A Not-So-Fast Renumberer for OSI BASIC

✓

Written in BASIC, this utility makes your listings neat and tidy.

John W. Aughey
27384 Lamplighter Lane
Elkhart IN 46514

This article describes a routine that will renumber BASIC programs for the Ohio Scientific BASIC-in-ROM computers. The program itself is written in BASIC and was designed on and for my personal machine, a Challenger C2-4P. However, it has also been tested and found to work without modification on the Challenger C1-P and C2-8P machines. Hence, any OSI computer with BASIC-in-ROM can make use of this renumbering routine. I would like to thank Phil Thornton of Elkhart Computer for providing a Challenger C1-P on which to test the program.

I decided to design this program and write this article for three major reasons. First, I have been the proud owner of my C2-4P for a number of months now and, as a result, have written a sizable library of BASIC programs that I would like to tidy up and expand. Second, I hope to make a few bucks from publishing this article so I can buy more hardware to write more programs that will need to be renumbered. And finally, in a February 1979 letter to the editor (p. 20), E. Morris of Midland, Michigan, said he would not renew his subscription unless there was an article oreinted toward us Ohio Scientific users in the next eleven months. I'm always glad to keep a fellow OSI user happy.

After having used their machines for a reasonable period of time, most OSI users would agree that one significant feature absent from the OSI version of Microsoft BASIC-in-ROM is the ability to renumber an existing program. This is a shortcoming that, until recently, I had managed to circumvent manually by writing programs with large gaps in the statement numbers and renumbering manually from printed listings when the source got too shabby to share with fellow programmers. However, my professionalism (I program operating system software for an Amdahl 470/V5 to support my hobby and family) got to me recently, and I finally decided that if I can renumber by hand, then I should certainly be able to tell the 6502 how to do it by itself.

In the process of collecting ideas for an OSI renumber routine, I read a number of articles by others who have written renumber routines for other systems—some in machine code and others in BASIC, some for 6502 machines and some not. The common foundation for all of these routines is a knowledge of how the BASIC interpreter stores the user's program in memory for execution, and I knew this was the key to designing a renumber routine for OSI's version of BASIC.

OSI's BASIC-in-ROM stores a user's source program starting at decimal location 769 in RAM. Each statement is composed of a four-byte header, followed by the compressed statement and terminated with a single byte of zeros. The four-byte header contains two 2-byte data words. The first word is the address of the next sequential statement, or zeros if this is the last statement in the program. The second word contains the statement number in binary format.

Routine Design

My first attempt at writing a renumber program was designed to renumber only the statements themselves, with no consideration of renumbering GOTOs, GOSUBs, THENs or RUNs embedded in the text of the statements. This was a relatively simple task that involved chaining from one statement to the next and inserting the new binary statement number into the second data word in the header I mentioned before.

The crux of this simpleminded renumberer is contained in lines 32000–32010 of the final version (see the listing). This first attempt at renumbering proved quite useful, but it was still a nuisance to have to go back and manually renumber the GOTOs, etc.

The tricky part comes when you go back and attempt to renumber the internals of the statements. As others who have written renumber routines have found, there is an inconsistency in the way statement numbers are stored. The numbers on the statements themselves are in binary form, but the statement number references in GOTOs, etc., are in ASCII.

Fortunately, the OSI BASIC has the very useful STR$ and ASC functions to aid in the conversion process from binary to ASCII. Luckily, the conversion in the other direction—from ASCII to binary—is not too difficult to perform in BASIC without support functions.

The OSI BASIC, as do most others, uses "tokens" to allow the compression of the BASIC source into a smaller package in memory. The tokens are simply single-byte flags with values in the range of decimal 128–255, beyond the range of valid ASCII codes, which are used to take the place of the BASIC command verbs.

Whenever the BASIC scanner finds a string of characters it recognizes as a keyword, such as GOTO, it replaces that character string with the single-byte token that corresponds to that keyword. The renumber routine must thus scan for the tokens requiring renumbering and alter the statement numbers that follow them. In the OSI version of Microsoft BASIC, the tokens we need to look for are decimal 136 (GOTO), 137 (RUN), 140 (GOSUB) and 160 (THEN).

