' COUT
1068	F1	POP PSW	Restore accumulator
1069	C1	POP B	Restore register B
106A	C3 47 12	JMP 1 47H	Add jump we "moved" above

Return to DOS and save the newly patched Monitor in a different file (until it is checked out) by typing:

```
OS          Return to DOS
CR M0E00* 8   CReate a new file (change name to suit)
TY M0E00* 1 E00  TType is a 1 and give the correct GO address
SF M0E00* E00   Save the file
```

Finally, test the new Monitor by typing:

```
GO M0E00*    Load and GO the new Monitor
TM 0 0      Test address 0 with 0 delay
```

Hit Control-C when you want to stop the test. You should also check each of the other commands to make sure they still work! Be careful when changing the input and output device numbers (ID and OD) since they can make your computer act as if it just died.

Making the modifications above does not increase the size of the program. It will still run within the 8 blocks of memory it loads into (the stack and pointer storage are carefully written!). The copyright notice is partly destroyed by the stack. One other interesting thing I found out is that the delay time is a hex number which can be very large (up to FF). Now you can try a longer delay and see it work! A delay of 2 seconds is the longest you will ever need to find those bad problems that dynamic memory seems to have at times.

Finally, if anyone wants a copy of the source code or a listing of same: please send me a diskette and \$2.00 for postage and handling (or \$7.00 total for one of my diskettes). Pack your diskette carefully! The listing is 16 pages with the symbol table and will cost \$4.00 for copying, postage, and handling. Note that the program has not been commented and does not contain the patch! I am not an assembly programmer and have no idea what most of the Monitor's code is doing (fooled you, didn't I?). The source code is 64 blocks long and can be assembled by the ASMB assembler in Allen Ashley's PDS package (or by others, it can be edited by the SECRETARY word processor). Please call or write if you need further details. The disassembler (ZDIS by Morris Miller) and my label program (SLABEL) are also available (free) at the same rates (combine orders and save \$1.00 from the total postage).

P.S. Mr. Gilbert also wanted a BASIC program to dump the variables used in a North Star BASIC program. BASIC is a very poor (slow) language to use for this purpose. As before (see Vol.II, No.4, p.33), I have to recommend the best cross-reference program that I know of (it is in the public domain). XREF was written by Larry Hudson (see Dr. Dobb's, No.49, October, 1980, p.32). I have (finally) figured out how to make it work with double-density files (it reads the file directly off of the diskette). The source code takes up 90 blocks and the listing 15 pages. The running program stores in only 6-8 blocks. Several versions are available at the same rates as above (note that I am not selling this program). Please tell me which printer port you are using (1 or 2), the length of your terminal and printer line (80 columns, etc.), and where you want the XREF program to load. Putting XREF below a DOS at 2000H or in high memory will prevent the running of large programs. Many output options are built-in to this program: call or write for details.

MAGIC-BIND: A REVIEW

Reviewed by
Alan H. Nelson

A question often asked in History of Printing classes: What was Gutenberg's most important invention? Answer: Not printing per se, but the adjustable type-mold. This adjustable mold permitted the casting of types of **variable width**, resulting in the nearly automatic production of books which were in effect facsimiles of manuscripts. The written word, whether manuscript or printed, incorporated information not merely cryptically encoded, but highly legible and easily decoded. The variable width of individual letters came to be an essential ingredient of a legible text.

Although the typewriter contributed a great deal to the ease and accuracy of verbal communication, it was a step backward in the sense that the letters of the alphabet all had to fit into a matrix of uniform height and width: i's and l's had to be given disproportionately large feet to be made to seem the same width as m's and w's. Typewritten text has never been other than ugly in appearance. Typewritten text is more legible than most handwritten text not because it is more esthetically pleasing, but because it is more uniform than most handwriting. True type-set text is uniform, easily legible (because proportionally spaced) and esthetically pleasing.

Most computer-driven printers are in essence typewriters. Dot-matrix printers are for the most part inferior typewriters, though appearances are being improved with fair rapidity. Perfection of dot-matrix printing is being achieved only with the application of laser techniques, which can be controlled to a fineness of thousands of dots per inch. Daisy-wheel, or "impact" printers offer superior legibility of individual letters, but for the most part produce text which looks like it has been printed on a high-class standard typewriter.

A practical approach to the achievement of the "type-set" look is to use a program which will drive an impact printer and also give characters their proper width. This is not a simple task, since the number of characters which will appear on any given line will differ with the accumulated widths of all the characters which happen to occur in that line. Thus a line with numerous w's and m's may

take fewer words, whereas a line with numerous i's and l's may take more words.

WordStar has tried to offer proportional spacing, but so far has only achieved an undocumented and unsupported version invoked by placing a **TP** at the beginning of the file. Apparently an occasional line will give problems because of too many wide letters. Also, variable spacing on printout is not generally compatible with WordStar's principle that "what you see on the screen is what you get on printout."

MagicBind

MagicBind, an updated version of a product formerly called MagicPrint, is an advanced and generally successful text formatting system for enhanced printing via microcomputer. In fact, the text you are reading now was formatted and printed with the assistance of MagicBind. This is one case in which the medium really is the message: if you like the appearance of this text, you will like MagicBind; if not, not.

The essential purpose of MagicBind is to support proportional-spaced printing on designated impact printers like the Diablo 1640, 1650, and 630 series, and the NEC. It runs on both CP/M and M/DOS. MagicBind boasts an ability to support most aspects of proportional printing, plus a host of additional functions normally gathered together in a program like the Mail-Merge adjunct of WordStar. Polished results require the use of a print-wheel with proportionally spaced letters and a carbon ribbon. (Fortunately, regular print-wheels and cloth ribbons may be used for all trial printing leading to the final printout.)

Sophisticated proportional spacing of characters results in text which cannot usually be displayed on the screen the way it will come out on paper. Since the number of characters per line is not a constant, the text is formatted on the fly. MagicBind has a function which allows the text to be presented on the screen, but the text looks little like the end result: essentially, it is possible only to check page breaks and certain print modes like underlining. Characters which are struck twice in printout

are presented side by side on the screen: thus underlining looks like t_h_i_s_, whereas boldfacing looks like **tthhiiss**.

The resulting text must eventually be printed out to be understood or appreciated esthetically; but what an improvement over regular typewriter-like printout! Legibility is vastly improved and saving in space is in the neighborhood of 10 to 20%.

MagicBind comes in two versions, one which can be used with most wordprocessing programs, and another designed to make life easier for WordStar users. An example of the difference is that underlining is invoked like this in the vanilla flavored version and like ^Sthis^S in the WordStar version.

Commands

Although a series of paragraphs can be printed out straight from a raw text file, all special features must be invoked by using embedded commands. Local commands affecting a single word or cluster of words are invoked as in WordStar, with toggle-type controls. Thus **†Bthis word†B** would be boldfaced on printout in the WordStar version. Formatting commands affecting character-spacing, paragraphing, justification, page breaks, and so forth, are invoked with dot commands. Cleverly, MagicBind uses a double-dot command like **..C** (for centering) and a single dot command for comments. Thus commands for MagicBind are comments for WordStar and vice versa, so a single text can be formatted to run on either program (naturally with differing results as to pagination, line breaks, and so forth).

A particularly useful feature of MagicBind is the **..B** command, which will cause the printer to roll the paper back toward the top of the page. This feature is useful for multi-column formatting. Of particular value is the fact that text in columns can be formatted and reformatted with relative ease. Compare this to the otherwise extremely useful column-control command (**^AKN**) in WordStar, which if invoked to produce two-column text will destroy the **^AB** reformat capability in both columns.

Programs

MagicBind comes with two major programs plus a selection of demonstration texts. The main program is **MERGE.COM**, which processes the text and interprets the imbedded print

control commands. It gets its name from the fact that it also incorporates sophisticated merge capability (like MailMerge in WordStar) to generate documents which incorporate data from a data file (like individually addressed letters), boiler-plate text (for legal forms, etc.), and chain-printed documents (like successive chapters of a book). MagicBind also handles sequential numbering of sections of a document, and (very important!) footnoting.

LABEL.COM is a secondary program which handles the features of MagicBind normally required to generate single-column or multi-column labels. A label-format file is established which can call a given data file; the result is beautifully printed labels with the option of proportional spacing (uniform spacing can also be invoked). Empty lines will not be printed, thus avoiding the ugly gap which sometimes occurs between the name and the address on the label. As for size and spacing of labels, two default formats are offered to the user; selecting the third option will allow the user to design his own label. (See "Problems" below.)

One minor program included with the package is **VERIFY.COM**, which can be used to check a database in advance for items which do not conform to the standard number and size of field. This program is useful for picking up cases in which, for example, a comma is missing or occurs where it should not, or cases in which a field is not of a standard size.

Problems

MagicBind, even with its revisions from MagicPrint, is not perfect in all respects. Most exasperating is its inability to handle multi-column formatting on the fly. Your editor has wasted hours and hours formatting the two-column text you see before you. It should be possible simply to give a "two-column" formatting command, with the necessary parameters for column width, space between columns, and so forth. Instead, the text first must be printed out in narrow columns; then the text is broken at the desired location (only finding a viable location without encountering error messages is not an easy task!); then the text must be backed up the required distance (not always easy to calculate!); a **^AR** must be inserted in the bottom line of the first column to achieve proper right justification; and the text must be printed out to determine the correct results. If I sound exasperated, I am. This is all simply a matter of calculations, and calculations is what I bought my computer for!

The screen-formatting option on printout is useful, but will not confirm the register of columnar printing: thus it is necessary to print out a page on the printer to learn whether the second column begins on the same line as the first column.

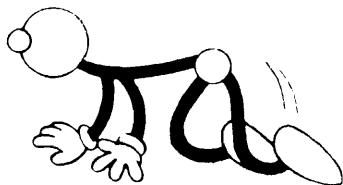
A problem for which MagicBind is not responsible, but which it might have solved, is the skewed line at the top of the second column. This is caused by slack in the printer tractor-feed system. It would have been useful if the printer could have been backed up a few extra lines so that the paper is always moving forward through the roller at print time.

Another problem which has your reviewer highly exasperated is the consequence of bi-directional printing for the LABEL.COM program. Zip-sorting normally requires labels printed in the following order:

```
1 2 3 4
5 6 7 8
9 10 11 12
. . . .
```

LABEL.COM prints labels as follows:

```
1 2 3 4
8 7 6 5
9 10 11 12
. . . .
```



This may make the program useless for commercial mail services. It also surprised me that "Cheshire" formatting, the industry standard, is apparently not supported as one of the standard options.

Documentation

Documentation is friendly, fairly thorough, and generally competent. Examples are usually informative and adequate. I would have preferred a separate section on printing envelopes. It took me some time to discover that LABEL.COM cannot handle individual envelopes because it cannot be made to pause after each label. MERGE.COM handles the task nicely.

P. 14-1 of my manual contains an error: a [or ^A (WordStar) is not used before the filename at the head of a label format file. I understand this has been corrected in later versions.

Conclusion

MagicBind is an excellent program which does virtually all it promises, and contains a few nice bonuses. MagicBind will allow individual microcomputer users and office staffs to produce documents which approach type-set texts in appearance and legibility. Users who produce massive documentation may find that the compression of texts results in a recovery of the cost of the program through savings of paper and perhaps postage.

MagicBind is currently available for use with a limited number of impact printers, including the NEC and many daisywheels (not the Diablo 1620). For many computer users, the cost of purchasing both MagicBind and an appropriate printer will be prohibitive. Those fortunate enough to own an appropriate printer will find the cost of the program minimal in comparison to its benefits. Those considering the purchase of a printer should certainly give an extra long look at printers which can be supported by MagicBind.

MagicBind is such a good program that it should be worth the efforts of the programmer(s) to make it better. In particular, it should be provided with a column-printing option which will generate columns of text automatically; the LABEL.COM program should handle mono-directional printing sequences; the screen-print option should be able to verify the results of "reverse leading" (the back-up procedure). I understand the company is working on true multi-column capability: I hope they succeed before my patience runs out!

Your editor is interested in your reactions to the new Compass appearance, which is the result of MagicBind. Please send your comments!

Availability

MagicBind is available from Computer EdiType Systems, 509 Cathedral Parkway, New York, NY 10025. Its cost is \$250.00, including software and manual. MagicPrint, which lacks filemerging and chapter numbering capability, is still available at the lower price of \$195.00. Unless you are an absolute pauper, I strongly recommend you spend the extra money for MagicBind.

MASTER CATALOG

A review by

Saul G. Levy
2555 E. Irvington Rd., Lot 47
Tucson, AZ 85714
(602) 889-7753

I have been looking for a disk catalog package to use with CP/M diskettes as a complement to the package I have been using with North Star DOS diskettes. I have seen the ad from Elliam Associates for their Master Catalog System and finally bought a copy of what is now the second version. They can be reached at 24000 Bessemer Street, Woodland Hills, CA 91367, (213) 348-4278. Their catalog lists a lot of interesting software for CP/M users.

