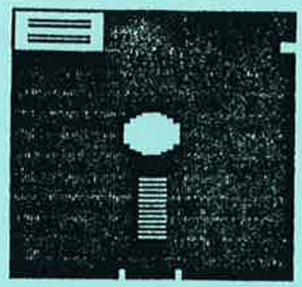
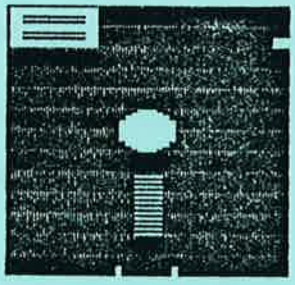


NIBBLES & BITS



Published by DIGITAL EXPRESS, Incorporated
Route one, Box 29 - 6
Oak Hill, WV 25901

THE COMPREHENSIVE MONTHLY NEWSLETTER FOR ADAM™ USERS

October 1986
vol: 1, nmb: 4
SINGLE ISSUE: $3.50

This issue includes 11 SmartBASIC program LISTs, 4 tables (charts), and 3 assembly language lists.

THE N&B STAFF

FOUNDER:
Vernon L. Whitman, Sr.
EDITOR-IN-CHIEF:
Dr. Solomon Swift
DESIGN DIRECTOR:
Tim Whetstine
TECHNICAL DIRECTOR:
Chris Davidson
CIRCULATION DIRECTOR:
Tony Michaels
CONTRIBUTING EDITORS:
Janet Weston
Ted Johnson
Cindy Harrington

PUBLIC NOTICE

NIBBLES & BITS is published monthly by DIGITAL EXPRESS, INC. Individual issues may be purchased for the current issue or a back issue (premier issue was July, 1986) for $3.50. The standard subscription rate for one year (12 issues) is $18.00 in the USA, its possessions, and Canada and $24.00 in other foreign countries. The standard subscription rate for six months is $12.00 in the USA, its possessions, and Canada and $16.00 in other foreign countries.

We welcome contributions of original reviews, programs, articles, questions, and comments. Please include your subscription ID number from your mailing label on all written correspondence to us.

Your subscription ID number is on the first line of your mailing label (affixed to the newsletter). It is a 10 digit code. The first four digits are the month and year of the final issue in your current subscription. Following the ID number is a brief message. If this is your final issue, the message will read "FINAL ISSUE!!!". If this is your penultimate issue, the message will read "TIME TO RENEW". Otherwise the message will apprise you of the exact number of issues remaining in your subscription (excluding the current issue). Please verify this information each month.

To insure that you don't miss an issue, please renew early and let us know promptly of any address change.

DISCLAIMER

The editor(s) and publisher have exercised due care in the preparation of this newsletter. Neither the N&B staff, nor DEI, nor any contributors of any capacity make any warranty either expressed or implied with regard to the information contained herein either by interpretation, use, or misuse. Reviews and opinions submitted by the readership at large do not necessarily reflect the opinions of the editor or staff. DEI has no affiliation with Coleco Industries, Inc.

EDITOR'S NOTE

The mail just keeps pouring in. As the new editor, I'd like to thank all of you for your support of our efforts. It is active ADAM users who keep newsletter editors writing, hardware manufacturers designing, and software developers creating. In short, **YOU KEEP ADAM ALIVE.**

Active ADAM users are some of the most devoted hobbyists for any personal computer. Our ADAM may be orphaned, but it is FAR FROM DEAD.

Indeed, ADAM will live for MANY years to come!!! And, to insure customer satisfaction, we have made a few changes at DEI. As you may have noticed, we've nearly doubled the information in each issue of <u>NIBBLES & BITS</u> -- this began with the September issue.

Another change is the address of our central base of operations (notice the letterhead on page one). This was not an easy decission, but it was necessary for *super fast* order processing. We now ship 99% of all orders (and answer mail) from subscribers in 24 hours or less, provided they are mailed to the new address. Mail sent to the original address may take a day or two longer for response.

We are dedicated to accomplishing our share in keeping this marvelous computer alive. Thank you, once again, for your support.

LONG LIVE ADAM!!!