The renumber routine is organized into two parts. The first part is the "simpleminded" renumberer I described earlier, with one additional function. While it is inserting the new statement numbers, it also must save the old statement numbers in a chunk of RAM so the second pass will know how to renumber the internals of the statements. In OSI systems with video boards, one of the most convenient chunks of RAM is the video display memory, which begins at 53248 decimal. Each statement number saved uses two bytes, and two bytes are required for an end-of-table flag. Hence in the C1-P machines with 1024 bytes of video RAM, you can renumber a program with as many as 511 statements. In the C2-4P you can handle 1023 statements with its 2K of video RAM.

The second part of the renumberer goes back and looks at the text in the state-

With these disks, I can turn your TRS-80 into a serious computer.

I'm Irwin Taranto, and I've put the first computer into more than 300 different businesses.

It's taught me that the TRS-80 is an elegant piece of hardware despite its low price. Given the right programs, it can jump through hoops.

Put simply, I have the right programs. Four of them are the genuine Osborne & Associates systems, originally designed for the $30,000 Wang computer. I've made a few minor modifications, and they now work on a $4000 TRS-80. The other two programs I added myself.

These programs are fully-documented, and you can buy the books locally or from me. I made them work on the TRS-80, and if you buy them from me, I'll make them work for you.

If you're not sure about that, call the number below and get the names of some of the people who've bought all over the world. Then ask them.

These programs cost $99.95 each. (The Cash Journal option on the General Ledger adds another $50.) That gets you the disk, all the instructions you need and my phone number. If you call, we answer all your questions. If your question's tough enough, I'll talk to you personally.

Because I plan to turn that TRS-80 of yours into a serious computer.

THE OSBORNE PROGRAMS

Accounts Payable: invoice-linked, it reports, does checks and links to general ledger.
Accounts Receivable: invoice-linked, it tracks invoices and aging, prints statement and links to general ledger.
General Ledger: handles 1750 transactions each on 200 different accounts. Cash journal option available.
Payroll: figures the pay, does the checks and all the bookkeeping.

AND MY OWN PROGRAMS

Inventory Control: gives immediate readout on any inquiry. It has many existing versions or can be individually tailored.
NEW! Invoicing: linked to accounts receivable. Prints invoices and feeds data into receivables.

✓T57

Taranto
& ASSOCIATES

P.O. Box 6073, 4136 Redwood Highway, San Rafael CA 94903 · (415) 472-1415. Add $3 per order for handling, 6% sales tax in California only. If you don't already have the books, add $15 each (invoicing book, $10). Mastercharge, Visa OK.

STOCK
CONTINUOUS
CHECKS & STATEMENTS

You can take advantage of the most complete line of stock, off the shelf continuous checks available, with more formats added monthly. We have an economical check that will meet your needs. You can actually save up to 60% on 1,000 checks. Delivery in either 1½ or 3 weeks.

SEND FOR YOUR FREE SAMPLES TODAY!

Are you getting this service on your check format? Send us a copy of the check your system uses, and we will be happy to consider adding it to our stock line.

Checks To-Go

✓C172

P.O. Box 148, Spring Valley, California 92077 (714) 460-4975

FREE

You'll save money, have fun, and learn by building it yourself — with easy-to-assemble Heathkit Computers. See all the newest in home computers, video terminals, floppy disk systems, printers and innovative software.

Send today for your FREE Heathkit® Catalog

If coupon is missing, write
Heath Co., Dept. 351-612,
Benton Harbor, MI 49022 ✓H49

Send to: Heath Co., Dept. 351-612,
Benton Harbor, MI 49022.

Send my free Heathkit Catalog now. I am not currently receiving your catalog.

Name ____________________

Address ____________________

____

City ____________ State ________

CL-728 Zip ________

ments, looking for the four tokens noted earlier. When it finds one of them, it looks behind it to see if there is a statement number. If the routine finds a statement number 1, it converts it from ASCII to binary and then compares it against the statement numbers that the first pass saved in the video RAM.

At this point one of two things can happen to the renumber program. The first is that it finds the old statement number in the table. If this occurs all is OK, and we proceed normally. The alternative is that the routine can't find the old statement number, in which case there was an error in your original source program, such as a GOTO with a missing destination.