A double-density, two-sided (flippy) diskette arrived in 13 days. It only cost \$15.00 (plus \$1.50 for shipping and handling) because the programs came from the CP/M Users Group's diskettes. Most of the programs included were written by Ward Christensen and will work on both single- and double-density. And what a collection of files are included! There are 47 files taking up 194K of disk space! This is so confusing at first that Elliam has included a READ.ME file to tell you how to get started, which is exactly what it does.

The Files (Sorry, nothing is in BASIC!)

The files include 15 .COM files (26K), 14 .DOC files (26K), 15 .ASM files (130K), one .LIB file (11K), one file to label the diskette (0K), and one file to hold the catalog (1K). This is quite a collection. How often do you get the .ASM files when you buy a program? Not very often!

The Problems

If you have purchased user group's diskettes before, you know the programs can often be trivial or buggy, or will work only on certain hardware. CP/M is supposed to prevent this last problem, which partly it does, but still there can be problems, along with other ones which I haven't even gotten to yet!

First, these diskettes contain an unedited hodgepodge of programs which is left to the user to figure out. This may intimidate a beginner, especially when you read all of the warnings included in the documentation files. I have had other CP/M programs destroy my files and even the directory which ended the usage of that diskette. I used every .COM program in this package without any real problems. Only one of these won't run on a standard, double-density, 58K North Star Horizon-1 (64K total) running Lifeboat's CP/M Version 2.2. If you get strange messages, re-boot and try something else.

Second, the documentation is substantial, but doesn't seem to include as much information as you really need. You have to explore this package to find out what is there and what it will do. This was fun, but may leave a beginner out in the cold.

Third, I noted that at least one of the .ASM files doesn't exactly reproduce its .COM file! This type of sloppy work is another problem you will find in users group's diskettes. This may not be the fault of the programmer since diskette librarians usually just collect what they receive and put all of it onto one diskette for distribution. I can't really blame the librarians either. How would you like to be responsible for programs written by someone else, especially in a language you don't know?

Fourth, some of the .COM files don't have .ASM files and vice versa (you can't publish what you don't receive). Two of the .ASM files can't be assembled by the normal CP/M assembler or the makro assembler! Not less than two, very fancy assemblers were used with these files. One file was easily edited to assemble with ASM.COM, but the other is a mess that even MAC.COM can't handle! These files handle user numbers and are obviously useful, but being able to assemble only one of them is not useful at all. All three of these related files shouldn't have

been put in this package. Also, two of the .COM files don't have .DOC files.

Fifth, I have found that user group's diskettes rarely tell you which computer system the programs were written for. This package is totally obtuse on this point. Two versions of one program (SAP) are included because the original version wouldn't run on some unspecified system! The documentation also includes a second modification which does tell you which system it was written for, but the documentation writer tried running it on a different double-density system and it didn't quite work! I tried the first, modified version (CATS) and it wouldn't run on my system, but the original version ran fine and it was written for eight-inch, single-density (IBM) only! Also, you are told that the first, modified version is for single-density only. Maybe so, but only the two, modified versions wouldn't run on my system. This is a point you must explore to find out what pertains to your system. I found that the .ASM files all used the same sector skew table as a North Star. Skewed sectors are sectors which are not written in sequential order, but skip some number of physical sectors after each written sector. The skew is set to six, but the actual order is a bit more complicated than this.

No Fault Programs

At the price, I can't blame Elliam for any of the problems above. I will send them my corrections and hope that they will upgrade this package to make it more useful. I had to ask my friends for help with a few of the things that needed fixing.

Catalog Usage

Side one of the original diskette contains a OK file named -MSTRCAT.099 which names this diskette and also numbers it. The minus sign before this dummy file name tells the Master Catalog System which file name will be used to label all of the files from that diskette so you can find the programs later on (all 12 characters are used). The number can run from .000 to .999 so you can catalog up to 1000 diskettes. To create this dummy file name, type: 'SAVE 0 -XXXXXXXX.YYY'. The X's are intended to describe what is on each diskette (MaSTeR CATalog here), the Y's to allow printing out a sorted list by diskette number, instead of the entire catalog in sorted order.

To add a diskette's directory to the master catalog file, called MAST.CAT, is a two-step process. There are no less than four programs which can be used for the first step! The fanciest program is FMAP which is a very slick program. Type: 'FMAP A: F' and wait a few seconds. FMAP creates a temporary file called NAMES.SUB which holds the ASCII-sorted file names from drive A: now labeled with the entire diskette name (the latter is also included, labeled with its own name). Only these names appear in the files.

There are three programs which can be used for the second step! UCAT is also a slick program. Type: 'UCAT' and wait what can be a much longer time as the MAST.CAT file grows larger. UCAT merges the NAMES.SUB file with the old MAST.CAT file by creating a NEW.CAT file which contains all of the labeled file names in ASCII-sorted order. Then it renames the old MAST.CAT as MAST.BAK, the NEW.CAT file as MAST.CAT, and erases the NAMES.SUB file. If an old MAST.BAK file exists, it is erased at the proper time. All of these files are straight ASCII text and can be listed with the TYPE command ('TYPE MAST.CAT', etc.) or edited.

If you give FMAP a one-character command (there are 10 of them), you must include the drive letter (A:). I found that you don't have to name each diskette in advance! UCAT won't work and will ask you to rerun it with the diskette name following ('UCAT -XXXXXXXX.YYY'). The original MAST.CAT file was supposed to contain a list of file names which UCAT would ignore and not write to the new MAST.CAT file. My copy only contained the file name DUMMY.FIL which I edited to what it was supposed to be. This feature will ignore all of the copies of STAT, PIP, etc. FMAP has an option which will ask you whether you want each named file written to NAMES.SUB. UCAT is also smart enough to update a changed diskette properly. Just enter the same diskette again with FMAP and UCAT, and UCAT will add or delete any new or erased file and not create extra copies of files which haven't been changed. Other FMAP options include an extra help mode ('FMAP H'), a more detailed STAT-type display including the sector allocations for each file ('FMAP'), and various modes to add SUBMIT-type, dummy commands (\$1, \$2) to each file name to use while down-loading files with a modem program. I could write a

whole book on the features of FMAP and its three, related programs, but don't have the space.

The Other Programs

Four of the other .COM programs will print the catalog data in various ways. Try them all to see which ones you prefer. Three others will print the current diskette's directory in various ways often including the file sizes in K bytes. This is nice, but not that important to the catalog system.

Two jackpot programs are FIND and SAP, either one of which is worth the entire cost of this package! FIND will search any text file for any ASCII string of characters you enter. This is very handy for other files not connected with the catalog system. SAP sorts the directory into ASCII order which normally you see only when using STAT. Finally, I can get my messy directories sorted into a reasonable order! SAP will erase all 0K files so you will have to create minus diskette, name files of 1K instead of 0K (CATS doesn't need this, but it won't run on my system). This is wasteful of space, but that is CP/M's fault.

Conclusion

In general I recommend this package to anyone who cares to explore what it really can do. Test it thoroughly before you go to the trouble of loading the directories off of all of your diskettes! Beginners can follow the READ.ME file to get going. Elliam recommends you use CATU.COM rather than UCAT.COM which I described above. They give the impression that both of these programs are the same. Not so! CATU.COM will delete the minus diskette name from the master catalog, UCAT.COM keeps it. I think it is handy to have these extra files at the beginning of the catalog so you can quickly check which diskettes have been run. You can just suit yourself on this point.

#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#-#

While exploring this package I decided to write a completely new catalog system for North Star DOS diskettes. The old system I have been using stores too much stuff which I don't really need. Watch for it in a future issue of the Compass!

#

```

100 REM      THIS PROGRAM ACCEPTS ANY TWO NUMBERS, THEN ADDS THEM TOGETHER,
110 REM      DISPLAYS THE RESULT, AND OFFERS THE OPTION OF DOING IT AGAIN
120 REM      COPYRIGHT BY E.F.COUDAL, OCTOBER, 1983
130 DIM A(1),B(1),C(1)
140 GOSUB 200 \REM INTRO PROGRAM
150 GOSUB 300 \REM INPUT NUMBERS
160 GOSUB 400 \REM ADD NUMBERS
170 GOSUB 500 \REM OUTPUT RESULT
180 GOSUB 600 \REM DO IT AGAIN?
200 !" THIS PROGRAM ADDS TWO NUMBERS TOGETHER"
210 RETURN
300 INPUT "THE NUMBERS TO BE ADDED?",A,B
310 RETURN
400 C=A+B
410 RETURN
500 !"THE ANSWER IS",C
510 RETURN
600 !" DO IT AGAIN?"
610 INPUT H$
620 IF H$ = "Y" THEN 150
630 END
640 RETURN

```




Figure for "Structured BASIC," p. 18.

STRUCTURED BASIC? WHY?

By Edgar F. Coudal

One of the major complaints heard about programs written in BASIC comes from elitists and CS snobs who decry its haphazard, unstructured appearance.

"Hard to read," one sniffs.

"Littered with GOTOs, a sure sign of programming expediency," says another while stuffing punch cards into his tweedy jacket pockets.

"BASIC programmers only write flow charts after the program is finished," still another protests. (Hell, I thought that's the way it was supposed to be done. Doesn't everyone begin planning a program by sitting down and typing AUTO 100,10?)

North Star owners who use North Star BASIC or the ultimate BASIC, APCBASIC from American Planning Corporation (reviewed in these pages by Bob Stek last issue) are guilty as charged. North Star owners using one of those two BASICs are probably even more guilty, in fact, than their brethren using crippled, little deformed interpreters such as MBASIC, simply because the North Star DOS BASICs do so much more.

If your use of BASIC embarrasses you (some have the feeling that BASIC should only be done behind locked doors by consenting adults and the hands should be washed immediately after) then I have a suggestion. Tell everyone you write in "Structured B." When they ask what that is, thinking they may have missed some vital issue of Byte or IEEE Proceedings or People, simply respond, "Structured B is a predecessor to C." (That's literally true, by the way, but if your meaning was misinterpreted, it's not your fault.) They won't pursue it further because hardly anyone in the world wants to pursue discussion of the predecessor to C, a language that lets you write things like ++ as a legitimate operator and doesn't even blink if you write a routine that will wipe out all your previous work and render the guy in the next office sterile.

Not to make a liar of you, let's actually learn how to write Structured BASIC by comparing a normal BASIC program and one that does exactly the same thing in structured fashion. Let's look at a program that takes two numbers input from the keyboard, adds them

together, and prints the result, then lets you do it again without RUNning the program a second time.

Any idiot, cretin, kindergartner or Dijkstra can write a short, fast useful, accurate North Star BASIC program to do this, thusly:

```
100 INPUT A,B
110 PRINT A+B
120 GOTO 100
```

Further, they can have it done, the answers accepted, and be off opening a Budweiser somewhere before the Pascal programmers faced with the same problem have figured out which semicolons are missing. (The APL programmers are in even worse shape: they have to remember how to use pixel addressing to make a symbol that vaguely resembles a Japanese character meaning "house of hungry dogs".)

But that little program above violates all the classical structured language precepts: variables were not declared, subroutines or procedures were not specified, types were not defined, remarks are not there to help next year's entry-level klutz figure out what it's supposed to do, and so on, and on. The final horror is that dreaded GOTO.

So let's rewrite it in Structured BASIC. This will make everyone happy. (See p. 17.) Now, let's look at it line by line:

100 and 110 Structured programmers demand that you tell everyone right way what the program is supposed to do. One of their favorite bon mots is "You're not writing an Adventure game, you know." Nudge. Nudge.

120 You also are supposed to say who did it and when. Every CS snob I know automatically copyrights everything. There must be more than 700,000 copyrighted versions of Kernighan and Ritchie's main printf "hello" program around. When forced to do this, I sometimes put in little notes like "Write if you get work," or "bring home a can of shaving cream." But I digress.

130 Declare your variables. DIMming arrays likes this is as good a way as any to declare

numeric variables in BASIC. To play the structured game right, if you forget to declare a variable in this line, you must SCRatch your whole BASIC program and start over. That wastes about as much time as compiling a Pascal program, then watching it crash on the first run because you forgot to declare a variable, then correcting the source code and recompiling.

140 thru 180 Structured guys like to zoom around to subroutines and defined functions and stuff they wrote last year and saved in a file called "LIBRARY." Some of those libraries are 180K and contain things like routines to center the Towers of Hanoi graphics display in your choice of 16,64 or 80 characters. It takes them longer to find the routine than to rewrite it. Tracing one of their programs is like diagramming a Ronald Reagan sentence. And remember the REMs. Got to tell everyone what we're about to do here.

200 A screen display! Tells the user what we're going to do!

210 And back to Master Control.

300 Moving right along here. Give us the numbers.

310 Back to Master Control.

400 Fill a variable with the result of the addition. I don't know why it's done this way. They rarely use the expression to express the result; instead, the structured guys use a variable to hold the value of the expression. I guess it's because you've got all these registers and memory and so on and if you don't use as many as possible, the little contacts will oxidize from neglect and cause trouble in February, 1997.