Dr. Solomon Swift
EDITOR-IN-CHIEF

N&B NEWS

→We would like to thank Wayne Motel of NIAD for his nice review of Intel-BEST 3.3 in their August issue -- rated 'A+'. Also, in that same issue (of NIAD) Mr. Motel has an easy-to-understand assembly language list of the SmartBASIC V1.0 bootstrap (block 0) routine. This is an article that many of you hackers may be interested in.

→Due to popular demand, DEI now offers most software on disk. This includes the public domain libraries.

→We have expanded the BULLETIN BOARD section. Entries up to five lines are now accepted. The line length limit is now 50 characters at normal size and 25 characters at double width.

→We apologize for the belated September and October issues. With the November issue, we'll be back on schedule.

→One of the disadvantages of using SmartBASIC V2.0 is that nearly all of the common POKE addresses have been changed. Because this trade-off will also be a problem with Dr. Swift's BASIC, we've postponed its completion. Instead, we're adding a new section to the HACKER'S DELIGHT department, 'PatchWORK'. Here, we'll take a detailed look at SmartBASIC and develop enhancements with elaborate explanations. Each 'patch' will have both a SmartBASIC V1.0 version and an Intel-BEST 3.3 version. This month we've listed a trick that insures *absolutely true randomization*.

→Please notice the SWIFT POLL this month. We have our first three month tally -- with last month's incentive announcement, more than 25% of you have responded. We also have our first winner of the BALLOT drawing. The lucky ADAMite is . . .

Raymond Tremor of Honolulu, Hawaii

→We did not have a winner for HACKER'S CONTEST #2. However, we've had several correct responses for the third contest. We'll have to have a drawing. The winner will be announced next month. Also, to allow more time for responses, we're chnaging the HACKER'S CONTEST to a bi-monthly competiion.

ADAM NEWS

→Most of the department stores mentioned in the premier issue still carry ADAM computers and some Coleco software. The average selling price is now $199.95.

→E & T software has revised their Softpack I and Business Pack programs. To get an updated version, send your original DDP or disk along with $2.50 (for shipping) to:

E & T Software
P.O. Box 821242
Dallas, TX 75382-1242

→Vinh Le has developed a graphics screen dump for most popular dot-matrix printers. For more info, write to:

Vinh Le
9150 Todos Santos
Santee, CA 92071

→Data Backup has developed an improvement to their ADAM tractor feed attachment which prevents it from working loose. For more info, write to:

Data Backup
P.O. Box 335
Iona, ID 83427

→Marathon Computer Press is offering a 30% discount on their own software to NIBBLES & BITS subscribers. See our review, in this issue, of their "THE SPANISH VOCABULARIAN". Be sure to include the following information on your order form.

1. MCP discount control number:
NIBB92286MC

2. Your N&B subscription ID number:
(from your mailing label)

To get their new, expanded Fall catalog, write to:

MARATHON COMPUTER PRESS
P.O. Box 68503
Virginia Beach, VA 23455

The discount applies to the following five software packages. The listed price is ALREADY discounted and is the same for DDP or disk versions.

CODEVISOR 4.1:	$15.05
THE INVESTMENT ANALYST:	12.25
THE SPANISH VOCABULARIAN:	12.95
CopyWriter 1.0:	11.02
MEGAUTIL:	23.06

→"THE HACKER'S GUIDE TO ADAM: VOLUME TWO" and the "ADAM Resource DIRECTORY" are two outstanding books. Both are reviewed in this issue. If you don't already have them, we highly recommend that you consider purchasing both.

ENTERING PROGRAMS

We usually include several BASIC programs in every issue. We try to keep them short so that they are easy to enter. However, keying in a program can sometimes be a frustrating endeavor. Here are a few tips which may facilitate the process.

1. Enter NEW before you start typing. This clears RAM for a new program.

2. Always SAVE the program before RUNning it. This way, if the program does crash, your efforts won't be lost.

3. If you encounter problems, print a hardcopy LIST and compare it with the newsletter LISTing.

4. Ninety-nine percent fo the time, operational problems are caused by simple typos.

5. Pay particular attention to the numbers one and zero. Don't confuse them with the letter keys (lower case "L" and the upper case "O").

6. If you're a beginner, be EXTREMELY CAREFUL entering programs from the HACKER'S DELIGHT department. All of these programs include machine language. Even one incorrect keypress could lock up ADAM or, worse yet, ERASE a datapack or disk.