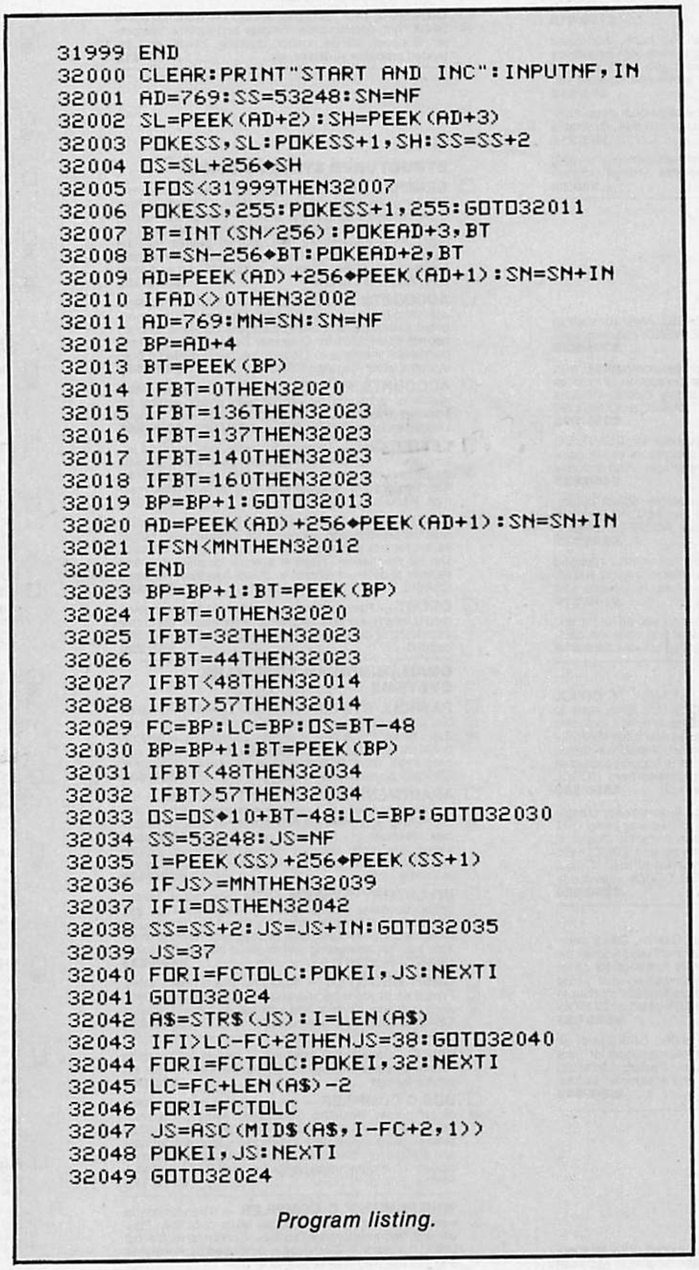
Improvisations

But at this point your old program is partially renumbered, and we can't just stop renumbering. So to recover, I chose to insert percent signs (%) where the missing statement number was, to indicate in the renumbered listing that something went wrong during renumbering. It would have been nice to print an error message at this point, but doing so would have disturbed the video RAM where the old statement numbers were stored. I discovered this the hard way after much head scratching!

If the program successfully found the old statement number in the video RAM, it must now insert the corresponding new statement number in the BASIC text in place of the old number. Here is where the STR$ and ASC functions of BASIC come in to play. One minor quirk that must be addressed here is that the STR$ function returns a leading blank in the character string, probably where a sign would go, and this blank should be skipped over when POKEing back the ASCII characters.

At this point we run into another possible error condition. What happens if the new statement number has more digits than the original statement number and, hence, won't fit over top of it? Again, I chose to overlay the old statement number with a special character, in this case the ampersand (&), to flag the error and distinguish this type of error from the "old statement not found" condition noted before.