410 Back to MC so we can continue to ripple thru the GOSUBs.

500 It worked! It prints the answer! It tells us that it is the answer!

510 To MC.

600 Now you get a choice. You have to type a key and RETURN to quit; you have to type a key and RETURN to do it again. In the normal BASIC program, you'd PANIC-C out. Messy, but really effective.

610-620 de rigneur routine. I honestly have never been able to tell the philosophical difference between this and a GOTO. Seems that you're using more stuff to accomplish the same thing.

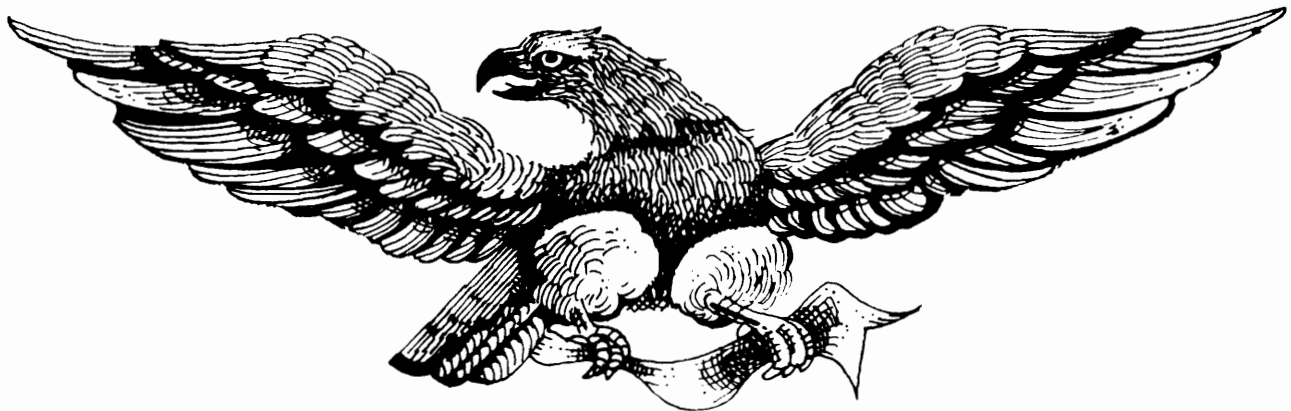
630 No more? OK. Quit. Finis. END.

640 Look, pal, for every GOSUB, there must be a RETURN. Never mind that it is never reached and that it doesn't serve any function. Just type it in and spell it right. Otherwise the program will be **messy** and **expedient** and **incomplete**.

Now you know how to program in Structured BASIC. You're welcome. Me, I'd write it: 100INPUTA,B !A+B GOTO100. If they can't figure it out, they shouldn't be looking in there anyway.

(The above was triggered by a letter in Portable 100 (October) and it's only fair to cite that as another example of structured BASIC explained.)

#####



IN THIS CORNER . . .

Commentary by Saul Levy

Levy to Coudal (see figs. 1-2, p. 21)

I was happy to see Ed Coudal's discussion of North Star BASIC (Vol. III, no. 3, p. 4). I agree that this BASIC is one of the finest and easiest to work with. BASIC allows rapid program development without the overhead of FORTRAN or Pascal which can make you give up before you even begin to code a program! Writing a program does create more work for you, especially if your tasks are small ones, but once you complete a program it can be run forever...(well, almost)!

Beginning programmers often need help in how to go about solving their problems. This help should be offered in two phases and is not just HOW to code a program (what Ed covered), but EXACTLY what it is that you are coding (the algorithm). You can't be expected to write a sort routine unless you have a very good idea of how you sort things. You have done this before by hand so think about how you did it. Only then should you try to write code that will do the same thing (this should be the second phase).

I wish that Ed had spent some time describing EXACTLY what he was coding. This is the most important phase and is usually not described! Once you understand the algorithm, then the actual coding will seem relatively straightforward. As an example, the DIMension statement in line 110 should have been defined before Ed started the actual coding. This will show that you know what you are doing!

Right there is where computers show us just how dumb we are. Computers can do only, and exactly, what we tell them to do. They can't learn anything, but provide us with a quite remarkable tool to teach ourselves how to provide the computer with detailed enough instructions to give us the correct answer. This is not a chicken or egg first situation! The programmer must use the computer to make sure his programs are accurate (testing becomes an art). In addition, any nontrivial programs should also solve people's problems. To do less is to waste a powerful tool. You know of many cases where the computer spit out the wrong answer (even if you aren't a programmer)!

Almost all of these are due to human error (the hardware is amazingly reliable, programmers are not).

The only way to learn how is to work at it, and it takes a lot of work to learn well (just like anything else worth doing). Contrary to the magazines I have been reading, not everyone can or should be a programmer. Ed Coudal doesn't want anyone to show him a better way (p. 9). I suggest that this is hurting his grasp of BASIC programming. All of us have a lot to learn!

When the term "elegant" refers to a program it means that this program is concisely written (no verbiage or poorly thought out algorithms), runs as fast as possible, and uses a minimum amount of memory. Note that an elegant program will also be efficient. Ed believes that this is unimportant, and believe it or not, he is right! The most important thing about a program is that it runs as described and gives the correct results. Programming is a trade-off much of the time and worrying about writing an elegant program is a waste of time for a beginner. Later on, when you learn about some of the standard tricks that programmers use, you can start using them too. You would not write last year's programs the same way this year. In five years your programs should show a much improved use of algorithms and be more elegant. Keep learning!

Ed used a partially correct, but overly simplified definition of recursion. The examples of routines that I have seen all call themselves. For example:

```
100 GOSUB 1000
110 code here
1000 code here
1010 GOSUB 1010  Calls itself as
many
times as
necessary
1020 code here
1030 RETURN
```

Ed's definition would imply that all loops are recursive, but this is not what is usually meant by programmers. I do admit to partially

```

10 REM PROGRAM COINS
20 REM FROM THE COMPASS, VOL.III, NO.3, P.4, INTERNATIONAL NORTH STAR USERS
30 REM ASSOCIATION
40 REM ENTERED AND MODIFIED BY SAUL G. LEVY, TUCSON, ARIZONA, NOVEMBER 17, 1983
50 REM

100 REM PROGRAM DETERMINES MINIMUM NUMBER OF COINS TO MAKE A TOTAL
105 REM BY EDGAR F. COUDAL PARK RIDGE, IL 4.9.83
110 DIM A(4)
120 DEF FNA(A1,A)=INT(A1/A)
130 INPUT "AMOUNT? ",A1
140 !
150 IF A1<.01 THEN 130
160 A1=A1*100
170 DATA 50,25,10,5
180 FOR B=1 TO 4
190 READ A
210 A(B)=FNA(A1,A)
220 T=T+A(B)
230 A1=A1-A(B)*A
240 NEXT B
250 !"COINS NEEDED:",T+A1
260 !
270 !"HALF-DOLLARS:",A(1)
280 !" QUARTERS:",A(2)
290 !" DIMES:",A(3)
300 !" NICKELS:",A(4)
310 !" PENNIES:",A1
320 RESTORE
330 T=0
340 !
350 GOTO 130
360 REM LAST LINE

100 REM PROGRAM DETERMINES MINIMUM NUMBER OF COINS TO MAKE A TOTAL
105 REM BY EDGAR F. COUDAL PARK RIDGE, IL 4.9.83
110 DIM A(4)
120 DEF FNA(A1,A)=INT(A1/A)
130 INPUT "AMOUNT? ",A1
140 !
150 IF A1<.01 THEN 130
160 A1=A1*100
170 DATA 50,0,25,0,10,0,5,99
180 FOR B=1 TO 4
190 READ A,C
200 IF A1<A THEN IF C=99 THEN 250 ELSE NEXT B
210 A(B)=FNA(A1,A)
220 T=T+A(B)
230 A1=A1-FNA(A1,A)*A
240 NEXT B
250 !"COINS NEEDED:",T+A1
260 !
270 !"HALF-DOLLARS:",A(1)
280 !" QUARTERS:",A(2)
290 !" DIMES:",A(3)
300 !" NICKELS:",A(4)
310 !" PENNIES:",A1

```



splitting hairs here. North Star BASIC "dislikes" recursion anyway. Each GOSUB call uses three bytes of the stack and you will eventually run out of memory. Even with a 58K system there is not enough memory to do anything especially useful (please send me examples that work).

The coin-change algorithm that Ed used is actually better than he gave it credit for. The loop from line 180 to line 240 will run four times (once for each coin over a penny). This means that you can remove the 0's and 99 in the DATA statement in line 170, and the ",C" in the "READ A,C" in line 190.

Line 200 is the major problem with this program. It is very "wordy" and complicated (is this necessary?). I also have to ask, what is returned by the user-defined function if A1 is less than A? The function will return a zero which is exactly what we want for the coin counter "A(B)" whenever none of that particular coin are in the calculated change. Therefore, line 200 is not needed at all! Remove it and the program still runs fine.

Finally, why is the user-defined function called twice? Ed stores the RETURNed value in "A(B)" the first time so why not just use "A(B)" in line 230 for the "FNA(A1,A)"? These three changes make his program more elegant and save 50 bytes of memory too (almost 10% of the program).

There is more than one harmful programming practice in line 200 (which can be deleted anyway). These are styles of coding which work fine in other BASICs, but not in North Star's. Ed mentions using an EXIT before the "250". This should be standard practice for North Star BASIC users as it will prevent any "CONTROL STACK" errors. Ed got away with it in his simple program, but I'll bet you won't! Learn the correct form now and it will save you much grief later on. Even more harmful is the "NEXT B" at the end of line 200. This style of coding is often found in Microsoft BASIC programs which can use it. When you convert programs from Microsoft to North Star BASIC you will be driven crazy with "CONTROL STACK" errors until all of the extra NEXTs are removed (you can have a dreadful time finding the last NEXT). What you should use here is simply "ELSE 240". This takes a little longer to execute, but can't cause this error.

I have included a listing and file with these changes and have added the code needed to continue the program without another RUN.

The A-array doesn't need to be reset to 0, only the total number of coins (T). I would have zeroed T at the beginning of the program.

One other thing needs correcting in Ed's "North Star Notes" on p .32. OD stands for Output Device, not Other Device! OD is meant to give you an exact output of whatever you type along with the responses from the computer. It does make a terrible typewriter!

Levy to Garst

Perry Garst's article (p .10) gives a useful though cumbersome way to save file space. If you need to do this, you can use his method which will take much longer to read or write the file (byte by byte), plus extra time to encode the numbers into bytes and vice versa (time may not be important for small files). His program segment has a number of serious errors which deserve comment. It appears that Perry doesn't understand the user-defined function (join the crowd, the function is complicated and poorly explained, I have tried my hand at it too with less success than I like).

Perry used what appears to be a multiple RETURN. I was surprised to see that several of the RETURNs in his functions used ANDs. For example:

```
RETURN A1 AND A2 AND A3 AND A4
```

```
or RETURN C1 AND C2
```

This format points out that the RETURN can indeed use a complicated statement instead of just a variable name, but it looks like four RETURNS for the price of one. Sorry, you can execute ONLY ONE RETURN, and must return ONLY ONE VALUE. Then, what value does he RETURN? A 0 for false or a 1 for true, of course!

The North Star AND is a logical function (not a bit-wise AND, as in Microsoft BASIC). The value returned can only be a 0 or a 1, which is much smaller than what is usually needed. We want A1, A2, A3, and A4 each to be set to some value from 0 to 255 so they can be saved in the file (in this program the RETURNed value is not even used and can be ignored, see below).

The code in the first user function (on the third line) also has a problem. A1 is usually set to a negative number which can get quite large due to the *100 factor. Was this intended to be

a round-off function? A negative or large (greater than 255) value for A1 is illegal in the WRITE statement, and the other variables will all be written as 0's or 1's.

I believe that lines 2-6 should have been written as:

```
DEF FNA(A)
A=INT(A*100)      Convert to pennies
                  and remove any
                  fractional cents
A1=INT(A 65536)  Highest-order byte
A5=A-A1*65536   Remainder
A2=INT(A5 256)  Middle-order byte
A3=A5-A2*256    Remainder (lowest-order
                byte)

RETURN A
FNEND
```

Note that this really is a three-byte function as Perry says (only A1, A2, and A3 need to be written to the file). We get the same accuracy and need to write only three bytes by simply converting our amount to pennies before turning it into bytes (Perry didn't do it that way, but I did save an extra byte). My code can handle amounts up to 160+ thousands because you still can handle only 8-digits total. To handle amounts up to 16+ millions (with a 10-digit or more BASIC), use the following and DO tack on the pennies in A4:

```
DEF FNA(A)
A=INT(100*A) 100 Round off to nearest
              penny (two decimal
              places).
              Note the difference
              from above
A1=INT(A 65536) Highest-order byte
A5=A-A1*65536  Remainder
A2=INT(A5 256) Middle-order byte
A6=A5-A2*256  Remainder
A3=INT(A6)     Lowest-order byte
              (remove remainder)
A4=A6-A3       Save remainder (in
              fractional dollars)
A4=100*A4      Convert this to pennies
              to be tacked on

RETURN A
FNEND
```

Note that A4 will also be in the range from 0 to 255 (you can have from 0 to 99 pennies). Also note the simple RETURN statement. The

values calculated for A1, A2, A3, and possibly A4 remain after the function ends and will be written to the file. Only the variables named in the parameter list (the "A" within the parentheses in the function DEFINITION and calls) will keep their prefunction-call values. Whatever happens to these variables WITHIN the function is lost when the function ends, UNLESS a RETURN of one of these variables occurs. Using "RETURN A" will force the "A=" in the function call to the new value calculated within the function itself.