EXPANDING YOUR SYSTEM

MODEMS

Modem is an acronym for MOdulator/DEModulator. It's a device that links your computer, over telephone lines, to other computers (micros or mainframes).

Coleco manufactured a modem specifically for ADAM, the ADAMlink. It plugs into slot #1, ie, the leftmost (of the three) interface under the cover on the Memory Console. Coleco provided software to operate this modem, ADAMlink I. This program allows you communicate with other computers (even other ADAMs) in a user friendly environment.

The primary drawback of the software is that is does not allow for uploading and downloading files, ie, transferring data to/from disk or DDP via the modem. Coleco used to offer an update to the program, ADAMlink II, which corrected this shortcoming (for $9.00). NIAD now offers this update as a public domain volume. You should note that both programs only permit SmartWriter compatible files to be transferred.

With CP/M and an RS-232 interface (available from several hardware developers), you can use almost any popular modem. Most of the modem software that utilize the CP/M operating system permit the transfer of any file type. However, many of these programs are complex and can take several hours to learn to use.

The speed of data transfer, over the telephone line, of a modem is referred to as "baud rate". Unlike most computer terminology, "baud" is not an acronym. Rather, the term is a contraction of the surname of the Frenchman J.M. E. Baudot, whose five-bit code was adopted by the French telegraph system in 1877. Although baud rate and "bits per second" are commonly used synonymously, actual baud rate is generally greater than the number of bits being transferred each second.

The ADAMlink modem is designed for 300 baud transfers. For many years this rate has been a standard. However, 1200 and 2400 baud are becoming much more popular.

There are two general types of modems, "acoustic couplers" and "hard-wired modems". Acoustic couplers, which are used infrequently today, allow you to place the phone atop the device so that tones are sent through the mouthpiece. Hard-wired modems, like ADAMlink, connect directly to the phone line.

So . . . why can't you connect the phone line directly to ADAM? This is where the MOdulation and DEModulation comes in.

The human voice, with its vast range of tones, is transmitted over phone lines using continuously variable signals. Thus, the normal telephone transmission is in analog form. Computers, however, work with digital data. The modem is necessary in order to convert to and from computer compatible digital format and telephone line compatible analog format.

ADAM USERS FORUM

The following questions and comments were culled from recently received mail. Generally, both the reader's input and our response are excerpted from the actual correspondence.

FAN-FOLD PAPER TIP

Here's a very inexpensive way to keep fan-fold paper in line (on the ADAM printer) without having a tractor feed. Purchase two small coaxial, plastic, double-backed tape guides (the kind you use for VCR or cable TV to put it neatly along a baseboard). Put them on your printer in line with the paper alignment guides. Place one on each side and your paper feeds straight. The cost . . . about twenty cents!

Walt Wright
490 17th Street
West Babylon, Long Island
New York 11704

EDITOR'S NOTE: Thank you, Walt, for the tip. I'm sure many ADAMites will find this to be a very useful tip. We'd love to hear from any of you who have helpful hints and/or interesting discoveries to share. Who knows . . . hundreds of ADAMites may benefit from your submission (and some of them may even write to you). Software tips are welcomed, as well.

PARALLEL PRINTER PROBLEM

I believe that I have discovered a new problem with the ADAM. I thought that you may have already solved or be interested in solving it. I purchased a printer interface and the Star NX-10 graphics printer. This printer is truly a class act. It has the capability (in BASIC) of printing graphics, right justified printing, centering, reverse line feed, and many, many other special functions. ADAM doesn't respond correctly when I enter the proper codes which enable some of the special printer functions.

Michael Bogrees
308 East Wenger Road
Englewood, OH 45322

IN RESPONSE: Have you tried the Intel-BEST 3.3 'PR#2' command? The problem is not with the printer or ADAM circuitry. The 'patched' routines with allow you to use the 'PR#1' command with the printer do so by modifying the EOS (Elementary Operating System). Some ASCII values are automatically winnowed out in order to insure compatibility with SmartWriter, SmartFiler, etc. However, with SmartBASIC this EOS modification can be restrictive. A 'PR#2' command (which doesn't alter the EOS), on the other hand, provides a viable alternative.