A few other minor changes are required to make this renumbering technique work. Most important is to make sure that the program doing the renumbering does not try to renumber itself. Strange and undesirable things can happen if a program attempts to dynamically renumber itself. To prevent this from occurring, the renumber program checks for statement numbers greater than 31998 and leaves them alone. Since the renumberer starts at statement 31999, it will remain intact.



```
31999 END
32000 CLEAR:PRINT"START AND INC":INPUTNF,IN
32001 AD=769:SS=53248:SN=NF
32002 SL=PEEK(AD+2):SH=PEEK(AD+3)
32003 POKESS,SL:POKESS+1,SH:SS=SS+2
32004 OS=SL+256◆SH
32005 IFOS<31999THEN32007
32006 POKESS,255:POKESS+1,255:GOTO32011
32007 BT=INT(SN/256):POKEAD+3,BT
32008 BT=SN-256◆BT:POKEAD+2,BT
32009 AD=PEEK(AD)+256◆PEEK(AD+1):SN=SN+IN
32010 IFAD<>0THEN32002
32011 AD=769:MN=SN:SN=NF
32012 BP=AD+4
32013 BT=PEEK(BP)
32014 IFBT=0THEN32020
32015 IFBT=136THEN32023
32016 IFBT=137THEN32023
32017 IFBT=140THEN32023
32018 IFBT=160THEN32023
32019 BP=BP+1:GOTO32013
32020 AD=PEEK(AD)+256◆PEEK(AD+1):SN=SN+IN
32021 IFSN<MNTHEN32012
32022 END
32023 BP=BP+1:BT=PEEK(BP)
32024 IFBT=0THEN32020
32025 IFBT=32THEN32023
32026 IFBT=44THEN32023
32027 IFBT<48THEN32014
32028 IFBT>57THEN32014
32029 FC=BP:LC=BP:OS=BT-48
32030 BP=BP+1:BT=PEEK(BP)
32031 IFBT<48THEN32034
32032 IFBT>57THEN32034
32033 OS=OS◆10+BT-48:LC=BP:GOTO32030
32034 SS=53248:JS=NF
32035 I=PEEK(SS)+256◆PEEK(SS+1)
32036 IFJS>=MNTHEN32039
32037 IFI=OSTHEN32042
32038 SS=SS+2:JS=JS+IN:GOTO32035
32039 JS=37
32040 FORI=FCTOLC:POKEI,JS:NEXTI
32041 GOTO32024
32042 A$=STR$(JS):I=LEN(A$)
32043 IFI>LC-FC+2THENJS=38:GOTO32040
32044 FORI=FCTOLC:POKEI,32:NEXTI
32045 LC=FC+LEN(A$)-2
32046 FORI=FCTOLC
32047 JS=ASC(MID$(A$,I-FC+2,1))
32048 POKEI,JS:NEXTI
32049 GOTO32024
```

Program listing.

Operation

The procedure to use the renumber program is relatively simple. First, load in the renumber program, which starts at statement 31999. Actually, the first executable statement is at 32000; the END at 31999 is inserted to stop a user program that terminates by falling through to the end of the program without an explicit END statement. After loading in the renumbering program, load or key in the program you wish to renumber. It is assumed that this program will not have statement numbers greater than 31998.

After loading is complete, key in RUN 32000 to begin renumbering. You will be prompted for the desired beginning new statement number and increment value. After this, the only visible evidence that renumbering is in process is that some apparently meaningless characters will appear at the top portion of your video monitor during the first renumbering pass: These are the old statement numbers being saved in the video RAM. These may not be visible if you are renumbering a short program on a C1-P system, due to video overscan.

After this there will be a relatively long pause, possibly several minutes, depending on the size of the program being renumbered. Be patient; do not press control-C or BREAK during this period or the program being renumbered will be left only partially renumbered, since the video RAM will be disturbed. When renumbering is completed, BASIC will prompt you with an OK, and you can proceed to list and save your renumbered program. To save or list just your renumbered program and not the renumbering code, key in LIST 1-31998, and any statements in your program will be listed.

The renumberer can be a valuable tool during program development by allowing dynamic renumbering while you are in the process of coding and testing a new program. It gives the added benefit of checking for missing destinations on GOTOs and GOSUBs that might otherwise go undetected until an unusual condition arose in program execution.