To retrieve the three or four-byte number created above, use the corresponding function below (the colons represent backslashes which drive my word processor crazy):

```
DEF FNB(A)      Three-byte function
A1=A1*65536 :
A2=A2*256 :
A=(A1+A2+A3) 100 :
RETURN A
[the above is one line]
FNEND
```

or

```
DEF FNB(A)      Four-byte function
A1=A1*65536 :
A2=A2*256 :
A4=A4 100 :
A=A1+A2+A3+A4 :
[the above is one line]
RETURN A
FNEND
```

The rest of Perry's code is correct, but you can change the "RETURN C1 AND C2" in the third user-defined function to just "RETURN C". Note that using my simple RETURN in his code would NOT change the values of A or C when the function RETURNS (my code will change the amount to pennies or remove anything less than a penny). Also note that the value of A or C is not used in the READ or WRITE statements and we don't really care what values they assume. This means that these functions RETURN a dummy value that is not used again (we really want A1 through A4, but still have to RETURN one value). This is a useful way to use these functions.

#

AND IN THIS CORNER . . .

By Mark Perkel
Perkel Software Systems
1452 North Clay
Springfield, MO 65802
(417) 862-9830

In response to Saul Levy's article entitled "Levy on Forth" in Compass III, no. 3, pp. 16-19, I was rather surprised to see such a negatively biased article printed in the Compass about something that wasn't supposed to be controversial in the first place. I believe in a free press, but I feel that it was rather tasteless of the editorial staff of the Compass as well as of Mr. Levy to publish such an article without contacting me ahead of time.

From Mr. Levy's article, it is apparent that he was not having a good time at Kitt Peak. If you read through the frustrations, you'll see a very strong testimonial for Forth. Obviously, Mr. Levy didn't agree but his opinion does not reflect the views of his employer as his article would have you believe. Kitt Peak dumped FORTRAN for Forth, and not the other way around.

Mr. Levy admits in his article that he doesn't understand Forth nor has he looked at any of the books available. This is obviously true judging from the accuracy of the information he presents.

What bothers me most is that editors are supposed to edit and such an article as Mr. Levy's would never have gotten published in a professional periodical just on the basis of his admission that he doesn't know what he's talking about. I would like to see the Compass keep to informative and interesting material as it should. I would much rather have had this space used for a tutorial on Forth and examples of Forth programming.

On the other side of the coin, Mr. Levy did bring up an interesting point. Forth is an elitist language. It is not as easy to learn as BASIC nor was it ever intended to be. It is not the language for the masses. It is a language for programmers who want total control of the computer. It lets a programmer do exactly what he wants. If I don't like the way Forth adds, I just change it. I'm not limited to eight open files. I'm not even limited to files.

I hope the readers of Compass will give Forth a fair shake. It's one of the few

languages available that run under DOS, and I feel that the typical North Star owner tends to be more of an elite programmer who would appreciate the extra power of the language.

Presently I'm writing a new compiler based on Forth--I'm calling it Slick. Slick will be available by the time this is published. Slick is a kind of universal compiler that will compile almost any language including BASIC. Essentially, if you describe a way you want the source to appear, you can write a front-end in Slick to compile it.

Thank you for this space to reply, good luck, and may the Forth be with you!

#

[Editor's comment: We apologize to Mr. Perkel for any discomfort he may have felt on reading Mr. Levy's commentary. Compass will strive to maintain a tradition of informative articles, avoiding any suspicion of personal attack. We interpreted Mr. Levy's article as a spirited challenge to the Forth language, not as an attack on Mr. Perkel. It must be conceded that few things are more controversial than preferences for one programming language over another (or all others).]

See for Yourself

ADVERTISEMENT OF SERVICES:

Repair of Horizons, including Quad Drives, Shugart Drives; routine drive alignments for Horizons and Advantages, etc., at reasonable prices and fast turnaround. As the Disk Turns (415) 276-5245.

TWO-COLUMN UPDATE

By Joe Maguire

Here is an update for the "Two Column LIST Routine" which appeared with the article "Assembly Language Programming" (Compass Vol. 2 No. 4, p. 9).

The update provides the necessary patches for North Star's new DOS 2.1.1 DQ and also fixes a bug which was in the original program for DOS 5.2 DQ.

The bug prevented correct Listing when the AUTOS byte was enabled. That is, if auto start to bring up BASIC was selected, the DOS would soon bomb. The bug was traced to the warm-start routine of the DOS. If AUTOS is enabled, warm-start is not executed and the patch was never moved out of the buffer.

The update places the patch mover in the initialization routine which is always executed. Another improvement is in PAGES. In the version for DOS 5.2, LI and CAT worked as desired but a Basic LIST would overrun the screen. The update places the number of files to be displayed directly into the DOS LIST routine (DOS+6A9H).

DOS 2.1.1 is essentially a complete rewrite (much better programming, by the way, than previous versions) and contains more routines and more code. The buffer area is now more crowded than in version 5.2. As a result, more patching is required. The signon message must be shortened slightly in order for the patch to fit.

The accompanying listing gives the necessary patches and addresses. Follow the same procedure as given in the article mentioned above.

```
;*****  
; Two column List routine for North Star DOS 2.1.1 DQ  
;*****  
;  
;      from INSUA Compass Vol. 1, #4, p. 25  
;      modified for DOS 2.1.1 DQ, January 84, by:  
;  
;      Joe Maguire  
;      2321 Foxhall Dr.  
;      Anchorage, AK 99504  
;  
; This routine will enable the LIST command to  
; display the directory in two columns instead of  
; the present single column. For example: Up to  
; 48 file names can be displayed on a 24 line  
; video terminal.  
;  
0100 = DOS: EQU 100H ;For DOS at 0100H as supplied  
;      on factory master disk. You  
;      can change this as required.  
;  
;      Patch addresses within DOS 2.1.1 DQ  
;  
066B DONE: ORG DOS+56BH ;Exit routine of LIST  
066B C3ED00 JMP RESTOR ;Go clean up  
;  
06EC GOADD: ORG DOS+5ECH ;Print GO address  
06EC E500 DW OUT5 ;5 spaces if no GO  
;  
06F9 NULIN: ORG DOS+5F9H ;Old start of new line
```

```

06F9 CDC900      CALL DOUBL ;Start 2 col routine
;
07A9             LINES: ORG  DOS+6A9H ;2x # of lines on CRT
07A9 3E30        MVI  A,24*2 ;Corrects LI & CAT but
07AB 00          NOP   ;keeps Basic LIST same
;
08BB            PATCH: ORG  DOS+7BBH ;Patch loader to move
08BB CD0008      CALL  MOVE ;new code
;
08C6            SGNON: ORG  DOS+7C6H ;Change pointer to the
08C6 2B08        DW    MSG ;shortened signon msg.
;
0113 =          INIT EQU  DOS+13H ;Terminal initialization
075E =          SPACE: EQU  DOS+65EH ;Output a space
077E =          CRLF:  EQU  DOS+67EH ;Output a C/R & L/F
;
; Note: The factory master disk as supplied for
; the Horizon computer has the I/O block complete-
; ly filled. Therefore, this routine is stored in
; the BUFFER area of the DOS (which will be over-
; written) and must be moved to an area outside
; the DOS. In this example, the code is moved to
; just below the DOS origin at 100H. Users who
; are not using the HRZ version can locate the
; routine at the end of the I/O area. In that
; case, delete the PATCH, SGNON and MOVE routines
; and change the ORG after the label CODE1 to your
; free memory area. DOS 2.1.1 has the sector
; loader code (for sectors 8-10) and the signon
; message in the middle of the BUFFER area. The
; MOVE routine and the patch code must be split
; into two parts in order to fit around it. In
; addition, part of the signon message must be
; shortened in order to make additional room.
; The ORGs below place the MOVE routine just
; before the sector loader and signon message and
; the two column list code just after it.
;
0800            ORG  DOS+700H
;
0800 CD1301      MOVE: CALL  INIT ;Do TINIT routine
0803 21C800      LXI  H,FLAG ;Point to RAM dest.
0806 111C08      LXI  D,CODE1 ;Point to code
0809 0E0E        MVI  C,BYTE1 ;Byte count 1st part
080B CD1308      CALL  M1 ;Move 1st part
080E 0E27        MVI  C,BYTE2 ;Byte count 2nd part
0810 11D808      LXI  D,CODE2 ;Move 2nd part
0813 1A          M1: LDAX  D ;Get a byte
0814 77          MOV   M,A ;Move it!
0815 23          INX  H ;Bump pointers
0816 13          INX  D
0817 0D          DCR  C ;Decrement counter
0818 C21308      JNZ  M1 ;Loop until done
081B C9          RET   ;Return to loader
;
; Note: The following code, stored in the BUFFER
; area of the DOS, is relocated by the MOVE routine
; above during bootup. Enter the 1st group of bytes

```

```

; in the BUFFER area beginning at the address shown
; to the left of the label CODE1. Next, enter the
; shortened signon message at the address shown to
; the left of MSG. Enter the 2nd byte group beginning
; at the address of CODE2. The last byte, C9, should
; fit just two bytes before the start of the I/O
; block. Don't forget that the DOS is stored in com-
; pacted form on the disk. See the N* software
; manual for additional details.
;
081C = CODE1: EQU $ ;Address marker
;
00C8 ORG DOS-38H
;
00C8 00 FLAG: DB 0 ;Been here before flag
00C9 3AC800 DOUBL: LDA FLAG ;Get the flag
00CC B7 ORA A ;See if zero or non
00CD C2DE00 JNZ D2 ;If NZ do CRLF else spaces
00D0 2F CMA ;Flip flag
00D1 32C800 STA FLAG ;Store it, we've been here
;
; Note: If you have 64 col video, use 7 spaces
;
00D4 0E08 MVI C,8 ;Set up space counter
;
; What follows is the shortened signon message.
; Only the words "Disk Operating System" have been
; shortened to "DOS". The rest is unchanged.
;
082B ORG DOS+72BH ;It starts here
;
082B 0D0A MSG: DB 0DH,0AH
082D 4E6F727468 DB 'North'
0832 2053746172 DB ' Star'
0837 20444F53 DB ' DOS'
;
00D6 ORG DOS-2AH ;Reset origin
;
08D8 = CODE2: EQU DOS+7D8H ;Where part 2 starts
;
00D6 CD5E07 D1: CALL SPACE ;Output spaces
00D9 0D DCR C ;Decrement counter
00DA C2D600 JNZ D1 ;Loop until finished
00DD C9 RET ;Return for next entry
;
00DE AF D2: XRA A ;Zero the flag
00DF 32C800 STA FLAG ;Store it
00E2 C37E07 JMP CRLF ;New line & next entry
;
00E5 0E05 OUT5: MVI C,5 ;5 spaces this time
00E7 CDD600 CALL D1 ;Put em out
00EA C3F906 JMP NULIN ;Next entry in col 2
;
00ED 3AC800 RESTOR: LDA FLAG ;Get the flag
00F0 B7 ORA A ;See if zero or not
00F1 CAF700 JZ NOCR ;Zero means cursor is
; at left margin
00F4 CD7E07 CALL CRLF ;Give C/R if not

```

```

00F7 AF      NOCR: XRA  A ;Zero the flag
00F8 32C800  STA  FLAG ;Store it
00FB 3C      INR  A ;Make A non zero
00FC C9      RET  ;Return to DOS
;
00FD =      ;FIN: EQU  $ ;Address marker
;
000E =      ;BYTE1: EQU  D1-FLAG ;Length of 1st part
0027 =      ;BYTE2: EQU  FIN-D1 ;Length of 2nd part
;
08FE =      ;MARK: EQU  CODE2+BYTE2-1 ;Last addr
;          ;in BUFFER
00FD      END

```

* * * * *

A memory dump of the patches required in DOS 2.1.1 DQ for enabling the two column Llist routine. The addresses shown are those applicable after a copy of the DOS has been loaded at 2100H. The patches are for the standard DOS which loads at address 100H:

```
>DH 266B-266C
266B C3 ED 00
```

FOR SALE:

```
>DH 26EC-26ED
26EC E5 00
```

Double Density Controller Boards

```
>DH 26F9-26FB
26F9 CD C9 00
```

I recently salvaged several North Star DD controllers from a bankruptcy sale. These are a few years old but unused. All manuals and software included. (DOS Ver. 2.1.1) Price: \$290 ea

```
>DH 27A9-27AB
27A9 3E 30 00
```

```
>DH 28BB-28BD
28BB CD 00 08
```

Joe Maguire
2321 Foxhall Dr.
Anchorage, AK 99504
(907) 333-0897

```
>DH 28C6-28C7
28C6 2B 08
```

```
>DH 2800-2829
```

```

2800 CD 13 01 21 C8 00 11 1C 08 0E 0E CD 13 08 0E 27
2810 11 D8 08 1A 77 23 13 0D C2 13 08 C9 00 3A C8 00
2820 B7 C2 DE 00 2F 32 C8 00 0E 08

```

```
>DA 282B-283A
```

```

282B 0D 0A 4E 6F 72 74 68 20 53 74 61 72 20 44 4F 53
      N o r t h S t a r D O S

```

```
>DH 28D8-28FE
```

```

28D8 CD 5E 07 0D C2 D6 00 C9 AF 32 C8 00 C3 7E 07 0E
28E8 05 CD D6 00 C3 F9 06 3A C8 00 B7 CA F7 00 CD 7E
28F8 07 AF 32 C8 00 3C C9

```

#####

NORTH STAR ANSWERS

by Bob Cowart

[Bob Cowart, formerly an employee of North Star, has now taken up a career as a free-lance writer. -Ed.]