SPECIAL NOTE: In a later issue, we'll show you how to create a 'PR#2' command and LIST a BASIC word processor which makes use of the special printer functions (which, by the way, are built-in features on most newer dot-matrix printers).

MERGING BASIC PROGRAMS

In your July issue, you mentioned a trick for merging programs directly from BASIC. You modified the NEW command. I was wondering if it would not be easier to modify the LOAD command instead.

George A. Havach
550 - 27th Street, #202
San Francisco, CA 94131

EDITOR'S NOTE: Mr. Havach wrote this letter before he read the "HACKER'S GUIDE TO ADAM: VOLUME TWO". Among an abundance of other valuable tips, the Hinkle's describe the LOAD modification. See our "POKES TO PLAY WITH" section in this issue.

SMARTWRITER BUGS

Mr. White (address below), a relatively new ADAMite, wrote to us concerning some bugs (glitches) with SmartWriter. To abreviate, we've combined both his inquiry and our response into a short list that may be of help to other SmartWriter users.

Harlow H. White
97 Sunset Drive, #303A
Sarasota, FL 33577

SPECIAL FORMAT: SmartWriter does have its limitations and, indeed, it does have a few bugs. The following list itemizes of few of these glitches.

1. The sound going out is a common problem. It is generally caused by using the 'CLEAR' or 'DELETE' functions. To avoid, press these keys slowly.

2. If you press the 'CLEAR' workspace key-sequence too fast, ADAM can lock-up. To avoid, pause 2 to 3 seconds between keypresses in this sequence.

3. Sometimes, if you use a filename (for STOREing) that already exists on the datapak (or disk), ADAM will lock-up. With this one, the tape just spins *ad infinitum*. To avoid, take care to not use a filename that already exists.

4. In the 'MOVING WINDOW' screen format option, DELETE, INSERT, and MOVE/COPY can cause a temporary text displacement on the screen. Pressing the HOME+left arrow will re-align the screen.

NOTE: SmartWriter is usually much easier to work with using the 'MOVING WINDOW' option. Here, the screen scrolls faster and you have better cursor control. To make the best use of this feature, leave the left margin at '10' and set the right margin to '45'. This way the entire 'window' is on the screen at the same time. Then, when you're ready to PRINT, adjust the margins to your preferences.

BIT BY BIT

PROGRAMMING PERSPECTIVE

Creating programs is sometimes compared to assembling a jigsaw puzzle. In this analogy, each BASIC command is considered to be an individual puzzle piece. However, BASIC commands can be organized into an almost infinite number of combinations each producing its own unique end result.

When you first start programming, you need to familiarize yourself with each command. Learn its function, its acceptable parameters, and how it can be interwoven with other commands and then you're well on your way to becoming a proficient programmer.

In the beginning this learning process is usually time consuming and can sometimes be a little frustrating. The single characteristic that distinguishes successful programming hobbyists from those who fail is DETERMINATION.

Set goals for yourself. First, develop a thorough working knowledge of each command. Experiment, using simple programs (one to ten lines), with each BASIC command. During this process, type in programs from books, magazines, newsletters, etc. Then, modify these programs. Add your own personal touch or, maybe, embellish them with added features.

The next step is to start developing your own programs -- creations of your own mind. This is the beginning point where programming becomes most enjoyable and very rewarding. Think of what you want a program to do. Then divide that primary idea into several subordinate routines. Develop each routine individually and then combine them all together for a finished product. Later, you may want to add minor improvements here and there.

If you maintain your determination, one day you'll reach the point that you're limited only by your creativity and the peripherals attached to your system. Computers, as we all know, are working their way into nearly every profession. Although ADAM may not be the most popular personal computer, there are numerous similarities with others. SmartBASIC resembles other BASICs, SmartLOGO resembles other LOGOs, CP/M resembles other "universal" operating systems, and even Z-80 assembly language is similar to the assembly language of other CPUs. Regardless of the system that you may one day use at work, you'll find that your use of ADAM (at home) is an invaluable aid.

THE END COMMAND

The END command is used to terminate a program's execution. SmartBASIC will automatically end a program with the last statement. In some situations, however, you may want to stop a program before the highest line number is reached. Consider this example.

```
10 PRINT " Programming sure is"
20 END
30 PRINT " simple and fun!!!"
```

When you RUN this program, it will stop as soon as it reaches the END command. Line# 30 won't be PRINTed.