The renumberer does not affect the execution of the user program while coexisting with it in the machine, other than by occupying memory that would otherwise be available for variables. The program statements for the renumberer occupy just under 1K bytes, and the requirement for variables during execution will bring the storage re quirement up somewhat beyon that. ■

TRS-80 MODEL II FORMAT NOW AVAILABLE

DIGITAL RESEARCH

Software with Manual / Manual Alone

□ Ⓜ **CP/M* FLOPPY DISKETTE OPERATING SYSTEM** — Packages supplied on diskette complete with 8080 assembler, text editor, 8080 debugger and various utilities plus full documentation. CP/M available configured for most popular computer/disk systems including: North Star Single, Double or Quad density, Altair 8" disks, Helios II, Exidy Sorcerer, Vector MZ, Heath H17† or H89†, TRS-80†, iCOM 3712 and iCOM Micro Disk plus many other configurations available off the shelf. .... **$145/$25**

CP/M version 2 (not all formats available immediately) .... **$170/$25**

□ **MAC** — 8080 Macro Assembler. Full Intel macro definitions. Pseudo Ops include RPC, IRP, REPT, TITLE, PAGE, and MACLIB. Z-80 library included. Produces Intel absolute hex output plus symbols file for use by SID (see below) .... **$100/$15**

□ **SID** — 8080 symbolic debugger. Full trace, pass count and break-point program testing system with back-trace and histogram utilities. When used with MAC, provides full symbolic display of memory labels and equated values .... **$85/$15**

□ **TEX** — Text formatter to create paginated, page-numbered and justified copy from source text files, directable to disk or printer .... **$85/$15**

□ **DESPOOL** — Program to permit simultaneous printing of data from disk while user executes another program from the console .... **$50/$5**

All Microsoft prices are discounted!

MICROSOFT

□ Ⓛ Ⓜ **BASIC-80** — Disk Extended BASIC, ANSI compatible with long variable names, WHILE/WEND, chaining, variable length file records .... **$300/$25**

□ Ⓛ Ⓜ **BASIC COMPILER** — Language compatible with BASIC-80 and 3-10 times faster execution. Produces standard Microsoft relocatable binary output. Includes Macro-80. Also linkable to FORTRAN-80 or COBOL-80 code modules .... **$350/$25**

□ Ⓛ Ⓜ **FORTRAN-80** — ANSI '66 (except for COMPLEX) plus many extensions. Includes relocatable object complier, linking loader, library with manager. Also includes MACRO-80 (see below) .... **$400/$25**

□ Ⓛ Ⓜ **COBOL-80** — ANSI '74 Relocatable object output. Format same as FORTRAN-80 and MACRO-80 modules. Complete ISAM, interactive ACCEPT/DISPLAY, COPY, EXTEND .... **$625/$25**

□ Ⓛ Ⓜ **MACRO-80** — 8080/Z80 Macro Assembler. Intel and Zilog mnemonics supported. Relocatable linkable output. Loader, Library Manager and Cross Reference List utilities included .... **$149/$15**

□ Ⓛ **EDIT-80** — Very fast random access text editor for text with or without line numbers. Global and intra-line commands supported. File compare utility included **$89/$15**

MICRO FOCUS

□ Ⓛ **STANDARD CIS COBOL** — ANSI '74 COBOL standard compiler fully validated by U.S. Navy tests to ANSI level 1. Supports many features to level 2 including dynamic loading of COBOL modules and a full ISAM file facility. Also, program segmentation, interactive debug and powerful interactive extensions to support protected and unprotected CRT screen formatting from COBOL programs used with any dumb terminal ..... **$850/$50**

□ Ⓛ **FORMS 2** — CRT screen editor. Automatically creates a query and update program of indexed files using CRT protected and unprotected screen formats. Output is COBOL data descriptions for copying into CIS COBOL programs. No programming experience needed. Output program directly compiled by CIS COBOL (standard). .... **$200/$20**

EIDOS SYSTEMS

□ Ⓛ **KISS** — Keyed Index Sequential Search. Offers complete Multi-Keyed Index Sequential and Direct Access file management. Includes built-in utility functions for 16 or 32 bit arithmetic, string/integer conversion and string compare. Delivered as a relocatable linkable module in Microsoft format for use with FORTRAN-80 or COBOL-80, etc. .... **$535/$23**

□ Ⓛ **KBASIC** — Microsoft Disk Extended BASIC with all KISS facilities, integrated by implementation of nine additional commands in language. Package includes KISS.REL as described above, and a sample mail list program .... **$995/$45**

All Micropro prices are discounted!