Q: We would appreciate any help you can give us regarding the use of the HSIO-4 board in the Horizon under CP/M. We are using North Star CP/M 2.2 and are trying to install the HSIO-4 I/O routines into the user area of CP/M. Being unfamiliar with assembly language we constantly develop error messages during assembly of our code and are unable to cope with them. Our goal is merely to be able to use the ports for printer output. My communications with North Star have been fruitless as they seem unable or noninterested in aiding this registered dealer. Maybe we'll be able to use the nice looking HSIO-4 board for things other than a paper weight. -P.H. Michigan

A: That's a tough one. Modifying your CP/M BIOS (basic input-output system) gets a little tricky. However, your CP/M manual explains a little about how it is done. Maybe these hints will help.

1. Make a copy of the USER.ASM file which was supplied on your factory CP/M disk. Name it NEWUSER.ASM.
2. Using a text editor in NON DOCUMENT MODE, change the I/O routines as you need to. Make sure to include the initialization routines needed for the HSIO-4. There is a section of USER.ASM clearly set aside for this purpose. The routines the HSIO-4 needs are in the HSIO-4 manual. Refer to the Asynchronous section for input and output examples.
3. Make sure to set the switches on the HSIO to something other than 0000, otherwise there will be a conflict with the Motherboard ports. Use 0010, which will set the base port at 20(hex) if you want to use the assembly language code I've included below without modifications.
4. Assemble the new code using ASM.COM. If you get errors, refer to a good CP/M manual like the Osborne one to determine what the errors were, and correct them. Then reassemble.
If you get lots of errors, check your semi-colons. Try running the file through PIP with the "Z" function to strip off the high bits which some word processors can add if used in the document mode. Example: PIP NEWUSER.ASM = NEWUSER.ASM[Z]. This sometimes eliminates lots of errors. Always use the "N"(non document) mode in WordStar to edit ASM files, and you shouldn't need to do this.
5. Once assembled, you use DDT.COM and SYSGEN.COM to "patch" your new BIOS into the CP/M operating system. First you run SYSGEN and load in a standard CP/M from disk. Then do a Control-C to get out of SYSGEN. This leaves a copy of CP/M (other than the CP/M which is already running your Horizon) in memory. This is the one you want to play around with and later save onto a disk. The problem here is figuring out where to overlay your newly assembled BIOS before you do save the whole mess on disk.

The North Star CP/M manual explains how to use DDT to figure out where to load the new BIOS into RAM. Please refer to that. Once done, you CONTROL-C again to get out of DDT. Run SYSGEN again, and this time hit RETURN when asked which drive to read the system FROM. Then write your new system out TO a disk in drive 1 or 2. Next, cross your fingers and try cold booting with the new system disk. If it boots, and the printer works, make lots of copies of your new system by running SYSGEN repetitively.

Now for some code. Below is something close to (maybe exactly) what your NEWUSER.ASM should look like. This is a modification of my own version. I use the HSIO-4 for printer and modem, but I have not actually tried this version. However, it did assemble correctly. I hope it will be of help. No guarantees though.

```

*****
; PRTP18 JMP COUTP18 ;PARALLEL PRINTER HUNG ON HD-18
; *** CRMCEN ALTERS BIOS+10H TO JUMP TO "PRTSER" OR "PRTPAR"
; DEPENDEN UPON PRINTER INTERFACE ANSWER.
; ***** END OF DEFINED JUMP VECTOR *****
;
; ***** STANDARD HORIZON USART I/O ROUTINES *****
;
; INPUT FROM STANDARD (LEFT) SERIAL PORT
CINL IN 3 ;GET STATUS FOR LEFT SERIAL PORT
NOP ;ROOM FOR PATCHING
ANI 2 ;RCVR READY ?
JZ CINL ;NO, KEEP TESTING
IN 2 ;GET THE ACTUAL CHARACTER
NOP ;ROOM FOR PATCHING
ANI 7FH ;STRIP PARITY BIT (D7)
RET ;RETURN WITH CHAR IN A REGISTER
;
; INPUT FROM SECONDARY (RIGHT) SERIAL PORT
CINR IN 5 ;GET STATUS FOR RIGHT SERIAL PORT
NOP ;ROOM FOR PATCHING
ANI 2 ;RCVR READY ?
JZ CINR ;NO, LOOP TILL RECEIVED
IN 4 ;GET CHAR FROM DATA PORT
NOP ;ROOM FOR PATCHING
ANI 7FH
RET
;
; INPUT FROM PARALLEL PORT
CINP IN 6 ;MOTHERBOARD STATUS
ANI 2 ;PI FLAG
JZ CINP ;LOOP UNTIL DATA
IN 0 ;READ DATA FROM PARALLEL PORT
PUSH PSW ;SAVE BYTE
MVI A,30H ;RESET VALUE
OUT 6 ;RESET PI FLAG
POP PSW ;RESTORE INPUT BYTE
NOP ;ROOM FOR PATCHING
ANI 7FH ;CLEAR PARITY BIT
RET
;
; CONSOLE INPUT PENDING TEST
CONVST IN 3 ;GET LEFT USART STATUS
NOP ;ROOM FOR PATCHING
ANI 2 ;HAS BYTE BEEN RECEIVED?
RZ ;NO, RETURN WITH ZERO IN A
MVI A,0FFH ;YES, RETURN WITH FFH IN A
RET
;
; OUTPUT TO CONSOLE OR STANDARD (LEFT) SERIAL PORT
COUTL IN 3 ;GET STATUS
NOP ;ROOM FOR PATCHING
ANI 1 ;IS TX READY?
JZ COUTL ;NO, LOOP UNTIL IT IS
MOV A,C ;GET BYTE TO SEND
NOP ;ROOM FOR PATCHING

```

```

*****
* FILE NAME IS NEWUSER.ASM
* modified CP/M BIOS for
* Horizon/H5IO-4
*
*****
;
; After SYSGEN or CRMCEN, load this file with ddt using syntax:
;
; Inewuser.hex <RETURN>
; R<offset>. In the case of a 58K hard disk system it is 5000h
;
; The 5000 has to be changed if you change the memsize. See the
; North Star CP/M manual for calculations using DDT's "H"
; command.-b.c.
;
; ALSO: Don't forget to change the "org" statement at the
; beginning (just below) to reflect the amount of memory
; you have, and whether you have a hard or floppy system..
;
; Any line with a semi-colon will not assemble and is used
; for comments only. Better to use a semi-colon than to
; wipe out a line so you can refer to it later on. Be sure
; to use the "H" after all hexadecimal numbers. Any Hex
; numbers beginning with a letter have to be preceded by
; a "0". Example F800H should be 0F800H.
; *****
;
; org (58*400h)-1300h ;58 K system w/hard disk
; org (58*400h)-600h ;formula for 58K floppy system
; org (64*400h)-600H ;formula for 64K floppy system
; ORG 07A00H ;ORIGIN FOR 32K SYSTEM
;
; ***** USER AREA ***** USER AREA *****
; SEQUENTIAL BASIC INPUT OUTPUT SYSTEM
; PERFORMS CHARACTER INPUT/OUTPUT FOR THE
; CP/M SEQUENTIAL DEVICES - CONSOLE & LIST
;
; THESE ROUTINES ARE FOR STANDARD HORIZON SYSTEMS
; MAY BE REPLACED IN EITHER THE SYSGEN-RAW-IMAGE AT
; 3200H-33FFH, OR IN SECTOR 8 (IN THE SYSTEM TRACK AREA),
; OR IN ITS ACTUAL EXECUTABLE LOCATION (7A00H-7BFFH IN 32K)
;
; ***** BEGINNING OF JUMP VECTOR *****
;
; USERBASE EQU $ ;MARK ORIGIN OF USER AREA
; JMP INIT ;COLD BOOT INIT (USART & PARITY SETUP)
; JMP CONVST ;CONSOLE TEST FOR INPUT
; JMP CINL ;CONSOLE INPUT (LEFT PORT)
; JMP COUTL ;CONSOLE OUTPUT(LEFT PORT)
; PRTSER JMP COUTR ;SERIAL PRINTER(RIGHT SERIAL PORT)
; JMP COUTL ;PUNCH TO CONSOLE
; JMP CINL ;PAPER TAPE READER FROM CONSOLE
; JMP PRTP18 ;TEST PRINTER STATUS - NULL IMPLEMENTATION
; PRTPAR JMP COUTP ;PARALLEL PRINTER (IF CRMCEN TOLD PARALLEL)

```

```

INIT:
; ROUTINE FOR INITIALIZING THE HSIO-4 BOARD FOR PRINTER AND MODEM
; ON BOB COWART'S IMSAI-North Star COMPUTER. 11-11-82
;
; BASE EQU 20H ;BASE PORT OF HSIO
PORTA EQU BASE+00H ;ADDRESS OF PORT A
PORTB EQU BASE+04H ;B ADDR
PORTC EQU BASE+08H ;C ADDR
PORTD EQU BASE+0CH ;D ADDR
;
BAUDA EQU 04H ;1200 baud for diablo on port a
BAUDB EQU 06H ;300 BAUD for modem on port b
BAUDC EQU 06H ;300 BAUD
BAUDD EQU 06H ;300 BAUD
;
BAUD EQU 0 ;RELATIVE LOCATION OF BAUD PORT
MAK EQU 1 ;INTERRUPT PORT LOCALE
UDAT EQU 2 ;USART DATA PORT
USTA EQU 3 ;USART STATUS PORT
;
INITH MVI A,BAUDA ;send baud rate bytes to baud register
OUT PORTA+BAUD
MVI A,BAUDB
OUT PORTB+BAUD
MVI A,BAUDC
OUT PORTC+BAUD
MVI A,BAUDD
OUT PORTD+BAUD
;
MVI A,80H ;USART COMMANDS FOR RESET
CALL USTAT ;sent to all ports
CALL USTAT ;needs TWO times to clear
MVI A,40h ;reset command
CALL USTAT ;send it
MVI A,0CEH ;give mode byte for....
CALL USTAT ;2 STOP BITS,16 * CLOCK,NO PARITY
MVI A,37H ;GIVE COMMAND...
CALL USTAT ;CMD:RTS,ER,RF,DIR,TXEN
CALL INJNK ;clear out any garbage in data port
ret
;
INJNK IN PORTA+UDAT ;READ ALL DATA PORTS TO CLEAR JUNK
IN PORTB+UDAT
IN PORTC+UDAT
IN PORTD+UDAT
RET
;
USTAT OUT PORTA+USTA ;OUTPUT COMMAND TO
OUT PORTB+USTA ;ALL 4 CHANNELS
OUT PORTC+USTA
OUT PORTD+USTA
RET
;*****
;***** End HSIO-4 Init. *****
;***** Horizon Initialization below *****
;INIT NOP ;THREE NOP'S FOR PATCHING