THE STOP COMMAND

The STOP command is almost identical to the END command. The only difference, of any consequence, is that when the program STOPs, SmartBASIC displays an error message.

In the program above, replace END in line# 20 with STOP. This time when you RUN the program, it will PRINT " Programming sure is" and then display "?Break In 20". Whether or not you use END or STOP is purely a personal preference.

THE CONT COMMAND

The CONT command will restart a program that was stopped with END, STOP, or CNTL-C (providing that the program is not logically finished). In the example above, you can PRINT line# 30 by simply entering CONT [RETURN].

BYTE-SIZED BASIC

POKES TO PLAY WITH
(part 4)

CONTROL FUNCTIONS:

Several of BASIC's control functions (CNTL + another key) have corresponding single keypress equivalents. For example, [RETURN] is the same as CNTL-M, [TAB] is the same as CNTL-I, and [BACKSPACE] is the same as CNTL-H.

You may find that BASIC is a little easier to work with by converting some of the other common control functions to single keypresses. CNTL-C (break), CNTL-N (insert space on screen line), CNTL-O (delete character on screen line), and CNTL-S (pause printing) are each used frequently by the typical programmer.

Address 16134 contains the ASCII value for CNTL-C. By default, it is a 3. If you POKE a 27 into 16134, you can convert the CNTL-C function to the single keypress [ESCAPE]. If you POKE a 255 into that address you will effectively disable CNTL-C.

Address 12374 contains the ASCII value for the CNTL-N function. If you POKE a 148 into 12374, you can convert the CNTL-N function to the single keypress [INSERT] (unSHIFTed).

Address 12375 contains the ASCII value for the CNTL-O function. If you POKE a 151 into 12375, you can convert the CNTL-O function to the single keypress [DELETE] (unSHIFTed).

Address 16135 contains the ASCII value for CNTL-S. By default, it is a 19. If you POKE a 144 into 16135, you can convert the CNTL-S function to the single keypress [WILDCARD] (unSHIFTed). Address 16136 contains the "PRINT pause" status. You can force a PRINT pause within a program by POKEing a zero into that address.

HOW TO DISABLE "ONERR GOTO":

Have you ever come across a BASIC program that uses error trapping so effectively that you can not LIST it? Some programmers do this with binary converted programs making them virtually LISTproof. The following technique may be of help; it disables ONERR GOTO.

Address 8114 is the start of the machine language execution routine for ONERR GOTO. Its default value is 217. If you POKE a 201 (machine code for return) into address 8114, you'll effectively disable ONERR GOTO.

As an alternative, you may choose to POKE a zero into 8114. With this technique, BASIC will display an error message as soon as the ONERR GOTO command is encountered. This causes BASIC to respond as if it's in immediate execution mode (without line numbers) instead of programming mode. The error message is "?Illegal Mode Error".

MERGING BASIC PROGRAMS:

There are many situations in which it is beneficial to merge BASIC routines and/or programs into one larger program. And, there are at least four different ways of accomplishing this. However, you should note that when two line numbers are the same, the last one entered (or LOADed) takes precedence and erases the previous identical line number.

Programs can be merged with SmartWriter. It can also be done, from BASIC, in conjunction with the OPEN command. However, both of these methods are a little difficult and time consuming. The easiest technique is to slightly modify BASIC, so that the LOAD command performs a merge function directly in RAM. (SmartBASIC 2.0 includes a MERGE command.)

In the July issue we revealed a simple trick which disables the NEW function, thus causing LOAD to merge programs. To enable merging, POKE a 201 into address 6356. To restore the NEW function, and disable merging, POKE a 205 (the default value) into 6356.

An alternative is to modify the LOAD command itself. LOAD jumps to the NEW function with the machine language command at 24009. The default value of 24010 is 212. The default value of 24011 is 24 (24*256 + 212 = 6356). If you POKE a 224 into 24010, LOAD will skip the part of the NEW function that deletes the current program. Be sure to POKE 212 back into 24010 when you're finished merging.

Please note that these POKE tricks only work with 'A' filetypes LOADed from a storage medium. When a binary converted program is BRUN, it automatically erases the current program.