MICROPRO

□ Ⓛ **SUPER-SORT I** — Sort, merge, extract utility as absolute executable program or linkable module in Microsoft format. Sorts fixed or variable records with data in binary, BCD, Packed Decimal, EBCDIC, ASCII, floating, fixed point, exponential, field justified, etc. etc. Even variable number of fields per record! .... **$225/$25**

□ Ⓛ **SUPER-SORT II** — Above available as absolute program only .... **$175/$25**

□ Ⓛ **SUPER-SORT III** — As II without SELECT/EXCLUDE .... **$125/$25**

Software with Manual / Manual Alone

□ Ⓛ **WORD-STAR** — Menu driven visual word processing system for use with standard terminals. Text formatting performed on screen. Facilities for text paginate, page number, justify, center and underscore. User can print one document while simultaneously editing a second. Edit facilities include global search and replace, read/write to other text files, block move, etc. Requires CRT terminal with addressable cursor positioning **$445/$25**

□ Ⓛ **WORD-MASTER** Text Editor — In one mode has superset of CP/M's ED commands including global searching and replacing, forward and backwards in file. In video mode, provides full screen editor for users with serial addressable-cursor terminal .... **$125/$25**

SOFTWARE SYSTEMS

□ Ⓜ **CBASIC-2** Disk Extended BASIC — Non-interactive BASIC with pseudo-code compiler and runtime interpreter. Supports full file control, chaining, integer and extended precision variables, etc. .... **$109/$15**

Structured Systems prices are discounted!

STRUCTURED SYSTEMS GROUP

□ **GENERAL LEDGER** — Interactive and flexible system providing proof and report outputs. Customization of COA created interactively. Multiple branch accounting centers. Extensive checking performed at data entry for proof, COA correctness, etc. Journal entries may be batched prior to posting. Closing procedure automatically backs up input files. All reports can be tailored as necessary. Requires CBASIC .... **$899/$25**

□ **ACCOUNTS RECEIVABLE** — Open item system with output for internal aged reports and customer-oriented statement and billing purposes. On-Line Enquiry permits information for Customer Service and Credit departments. Interface to General Ledger provided if both systems used. Requires CBASIC .... **$699/$25**

□ **ACCOUNTS PAYABLE** — Provides aged statements of accounts by vendor with check writing for selected invoices. Can be used alone or with General Ledger and/or with NAD. Requires CBASIC .**$699/$25**

□ **LETTERIGHT** — Program to create, edit and type letters or other documents. Has facilities to enter, display, delete and move text, with good video screen presentation. Designed to integrate with NAD for form letter mailings. Requires CBASIC .... **$179/$25**

□ **NAD** Name and Address selection system — interactive mail list creation and maintenance program with output as full reports with reference data or restricted information for mail labels. Transfer system for extraction and transfer of selected records to create new files. Requires CBASIC .... **$79/$20**

□ **QSORT** — Fast sort/merge program for files with fixed record length, variable field length information. Up to five ascending or descending keys. Full back-up of input files created. .... **$95/$20**

GRAHAM-DORIAN SOFTWARE SYSTEMS

□ Ⓛ Ⓜ **PAYROLL SYSTEM** — Maintains employee master file. Computes payroll withholding for FICA, Federal and State taxes. Prints payroll register, checks, quarterly reports and W-2 forms. Can generate ad hoc reports and employee form letters with mail labels. Requires CBASIC. Supplied in source code. .... **$590/$35**

□ Ⓛ Ⓜ **APARTMENT MANAGEMENT SYSTEM** — Financial management system for receipts and security deposits of apartment projects. Captures data on vacancies, revenues, etc. for annual trend analysis. Daily report shows late rents, vacancy notices, vacancies, income lost through vacancies, etc. Requires CBASIC. Supplied in source code. .... **$590/$35**

□ Ⓛ Ⓜ **INVENTORY SYSTEM** — Captures stock levels, costs, sources, sales, ages, turnover, markup, etc. Transaction information may be entered for reporting by salesman, type of sale, date of sale, etc. Reports available both for accounting and decision making. Requires CBASIC. Supplied in source code. .... **$590/$35**