```

```

OUT 2 ;SEND IT
RET ;DONE

;*****
;***** This is the Printer output Routine *****
;***** Modified for the HSIO-4 set with *****
;***** Base address of 20H. Switches on *****
;***** HSIO are 0010. See HSIO manual. *****
;*****
;*****
;***** OUTPUT TO SECONDARY (RIGHT) SERIAL PORT (PRINTER)
;***** Now is....MODIFIED FOR HSIO-4 WITH PORT ADDRESS OF 20H.
COUTR IN 23H ;GET USART STATUS
NOP ;ROOM FOR PATCHING
ANI 1 ;IS TX READY?
JZ COUTR ;LOOP TIL READY
MOV A,C ;BYTE FROM C TO A
NOP ;ROOM FOR PATCHING
OUT 22H ;SEND BYTE VIA DATA PORT
RET ;DONE
;
; Next stuff is all for the parallel ports. You may eliminate it
; with semi-colons if you are not going to use it. But don't
; forget what you did in case the parallel ports mysteriously
; fail to work someday. That means an HD-18 which connects to the
; parallel ports also. -b.c.
;
;OUTPUT TO PARALLEL PORT - IF CEMGEN TOLD THAT PRINTER
; IS PARALLEL, JUMP VECTOR ALTERED TO JUMP HERE
COUTP IN 6 ;MOTHERBOARD STATUS
ANI 1 ;TEST IO FLAG
JZ COUTP ;EXTERNAL DEVICE NOT READY YET
MVI A,20H ;RESET THE FLAG
OUT 6
MOV A,C ;CHAR TO BE SENT TO A
NOP ;ROOM FOR PATCHING
TOGGLEP ORI 80H ;SET STROBE FALSE
OUT 0 ;SEND CHARACTER
XRI 80H ;TOGGLE STROBE TO TRUE
OUT 0 ;KEEP SENDING
XRI 80H ;TOGGLE STROBE BACK TO FALSE
OUT 0
RET ;DONE
;
;NULL IMPLEMENTATION OF PRINTER TEST
PRITST XRA A ;LEAVE ZERO IN A
RET ;DONE
;*****
;***** INITIALIZATION CODE GOES BELOW. *****
;***** Performed only on Cold-Boot. *****
;*****
;*****

```

```

; NOP
; NOP
;
; INITIALIZE MOTHER BOARD AND SERIAL PORTS
  XRA A ;ZERO A
OUT 6 ; INIT MOTHERBOARD
OUT 6 ; REPETITIVELY
OUT 6
OUT 6
; STANDARD (LEFT) SERIAL PORT INITIALIZATION
MWI A,0CEH ; 2 STOP BITS, 16X CLOCK
OUT 3 ;& 8 DATA BITS, NO PARITY
MWI A,37H ;CMD: RTS,ER,RF,DIR,TXEN
OUT 3
; SECONDARY(RIGHT)SERIAL PORT INITIALIZATION
MWI A,0CEH ; 2 STOP BITS, 16X CLOCK
OUT 5 ;& 8 DATA BITS, NO PARITY
MWI A,37H ;CMD: RTS,ER,RF,DIR,TXEN
OUT 5
; PARITY INITIALIZATION
MWI A,40H ;DISABLE PARITY LOGIC
OUT 0C0H ;BEFORE READING UNWRITTEN RAM
LXI H,0 ;PREPARE TO CYCLE THROUGH RAM
;BEFORE SETTING PARITY FOR EACH KILOBYTE, CHECK IF REALLY RAM
TINXK MOV A,M ;GET FIRST BYTE OF KBYTE
CMA ;FORM ITS COMPLEMENT
MOV M,A ;CHANGE IT IN RAM
CMP M ;SEE IF CHANGED IN RAM
CMA ;RESTORE IT
;***NOTE, THESE FEW BYTES BETTER NOT BE ON A 1K BOUNDARY
MOV M,A ;FOR THEY COULD ZAP THEMSELVES!!!
JZ TINPAR ;IS RAM, GO INITIALIZATE IT
INR H ;NOT RAM, SO SKIP OVER
INR H ;THIS KILOBYTE
INR H
JMP TINKADV ;SKIP OVER INIT CODE
TINPAR MOV B,M ;READ BYTE FROM RAM
MOV M,B ;RESTORE IT WITH CORRECT PARITY
INR L ;INCREMENT LOW ORDER ADDR BYTE
JNZ TINPAR ;LOOP IF NOT END OF 256B BLOCK
TINKADV INR H ;INCREMENT BLOCK NUMBER
JZ TINU ;WRAPPED AROUND, EXIT
MOV A,H ;SEE IF ON 1K BOUNDARY
ANI 3
JNZ TINPAR ;NO, TIGHT LOOP THRU ONE KILOBYTE
JMP TINKOX ;YES, TEST NEW KB FOR REAL RAM
;
; PARITY ALL CORRECT, ENABLE PARITY TESTING
TINU MWI A,41H ;TURN PARITY ON
OUT 0C0H ;ON RAM BOARDS
;
; REMOVE ANY JUNK APPEARING IN INPUT PORTS
IN 2 ;FROM STD PORT
IN 4 ;FROM SECOND PORT
;
; INIT PARALLEL PORTS
MWI A,30H ;RESET PAR. INPUT FLAG

```

```

OUT 6
MWI A,60H ;SET PAR. OUTPUT FLAG
OUT 6
;WRAP IT UP BY SENDING A CR TO THE PRINTER & EXIT
MWI C,0DH ;A CARRIAGE RETURN
JMP USERBASE-700H+15 ;GOTO PRINT JUMP IN BIOS VECTOR
;***** END OF INITIALIZATION CODE *****
;
; ***** PARALLEL PRINTER OUTPUT WHEN HUNG ON HD-18 HARD DISK
COUTP18 MWI A,20H ;SET COMMAND MODE
OUT 6
MWI A,0FEH ;SELECT CONTROLLER 101
OUT 0
MWI A,0E1H ;SET CITER TO I/O MODE
OUT 0
MWI A,60H ;SET DATA MODE
OUT 6
P18WAIT IN 6 ;TEST CITER'S FO FLAG
ANI 8 ;FOR READY
JZ P18WAIT ;LOOP UNTIL READY
IN 0 ;CLEAR CONTROLLER'S FO FLAG
MOV A,C ;CHARACTER TO SEND
ORI 80H ;SET STROBE FALSE
OUT 0 ;PUT CHAR IN PORT
XRI 80H ;TOGGLE STROBE *TRUE
OUT 0
XRI 80H
OUT 0
MWI A,20H ;BACK TO COMMAND MODE
OUT 6
MWI A,0FFH ;DESELECT CONTROLLER
OUT 0
RET ;DONE

```

```

; THE LAST 80H BYTES OF THIS 512 BYTE USER AREA IS NEEDED
; BY THE DISK BIOS FOR TWO SPECIAL PURPOSES.
; 1. SUPPORT NON-STANDARD DISK CONTROLLER FROM ADDRESSES
; 2. AUTOSTART HDBOOT AND SUPPLY THE WORK-FILE PATHNAME
; FOR NORTH STAR HARD DISK SYSTEMS.
;
; IF YOU HAVE NEITHER A NORTH STAR HARD DISK
; NOR A NON-STANDARD MDS FROM ADDRESSES,
; THEN YOU MAY FREELY OVERWRITE THE AREA AND USE THE
; ENTIRETY OF SECTOR 8 IN THE SYSTEM TRACK AREA.
;
; IF YOU DO HAVE A HARD DISK OR NON-STANDARD MDS PROMS,
; THEN YOU MUST LEAVE THE LAST 80H BYTES INTACT.
; IN THE SYSGEN RAM IMAGE, YOU MUST LEAVE 3380H THRU
; 33FFH UNMODIFIED, USE ONLY 3200H-337FH FOR YOUR
; REPLACEMENT USER AREA
;
; END OF NEWUSER.ASM

```


PROGRAMS FOR N* DOS

Leonard Morgenstern
304 Rheem Blvd.
Moraga, CA 94556
(415) 376-5241

(AND

BASIC)

Here are 3 programs for North Star DOS and BASIC:

1. DISKLIST was published in the Compass about 2 years ago. This version fixes a couple of minor bugs. It also computes the maximum number of directory entries, which on my system is over 2000. The original version accepted only 1000. Also, the original version required that the DOS be at 2000 hex; this is no longer necessary.
2. PRETTY will take a file created by a word processor and turn it into North Star BASIC statements. You may select DATA statements enclosed in quote marks, |#P statements, or REM statements. There is a certain amount of flexibility allowed with respect to the format of the source file.
3. RESTTYP2 will change a file into type 2. For example, the file created by PRETTY or MEMTODAT is type 3. To run as a BASIC program, it must be changed to type 2. It is not enough to change it via the DOS by means of a TY command, as the length byte also needs to be fixed.

```
10 !"DOCUMENTATION FOR DISKLIST"\!
20 PRINT "LISTING DEVICE, USUALLY SCR=0, PTR=2 ",\INPUT " ",P1
30 IF P1<>0 THEN LINE#P1,80
40 IF P1=0 THEN INPUT "NUMBER OF LINES ON YOUR TERMINAL ",L
50 DIM Q$(80)
60 K=1
70 READ Q$ \ IF Q$<>"*" THEN 100 ELSE END
100 !#P1,Q$ \ K=K+1 \ IF K<L-1 OR P1<>0 THEN 70
110 !"PRESS ANY KEY TO CONTINUE" \ Q$=INCHAR$(0) \ GOTO 60
120DATA"
130DATA"This program will combine the disk directories of a number"
140DATA"of NorthStar diskettes and list them three ways, by disk,"
150DATA"alphabetically by name, and by file type. The listing is in"
160DATA"three columns."
170DATA" It was originally published in the COMPASS, which is"
180DATA"the publication of the International North Star Users"
190DATA"Group."
200DATA" Several minor bugs have been corrected, and the"
210DATA"program now no longer requires DOS at 2000."
220DATA" The program must be configured for your system by"
230DATA"changing the personalization commands near the beginning."
240DATA"Two areas of reserved memory are required, one 2k, and the"
250DATA"other about 20 bytes."
260DATA"The maximum number of directory entries is calculated by the"
270DATA"computer, and is about 2000."
280DATA" The program will request you to insert diskettes one"
290DATA"at a time on drive 1. To each diskette, assign a"
300DATA"3-character name and a short description. These will be"
310DATA"printed, followed by the directory listing. After the last"
320DATA"diskette enter a CR instead of a name, and the first sort"
330DATA"phase will begin. Restore the paper to the beginning, and"
340DATA"press a key when the computer asks you to. The computer"
350DATA"will now list the directory sorted by file name.The next"
360DATA"sort phase is handled the same way."
370DATA" A useful hint is to arrange your diskettes in some"
380DATA"logical order before listing them. Then, the printed"
390DATA"directories will be easier to read."
400DATA" The factor A1 in the initialization needs to be"
410DATA"computed for your machine, as follows. Record the time for"
420DATA"the sort phase and the number of directory entries sorted."
430DATA"A1 is given by the formula, A1=T/N/LOG(N)"
440 DATA" T=time"
450 DATA" N=number "
460 DATA "!"
```

```

10 REM SAVE DISKLIST
20 REM
30 !"THIS PROGRAM WILL COMBINE N* DISK DIRECTORIES INTO"
40 !"ONE DIRECTORY, SORT AND LIST 3 WAYS"
50 !"BY LEONARD MORGENSTERN 1983"
60 !"FULL DOCUMENTATION IS IN FILE LISTDOC"
70 REM
80 D9=(11*16+3)*256 \ REM ADDRESS OF DOS
90 L1=57344 \ REM ADDRESS OF 800H (2K) OF FREE MEMORY
100 REM (FOR DIRECTORY BUFFER)
110 L=0 \ REM ADDRESS OF 20 BYTES OF FREE MEMORY FOR SUBROUTINES
120 P2=3 \ REM DEVICE NUMBER ASSIGNED TO PRINTER
130 B9=7 \ REM CHARACTER FOR BELL OR WHISTLE
140 A1=.116 \ REM SPEED FACTOR (SEE DOCUMENTATION)
150 REM
160 !"===== END OF PERSONALIZATION ====="
170 PRINT "LISTING DEVICE, 0=SCREEN, ",P2,\INPUT " =PRINTER ",P1
180 !
190 LINE#P2,80
200 IF P1=0 THEN 260
210 !"CHECK PRINTER. INSERT LOTS OF PAPER. THERE WILL BE"
220 !"THREE PASSES. PAPER SHOULD BE RESET TO BEGINNING AT"
230 !"START OF EACH PASS. YOU WILL BE PROMPTED BEFORE PRINTING BEGINS"REM
240 !
250 !\INPUT "TODAY IS ",D$\ I#P1,"TODAY IS ",D$\ I
260 !"LOAD DISKETTES ONE AT A TIME ON DRIVE 1. ON REQUEST, TYPE"
270 !"NAME OF DISKETTE (3 CHARACTERS MAXIMUM)"
280 !"AND DESCRIPTION OF DISKETTE (20 CHARACTERS MAXIMUM)"
290 L3=INT(FREE(0)/12-100)
300 DIM N$(L3*12),A$(21),B$(15)
310 N0=1 \ REM N0=POINTER FOR N$ STRING
320 !\I(N0-1)/12," ENTRIES SO FAR. DO NOT EXCEED",L3
330 ITAB(10),"CR ONLY TO START SORTING"
340 INPUT "DISKETTE NAME (MAX 3 CHARS) : ",D$
350 IF LEN(D$)=0 THEN 630
360 A$="" \ IF P1<>3 THEN 380
370 INPUT "DESCRIPTION OF DISKETTE (MAX=20 CHAR) ",A$
380 !\I"DISKETTE ",D$," ",A$
390 !"IS THAT CORRECT? HAVE YOU CHECKED DISK DRIVE?"
400 !"Y=YES"
410 B$=INCHAR$(0)\ IF B$<>"Y" THEN 320
420 IF P1<>3 THEN 450
430 !#P1\I#P1,"DISKETTE: ",D$
440 !#P1,A$\I#P1
450 ERASET 2300,E1,E2
460 I=FNL(1,P1,L)
470 ERASET 2310,E1,E2
480 I=FNC(L,4,0,L1,1,1,D9)
490 ERASET
500 IF I=0 THEN 520
510 !"DISK ERROR ",D$ \ GOTO 320
520 !"STAND BY"
530 FOR I=L1 TO L1+16*128-1 STEP 16
540 IF EXAM(I)=32 THEN 610
560 FOR J=0 TO 7 \ A$=A$+CHR$(EXAM(I+J)) \ NEXT
570 I1=EXAM(I+12)\ IF I1>=128 THEN I1=I1-128
580 A$=A$+CHR$(I1)
590 D$=D$+" " \ A$=A$+D$(1,3)
600 N$(N0)=A$ \ N0=N0+12
610 NEXT
620 GOTO 320
630 !"PASS 1 COMPLETE. RESTORE PAPER TO BEGINNING"
640 I1=N0/12\I1=A1*LOG(I1)#I1
I2=INT(I1/60) \ I3=INT(I1-I2*60)
I" EACH SORT PHASE WILL TAKE ABOUT",I2," MINS",I3," SECS"
O=1 \ GOSUB 1090\REM SORT
O=1 \ GOSUB 1090\REM SORT
680 !"PASS 2 COMPLETE. RESTORE PAPER TO BEGINNING"
690 FOR I=1 TO N0-1 STEP 12
700 A$=N$(I+8,I+8)+N$(I,I+7)+N$(I+9,I+11)
710 N$(I)=A$
720 NEXT
730 O=2 \ GOSUB 1090\REM SORT
740 !"JOB COMPLETE"
750 END
760 REM THIS SUBROUTINE WILL SET UP A DCOM COMMAND
770 REM AT LOCATION L
790 DEFFNC(L,S,D,M,A,D1,U,D9)
800 REM L=LOCATION,S=SECTORS,D=DISK ADDR,M=MEM ADDR,
810 REM A=ACTION TO TAKE (0=WRITE,1=READ),D1=DENSITY(1=DD),
820 REM U=DISK UNIT, D9=ADDR OF DOS)
830 REM RETURNS 0=OK, 1=ERROR
840 REM
850 RESTORE 960
860 FOR I1=0 TO 16\ READ I2 \ FILL L+I1,I2 \ NEXT
870 FILL L+1,S
880 FILL L+4,A
890 FILL L+3,128*D1+U
900 I1=FNM(D,256)\I2=INT(D/256)
910 FILL L+6,I1
920 FILL L+7,I2
930 FILL L+10,D9/256
940 RETURN CALL(L,M)
950 FNEND
960 DATA 62,0 \ REM MVI A,0
970 DATA 1,0,0 \REM LXI B,0
980 DATA 33,0,0 \ REM LXI H,0
990 DATA 205,34,0 \ REM CALL XX22
1000 DATA 33,0,0 \ REM LXI H,0
1010 DATA 208,35,201 \ RNC INX H RET
1020 REM (MODULUS FUNCTION)
1030 DEFFNM(X,M)=X-INT(X/M)*M
1040 REM *****
1050 REM FAST SORT, MINIMUM COMPARISON, MIN. STEPS
1060 REM VARIATION OF WOODRUM SORT BY RICHARD HART
1070 REM PUBLISHED IN CREATIVE COMPUTING JAN/FEB 1978
1080 REM NEXT LINE SHOULD SET N=NUMBER OF WORDS,W=LENGTH
1090 N=(N0-1)/12\W=12

```

```

1100 Z=P1 \ REM P1 IS USED DURING SORT
1110 W1=W-1
1120!*SORTING...
1130 REM ENTRY
1140 IF O<>1 THEN 1170
1150 L9=INT(LOG(N)/LOG(2)+.9) \ REM LINKS=N+LOG2(N)+2 ELEMENTS
1160 DIM L(N+L9+2) \ REM AUTO LINK ARRAY
1170 FOR I=1 TO N+L9+2 \ L(I)=0 \ NEXT
1180 I=O\K1=O\M1=O\T2=O\T4=O
1190 REM K1,I,M1,T2,T4=O
1200 LET J=N+1 \ REM HEAD OF SEQ 1
1210 L(1)=1 \ L(J)=1 \ K2=1
1220 IF N<=1 THEN 1950 \ REM EXIT, NOTHING TO SORT
1230 S1=N \ REM NUMBER OF LEAVES
1240 REM CLIMB TREE
1250 IF S1 < 4 THEN 1310 \ REM LOW ORDER TWIG VALUE
1260 K2=K2*2 \ REM TOTAL # OF TWIGS
1270 B2=S1/2
1280 S1=INT(B2)
1290 T4=T4+(B2-S1)*K2
1300 GOTO 1240
1310 REM INITIAL CALCULATIONS
1320 T4=K2-T4 \ REM # OF LO-ORDER TWIGS
1330 B2=K2/2 \ REM HI-BIT VAL OF BIN CTR
1340 REM NEXT TWIG
1350 IF K1=K2 THEN 1950 \ REM EXIT SORT COMPLETE
1360 T1=K1+1 \ K1=K1+1 \ REM TWIG #
1370 B1=B2 \ REM HI BIT VALUE
1380 T3=T2 \ REM PREV REFLECTED TWIG #
1390 REM ADD 1 TO REFL BIN CTR & CARRY
1400 T1-T1/2
1410 IF INT(T1)<T1 THEN 1460 \ REM NO MORE CARRIES
1420 M1=M1+1 \ REM # OF MERGES & CARRIES
1430 T2=T2-B1
1440 B1=B1/2 \ REM NEXT BIT VALUE
1450 GOTO 1390 \ REM CARRY 1
1460 REM TWIG CALCULATIONS
1470 T2=T2+B1 \ REM REFLECTED TWIG #
1480 IF S1=2 THEN 1540 \ REM 2-TWIGS & 3-TWIGS
1490 REM 3-TWIGS & 4-TWIGS
1500 IF T3<T4 THEN 1550 \ REM LO-ORDER TWIG (3-TWIG)
1510 REM 4-TWIG
1520 M1=M1 \ REM DISENGAGE # OF MERGES
1530 GOTO 1620
1540 IF T3<T4 THEN 1600 \ REM LO-ORDER TWIG (3-TWIG)
1550 REM 3-TWIG
1560 M1=M1+1 \ REM # OF MERGES
1570 I=I+1 \ REM NEXT LEAF
1580 L(I)=I \ L(J)=I \ REM GENERATE A LEAF
1590 J=J+1 \ REM NEXT SEQUENCE HEAD
1600 REM 2-TWIG
1610 M1=M1+1 \ REM # OF MERGES
1620 I=I+1 \ REM NEXT LEAF
1630 L1=I \ L(I)=I \ L(J)=I \ REM GENERATE A LEAF
1640 LO=J \ REM HEAD OF OLDER LEAF (LAST LINE)

1650 J=J+1 \ REM HEAD OF LATEST LEAF (NEXT 2 LINES)
1660 I=I+1 \ REM NEXT LEAF
1670 L2=I \ L(I)=I \ L(J)=I \ REM GENERATE A LEAF
1680 GOTO 1740 \ REM MERGE LEAVES
1690 REM MERGE TWIGS & BRANCHES
1700 J=J-1 \ REM HEAD OF LATEST BRANCH OR TWIG
1710 LO=J-1 \ REM HEAD OF OLDER BRANCH OR TWIG
1720 L1=L(LO) \ REM HEAD OF SEQUENCE 1
1730 L2=L(J) \ REM HEAD OF SEQUENCE 2
1740 P1=1+(L1-1)*W \ P2=1+(L2-1)*W \ REM WORD POINTERS
1750 IF N$(P1,P1+W1) <= N$(P2,P2+W1) THEN 1830 \ REM STAY IN 1
1760 L(LO)=L2 \ REM SWITCH TO SEQ 2
1770 LO=L2 \ REM TOP LEAF IN SEQ 2
1780 L2=L(LO) \ REM NEXT LEAF IN SEQ 2
1790 IF L2=L0 THEN 1880 \ REM END OF SEQ 2
1800 P1=1+(L1-1)*W \ P2=1+(L2-1)*W \ REM WORD POINTERS
1810 IF N$(P1,P1+W1) > N$(P2,P2+W1) THEN 1770 \ REM STAY IN 2
1820 L(LO)=L1 \ REM SWITCH TO SEQ 1
1830 LO=L1 \ REM TOP LEAF IN SEQ 1
1840 L1=L(LO) \ REM NEXT LEAF IN SEQ 1
1850 IF L1<>LO THEN 1740 \ REM NOT END OF SEQ 1
1860 L(LO)=L2 \ REM SWITCH TO SEQ 2
1870 GOTO 1890
1880 L(LO)=L1 \ REM SWITCH TO SEQ 1
1890 M1=M1-1 \ REM # OF MERGES
1900 IF M1>0 THEN 1690
1910 IF M1=0 THEN 1340
1920 REM GENERATE 2ND HALF OF A 4-TWIG
1930 M1=1-M1 \ REM RE-ENGAGE # OF MERGES
1940 GOTO 1620
1950 REM EXIT
1960 LO=N+1 \ REM FIRST LINK IN SEQ
1970 PRINT CHR$(B9)
1980 I"PASS",O+1," READY TO PRINT. ",
1990 I"PRESS ANY KEY TO PROCEED"\I1$=INCHAR$(0)
2000 FOR I= 1 TO N
2010 LO=L(LO) \ REM FOLLOW LINKS
2020 P=1+(LO-1)*W \ REM WORD POINTER
2030 GOSUB 2090 \ REM PRINT WORD
2040 NEXT I
2050 P1=Z
2060 RETURN
2070 REM
2080 REM THIS SUB PRINTS THE ENTRY
2090 ON O GOTO 2100,2130
2100 IF Z=3 THEN I#Z,TAB(35),
2110 I#Z,N$(P,P+7),%3I,ASC(N$(P+8)), " ",N$(P+9,P+11)
2120 GOTO 2150
2130 IF Z=3 THEN I#Z,TAB(60),
2140 I#Z,%3I,ASC(N$(P)), " ",N$(P+1,P+8), " ",N$(P+9,P+11)
2150 RETURN
2160 REM
2170 REM THIS WILL LIST A DISK DIRECTORY
2180 REM D=DRIVE, P=PORT, L=AREA TO SET UP COMMAND
2190 REM

```

```

2200 DEFNL(D,P1,L)
2210 RESTORE 2220
2220 DATA 235,124,195,37
2230 FOR I1=L TO L+3
2240 FILL L+4,D9/256
2250 READ I2 \ FILL I1,I2
2260 NEXT
2270 RETURN CALL (L,256*D+P1)
2280 FNEND
2290 REM DISK ERROR ROUTINES
2300 GOSUB 2340 \ "CHECK DISK!" \ GOTO 420
2310 GOSUB 2340 \ "PROBABLE FAULTY DISK"
2320 I"TYPE COMMAND (1=CONTINUE,2=DO NOT LIST THIS DISK)"
2330 IF INCHAR$(0)<"2" THEN 490 ELSE 320
2340 I#P1, "***DISK ERROR ",E1,E2 \ RETURN
2350 REM
2360 REM THIS SUBROUTINE LISTS N$
2370 I"NUMBER OF CHARS=",NO
2380 FOR I1=1 TO NO-1
2390 A$=N$(I1,I1)
2400 IF ASC(A$)<32 THEN I"({",ASC(A$),"}", ELSE I A$,
2410 NEXT
2420 I" "
2430 RETURN

10 REM PRETTY2
20 REM IMPROVED VERSION, SEPT 27, 1983
30 REM BY LEONARD MORGENSTERN
40 DIM I1$(40),I2$(40),L$(90)
50 REM ===== PERSONALIZATION =====
60 B9=7 \ REM ASCII OF BELL OR WHISTLE
70 B8=11 \ REM ASCII OF CLEAR SCREEN
80 L1=0 \ REM Bytes to omit at start of each line
90 L2=0 \ REM Line terminator, 0=CR, 1=CR-LF
100 E3=0 \ REM End of file byte
110 I#P"THIS is PRETTY, a program that will convert a file"
120 I#P"prepared by a word processor into NorthStar BASIC"
130 I#P"statements." \ I
140 I"DO you wish more documentation ", \ IF FNY(0)<>1 THEN I60~ END
150 GOSUB 930
160 I
170 F1$="???" \ F2$="???"
180 GOTO 210
190 CLOSE#1 \ CLOSE#2
200 ICHR$(B8)
210 I"FOR each parameter, if current value is ok, press CR."
220 I" Otherwise enter value followed by CR."
230 I
240 S1=10000\S2=10\E3=0
250 F1$=FNG$( "Source file (name,unit)",F1$)
260 F2$=FNG$( "File to be created (name,unit)",F2$)
270 F1=FILE(F1$) \ IF F1>-1 THEN 290
280 ICHR$(B9) \ I"ABORTED, ",F1$, " DOES NOT EXIST" \ END