□ Ⓛ Ⓜ **CASH REGISTER** — Maintains files on daily sales. Files data by sales person and item. Tracks sales, overrings, refunds, payouts and total net deposits. Requires CBASIC. Supplied in source code .... **$590/$35**

□ **tiny C** — Interactive interpretive system for teaching structured programming techniques. Manual includes full source listings .... **$75/$40**

□ Ⓜ **BDS C COMPILER** — Supports most major features of language, including Structures, Arrays, Pointers, recursive function evaluation, linkable with library to 8080 binary output. Lacks data initialization, long & float type and static & register class specifiers. Documentation includes "C" Programming Language book by Kernighan & Ritchie .... **$110/$15**

□ Ⓛ **WHITESMITHS' C COMPILER** — The ultimate in systems software tools. Produces faster code than Pascal with more extensive facilities. Conforms to the full UNIX*** Version 7 C language, described by Kernighan and Ritchie, and makes available over 75 functions for performing I/O, string manipulation and storage allocation. Compiler output in A-Natural source. Supplied with A-Natural (see below) requires 60K CP/M . . **$630/$30**

□ Ⓛ **A-NATURAL** — Narrative assembler with linking loader, librarian, extensive 8080 subroutine library in A-Natural relocatable format and translators from A-Natural source to Microsoft MACRO-80 source and from A-Natural rel to source .... **$330/$15**

Software with Manual / Manual Alone

□ Ⓜ **POLYVUE/80** — Full screen editor for any CRT with XY cursor positioning. Includes vertical and horizontal scrolling, interactive search and replace, automatic text wrap around for word processing, operations for manipulating blocks of text, and comprehensive 70 page manual. .... **$135/$15**

□ Ⓜ **POLYTEXT/80** — Text formatter for word processing applications. Justifies and paginates source text files. Will generate form letters with custom fields and conditional processing. Support for Daisey Wheel printers includes variable pitch justification and motion optimization. .... **$85/$15**

□ Ⓜ **ALGOL-60** — Powerful block-structured language compiler featuring economical run time dynamic allocation of memory. Very compact (24K total RAM) system implementing almost all Algol 60 report features plus many powerful extensions including string handling direct disk address I/O etc. Requires Z80 CPU .... **$199/$20**

□ Ⓜ **Z80 DEVELOPMENT PACKAGE** — Consists of: (1) disk file line editor, with global inter and intra-line facilities; (2) Z80 relocating assembler, Zilog/Mostek mnemonics, conditional assembly and cross reference table capabilities; (3) linking loader producing absolute Intel hex disk file. .... **$95/$20**

□ Ⓜ **ZDT** — Z80 Debugger to trace, break and examine registers with standard Zilog/Mostek mnemonic disassembly displays. $35 when ordered with Z80 Development Package .... **$50/$10**

□ **DISTEL** — Disk based disassembler to Intel 8080 or TDL/Xitan Z80 source code, listing and cross reference files. Intel or TDL/Xitan pseudo ops optional. Runs on 8080. .... **$65/$10**

□ Ⓜ **DISILOG** — As DISTEL to Zilog Mostek mnemonic files. Runs on Z80 only .... **$65/$10**

□ Ⓜ **TEXTWRITER III** — Text formatter to justify and paginate letters and other documents. Special features include insertion of text during execution from other disk files or console, permitting recipe documents to be created from linked fragments on other files. Has facilities for sorted index, table of contents and footnote insertions. Ideal for contracts, manuals, etc. .... **$125/$20**

□ Ⓜ **POSTMASTER** — A comprehensive package for mail list maintenance. Features include keyed record extraction and label production. A form letter program is included which provides neat letters on single sheet or continuous forms. Requires CBASIC .... **$150/$25**

□ **WHATSIT?****** Interactive data-base system using associative tags to retrieve information by subject. Hashing and random access used for fast response. Requires CBASIC .... **$125/$25**

□ **XYBASIC** Interactive Process Control BASIC — Full disk BASIC features plus unique commands to handle bytes, rotate and shift, and to test and set bits. Available in Integer, Extended and ROMable versions.
Integer Disk or Integer ROMable .... **$295/$25**
Extended Disk or Extended ROMable ..... **$395/$25**