```

```

290 OPEN#1F1,F1$
300 F2=FILE(F2$) \ IF F2>-1 THEN 350
310 I \ IF F2$, " does not exist. It will be created." \ I
320 I"Approximate number of text lines"
330 J=0 \ J=FNG(" in "+F1$,J)
340 CREATE F2$,INT(J*80/256+1),0 \ F2=0
350 I
360 I"Is that OK ", \ ON FNY(0) GOTO 370,190
370 GOSUB 920
380 OPEN#2F2,F2$
390 I"The following information is needed regarding the format"
400 I" of the source file, "F1$, " "
410 I \ I"Number of bytes to be deleted at the start"
420 L1=FNG(" of each line of "+F1$,L1)
430 I"Terminator of each line is "
440 L2=FNG(" 0=CR only: 1=CR+LF",L2)
450 E3=FNG("EOF mark",E3)
460 GOSUB 920
470 I"Please select the desired format of the output file,"F2$
480 I
490 L3=FNG("0=REM, 1=i#P, 2=DATA",L3)
500 S1=FNG("Starting line number",S1)
510 S2=FNG("Line number interval",S2)
520 I"Is that OK ", \ ON FNY(0) GOTO 530,190
530 IF L3=0 THEN S1$=".." + CHR$(143) + CHR$(34)
540 IF L3=1 THEN S1$=".." + CHR$(146) + "P" + CHR$(34)
550 IF L3=2 THEN S1$=".." + CHR$(135) + CHR$(34)
560 L$=S1$ \ IF L1=0 THEN 570 \ FOR I1=1 TO L1 \ READ#1,&J \ NEXT
570 READ#1,&J \ IF J=E3 OR J=13 THEN 590
580 L$=L$+CHR$(J) \ GOTO 570
590 L$=L$+CHR$(34)+CHR$(13)
600 L$(1,1)=CHR$(LEN(L$))
610 I1=INT(S1/256) \ I2=S1-256*I1
620 L$(2)=CHR$(I2)+CHR$(I1)
630 FOR I1=1 TO LEN(L$) \ WRITE#2,&ASC(L$(I1,I1)) \ NEXT
640 S1=S1+S2 \ I" "
650 IF J=E3 THEN 660 \ IF L2<>0 THEN READ#1,&J \ GOTO 560
WRITE#2,&I1 \
I"TRANSFER COMPLETE. IT WILL BE NECESSARY TO SET ",F2$
TO TYPE 2 BY RUNNING THE PROGRAM RESTYP2"
END
/uu REM===
710 DEFFNG$(I1$,X$)
720 REM THIS WILL UPDATE HEX STRING
730 I1$,TAB(30),X$,TAB(40),
740 INPUT" ",I1$
750 IF LEN(I1$)<>0 THEN 760 \ I X$ \ RETURN X$
760 I \ RETURN I1$
770 FNEND
780 REM ===
790 DEFFNG(I1$,X)
800 REM THIS WILL UPDATE NUMERICAL VALUE
810 I1$,TAB(30),X,TAB(40),
820 INPUT" ",I1$
830 IF LEN(I1$)<>0 THEN 840 \ I X \ RETURN X

```

```

840 ERRSET 860, E1,E2
850 I1=VAL(I1$)\ ERRSET \ I \ RETURN I1
860 I" ????" \ I TAB(40), \ GOTO 820
870 FNEND
880 DEFFNY(U)
890 I"(Y/N)? ", \ I1$=INCHAR$(U) \ I1$
900 I1=ASC(I1$)\ IF I1=89 OR I1=121 THEN RETURN 1 ELSE RETURN 2
910 FNEND
920 FOR I1=1 TO 63 \ I" ", \ NEXT \ I \ RETURN
930 I#P
940 I#P" PRETTY enables a programmer to create neat
950 I#P" documentation for NorthStar BASIC programs. A text
960 I#P" file may be converted into DATA, I#P, or REM
970 I#P" statements."
980 I#P" For maximum efficiency, change the personalization
990 I#P" section to match your system."
995 I#P" Written by Leonard Morgenstern. Updated Sept, 1983."
1000 RETURN

10 REM SAVE RESTYP2
20 REM THIS WILL FIX THE DIRECTORY ENTRY FOR A TYPE 2 FILE
30 REM THAT WAS NOT STORED FROM A BASIC PROGRAM
40 REM BY LEONARD MORGENSTERN/ Updated 9/25/83
50 I" This is RESTYP2, which will convert a file to type 2 "
60 I" Written by Leonard Morgenstern. Modified Sept. 83" \ I
70 I" Do you wish more documentation (Y/N)? ",
80 I3$=INCHAR$(0) \ I3$ \ IF I3$="Y" THEN GOSUB 890
90 DIM I1$(20), X$(20), I2$(20), F$(20)
100 I
110 I" For each parameter, input value. If current value is ok, "
120 I" then press CR" \ I
130 F$="???" \ F$=FNG$( "File name, unit", F$)
140 I2$="2000" \ I2$=FNG$( "Origin of DOS (hex)", I2$) \ O1=FNH1(I2$)
150 I" Address of 40 bytes of available"
160 I2$="0000" \ I2$=FNG$( " free memory (hex)", I2$) \ O2=FNH1(I2$)
170 Y$="N" \ Y$=FNG$( "Compress file (Y/N)?", Y$)
180 F$=F$+" "
190 I1=FILE(F$)
200 IF I1>-1 THEN 210 \ IF$, " does not exist" \ END
210 OPEN#O1,I1,F$
220 REM NEXT DETERMINE NUMBER OF BYTES IN FILE
230 L=0
240 READ#O1,L,&X
250 IF X=1 THEN 270
260 L=L+X \ GOTO 240
270 L=L+1
280 IF$, " is", L, " bytes long. ",
290 X=L \ GOSUB 490 \ IF X2<>0 THEN X1=X1+1
300 I" Requires", X1, " blocks."
310 Q=X1
320 Z=END(F$)
330 FILL Z+5,Q
340 I1=EXAM(Z+4) \ IF I1>127 THEN I1=130 ELSE I1=2
350 FILL Z+4, I1 \ I4=I1
360 I1=ASC(Y$(1,1)) \ IF NOT(I1=89 OR I1=89+32) THEN 400

370 IF I4=2 THEN 390
380 IF INT(Q/2)*2<>Q THEN Q=Q+1 \ Q=Q/2
390 X=Q \ GOSUB 490 \ FILL Z+2, X2 \ FILL Z+3, X1
400 I2=CALL(O1+31)
410 I" DONE"
420 END
430 DEFFNH$(A)
440 REM CONVERT A TO HEX DIGIT (STRING)
450 IF A<0 OR A>15 THEN RETURN "?"
460 IF A<10 THEN RETURN CHR$(48+A) ELSE RETURN CHR$(55+A)
470 FNEND
480 REM THIS SUBROUTINE SPLITS X INTO X1 AND X2 SUCH THAT
490 REM X=256*X1+X2
500 X1=INT(X/256) \ X2=X-X1*256 \ RETURN
510 REM
520 REM THIS SUBROUTINE SPLITS Y INTO Y1 AND Y2 SUCH THAT
530 REM Y=16*Y1+Y2
540 Y1=INT(Y/16) \ Y2=Y-Y1*16 \ RETURN
550 REM.....
560 REM THIS FUNCTION FINDS DIRECTORY ENTRY
570 DEFFND(I1$)
580 RESTORE 590
590 DATA 33,8,0,62,1,195,28,240
600 FOR I1=02 TO 02+7 \ READ I2 \ FILL I1,I2 \ NEXT
610 X=02 \ GOSUB 500 \ FILL O2+1,X2+8 \ FILL O2+2,X1
620 X=01 \ GOSUB 500 \ FILL O2+7,X1
630 FOR I1=1 TO LEN(F$) \ FILL O2+7+I1,ASC(F$(I1,I1)) \ NEXT
640 RETURN CALL ( O2)
650 FNEND
660 REM==
670 DEFFNG$(I1$,X$)
680 REM THIS WILL UPDATE HEX STRING
690 I1$,TAB(30),X$,TAB(40),
700 INPUT I1$,I1$
710 IF LEN(I1$)<>0 THEN 720 \ I X$ \ RETURN X$
720 \ RETURN I1$
730 FNEND
740 REM convert hex to decimal, return -1 on error
750 DEFFNH1(I1$)
760 I1=LEN(I1$)
770 IF I1=0 THEN RETURN -1
780 IO=0
790 FOR I2=1 TO I1
800 I3=ASC(I1$(I2,I2))
810 IF I3<48 OR I3>70 THEN EXIT 860
820 IF I3>57 AND I3<65 THEN EXIT 860
830 IO=IO*16+I3-48-(I3>65)*7
840 NEXT
850 RETURN IO
860 RETURN -1
870 FNEND
880 REM
890 I#P
900 I#P" NorthStar BASIC will load a program properly only if
910 I#P" the file is type 2 and the type-dependent information
920 I#P" is correctly set. This program will modify a directory
930 I#P" entry to conform with these requirements, and will
940 I#P" also compress the file, if desired."
950 RETURN

```

INSUA:

I was somewhat troubled by some of the letters in the last issue of the Compass which were somewhat derogatory and which blamed you for not doing this and that.

I am grateful that there is anyone around who is using these machines and who is willing to share the benefit of their experience, software, and more importantly information about common problems with the machines and the company.

This information about hidden "gotcha's" like the mysteries of lines 20 and 67 of the so-called S-100 bus have been very helpful to me in the exorcism of several demons that couldn't otherwise be explained. It's also helpful to hear that others are having problems with service and parts. SORBUS seems to do a good job, at least locally. They are expensive, however. They have a generous replacement exchange deal with NorthStar boards. If they can't fix the board by replacing chips they will give you a new board for \$190 or so. I found the local operation in the Detroit area willing to talk and very helpful in general.

Part of my hardware problems have been related to the fact that when I purchased my NorthStar Horizon there were a lot of NorthStar kits around. Evidently, the vendor that I was dealing with purchased one of these kits, put it together and couldn't make it work. As a result an alien I/O board was added and a custom DOS. The system I bought was therefore a kluge. The custom DOS does some very funny things at times and, of course, software written for NorthStar Horizon I/O doesn't work at all.

My dealings with NorthStar have been mixed. When calling to find the names of local dealers I have often gotten names and phone numbers of vendors who say they don't deal in this machine and offer no advice for who does. After many expensive phone calls they did come up with a local jobber who can get parts and boards from the local distributor but who offers no other support.

Please keep up the good work. I think the letters I mentioned above are probably a symptom of the frustration that some of us have experienced.

Sincerely,

Mel L. Barclay, Ann Arbor, MI

INSUA:

Have received Vol. III, no. 3, also some backnumbers as ordered--however I did send

\$40 hoping to get both Vol. I and II! Are they all gone? If so I shall be very sad--Compass is the best thing through my mailbox for a very long time. It is so tantalizing to read those references to Vol.I Compass articles that I do not have!

When I got my Horizon kit back in 1979 there was a rumour about a UK Northstar users group. But I wrote to the dealer who was supposedly organising it and they said they had not enough support to go ahead. The magazine Practical Computing in June 1981 published an article I wrote about my building the machine, but in spite of my comments about are there any more Horizon owners, I only got 2 letters.

So thank you again for Compass and the work that gets it to us. I do feel however that with the postage it costs you to reach me I am not paying my way--I imagine most overseas members would not object to paying more.

I must concur with Burt Andrews (Compass III, no. 1. p. 17) with reference to Allen Ashley. I rate my PDS as the best accessory to my Horizon I've bought. (I didn't know about INSUA then!)

Magazines--reference James C. Matthews (Compass III, no. 2, p. 25): I get Byte, which I feel is worth its cost (that's where I met your add). At work we take Microcomputing, but I reckon that has gone downhill lately. The British magazines are junk, mainly devoted to silly games. There is over here the Amateur Computer Club which publishes The Accumulator newsletter, very good too although at present a bit shaky due to lack of support.

I note the current Compass contains a suggestion by Peter de Jaeger about software contributions--I'm no hotshot programmer, just another pilgrim eager to learn. But I do have a couple of games I wrote plus a filter for my old Teletype that had a non-standard print head. Now I have just got a Juki Daisywheel, so I'll send you listings if they'll help anyone.

Best Regards,

Gordon J. Mitchell, Gayton Le Marsh, Alford
Lincolnshire, U.K.

[Back issues of Compass are available. Occasionally orders for back issues go astray (in the mail??). Please inform us of any problems. We plan to publish a full index of Volumes I-III of Compass in the next issue, which will be Volume IV, no. 1. Submissions of all kinds are always welcome! --Ed.]



Insua

***Publishers of the Compass Newsletter
International NorthStar Users Asociation
PO Box 2910 • Fairfield, CA 94533***

*Bulk Rate
U.S. Postage
PAID
Walnut Creek
Permit No. 203*

TATE 276G
JAMES TATE
23914 SPRING DAY LN.
SPRING, TX 77373