□ **SMAL/80** Structured Macro Assembled Language — Package of powerful general purpose text macro processor and SMAL structured language compiler. SMAL is an assembler language with IF-THEN-ELSE, LOOP-REPEAT-WHILE, DO-END, BEGIN-END constructs .... **$75/$15**

□ Ⓜ **SELECTOR III-C2** — Data Base Processor to create and maintain multi Key data bases. Prints formatted, sorted reports with numerical summaries or mailing labels. Comes with sample applications including Sales Activity, Inventory, Payables, Receivables, Check Register, and Client/Patient Appointments, etc. Requires CBASIC Version 2. Supplied in source code. **$345/$20**

□ **CPM/374X** — Has full range of functions to create or re-name an IBM 3741 volume, display directory information and edit the data set contents. Provides full file transfer facilities between 3741 volume data sets and CP/M files .... **$195/$10**

□ Ⓜ **BASIC UTILITY DISK** — Consists of: (1) CRUNCH-14 – Compacting utility to reduce the size and increase the speed of programs in Microsoft Basic and TRS-80 Basic. (2) DPFUN – Double precision subroutines for computing nineteen transcendental functions including square root, natural log, log base 10, sin, arc sin, hyperbolic sin, hyperbolic arc sin, etc. Furnished in source on diskette and documentation .... **$50/$35**

□ Ⓜ **THE STRING BIT** — Fortran character string handling. Routines to find, fill, pack, move, separate, concatenate and compare character strings. This package completely eliminates the problems associated with character string handling in FORTRAN. Supplied with source .... **$45/$15**

□ Ⓜ **BSTAM** — Utility to link one computer to another also equipped with BSTAM. Allows file transfers at full data speed (no conversion to hex), with CRC block control check for very reliable error detection and automatic retry. We use it! It's great! Full wildcard expansions to send *.ASM, etc. 9600 baud with wire, 300 baud with phone connection. **Both** ends need one. Standard and M versions can talk to one another .... **$150/$5**

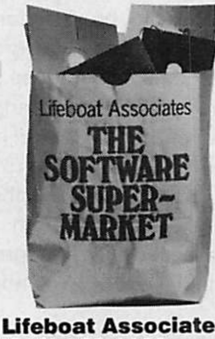
□ **Flippy Disk Kit** — Template and instructions to modify single sided 5¼" diskettes for use of second side in singled sided drives **$12.50**

*CP/M is a trademark of Digital Research.
**Z80 is a trademark of Zilog, Inc.
***UNIX is a trademark of Bell Laboratories.
****WHATSIT? is a trademark of Computer Headware.

†CP/M for Heath, TRS-80 Model I and PolyMorphic 8813 are modified and must use specially compiled versions of system and applications software.

Ⓜ Modified version available for use with CP/M as implemented on Heath and TRS-80 Model I computors.

Ⓛ User license agreement for this product must be signed and returned to Lifeboat Associates before shipment may be made.



Orders must specify disk systems and formats: e.g. North Star single, double or quad density, IBM single or 2D/256, Altair, Helios II, Micropolis Mod I or II, 5¼" soft sector (Micro iCOM/SD Systems Dynabyte), etc.

Prices F.O.B. New York. Shipping, handling and C.O.D. charges extra.

Manual cost applicable against price of subsequent software purchase.

The sale of each propriety software package conveys a license for use on one system only.

Shopping List No.7

Software for most popular 8080/Z80 computer disk systems including NORTH STAR, iCOM, MICROPOLIS, DYNABYTE DB8/2, EXIDY SORCERER, SD SYSTEMS, ALTAIR, VECTOR MZ, MECCA, 8" IBM, HEATH H17 & H89, HELIOS, IMSAI VDP42 & 44, REX, INTERTEC, VISTA V80 and V200, TRS-80 MODEL I and MODEL II, OHIO SCIENTIFIC and IMS 5000 formats.

™The Software Supermarket is a trademark of Lifeboat Associates

Lifeboat Associates, 2248 Broadway, N.Y., N.Y. 10024
(212) 580-0082 Telex: 668585