PLAYING WITH MUSIC

The addition of music to your programs can add an impressive touch. Intel-BEST 3.3, SmartBEST V1.0, and the HELLO program from the Hinkle's "HACKER'S GUIDE TO ADAM: VOLUME TWO" add actual sound commands to standard SmartBASIC.

However, there is a (somewhat restricted) alternative without delving too deeply into machine language. Simply modify the 'PRINT CHR$(7)' function. The three programs on page 10 do just that.

In the August issue we mentioned the POKEs which control the internal bell's tone. All you have to do is change the various values and you can create some interesting sounds.

The duration of the bell's tone is controlled by addresses 17962 and 17963. In machine language ADAM counts backwards until zero is reached using these values.

The volume of the tone is controlled by address 17958. For the first voice (which the bell uses) the volume may be any value between 144 (the loudest) and 159 (no sound).

A total of 1024 different sounds are possible. However, no individual address may contain a value greater the 255. So the sound chip requires that the tone value (0 - 1023) be separated into two smaller values. Here's how these two bytes are calculated:

```
first byte = INT( value / 64 ) + offset
second byte = value - INT( value / 64 ) * 64
```

The value must be an integer between 0 and 1023 inclusive. The offset is different for each voice. For the first voice it is 128. For the second voice it is 160 and for the third it is 192.

Address 17950 contains the first tone byte. And, address 17954 contains the second tone byte.

By experimenting with these various values you can not only have some fun with ADAM; but, you can also add sound effects to your own programs. You can use the three programs on the next page as guidelines.

CREATING A USEFUL QUIZ

The simple presidential quiz (LIST occupies pages 11 and 12) illustrates several aspects that are condusive to an effectively useful quiz. If you would like to brush up on your knowledge of the sequence of U.S. presidents, try this quiz.

The program uses the randomization of DATA technique mentioned last month. It groups the presidents into four chronological sets of ten. Learning and/or retention is easiest when taken in small steps rather than large chunks.

To begin, you select which set you want to quiz on. Then, it automatically shows you a review page for that particular set.

Now the quiz begins. You choose from one of five multiple choices. Each of these is selected from the chosen set of ten. Correct answers are rewarded with a simple sound effect and incorrect responses are buzzed. Your score is revealed after each answer. And, your final tally is given at the end of the quiz along with an evaluation of your performance.

In theory, you could take the quiz a thousand times and never have the presidents asked in the same sequence with the same multiple choices. This also helps in the learning process.

Except for lines 1040 through 1200 the program is very simple. This is the module that selects the multiple choices.

Line# 1040 resets the dimensioned 'du$(x)' variables for each question. The 'FRE' function is inserted here to prevent 'string garbage'. Line# 1050 selects the number (1 - 5) that the correct answer is assigned to. Line# 1060 begins the printing procedure. Line# 1070 checks to see if the current multiple choice is the one allocated for the correct answer. Line# 1080 selects a random answer. And, Line# 1090 checks to see if the random answer is Grover Cleveland's second term (he was 22nd and 24th president).

The value of the dimensioned 'du$(x)' variables are changed to 'taken' when they are used in each question. And, as a question is asked that president's rank, 'rk$(q2)', is changed to 'used'. These two steps are taken in order to prevent duplication.

```
  10 REM easy music #1
  20 REM (uses the built-in bell)
 100 TEXT: POKE 17963,3
 110 FOR x = 32 TO 1 STEP -1
 120 POKE 17954,x: PRINT CHR$(7);: NEXT
 130 POKE 17963,7: POKE 17954,17
 140 LIST
```

```
  10 REM easy music #2
  20 REM (uses the built-in bell)
 100 FOR x = 4 TO 48 STEP 4
 110 POKE 17963,(x/4): POKE 17954,x
 120 PRINT CHR$(7);: NEXT
 130 POKE 17963,7: POKE 17954,17
 140 LIST
```

```
  10 REM easy music #3
  20 REM (uses the built-in bell)
 100 TEXT: PRINT " 1 = change tone": PRINT " 2 = change volume"
 110 PRINT " 3 = change duration": PRINT " 4 = sound the bell"
 120 PRINT " 5 = exit the program"
 130 GET key$: k% = VAL(key$)
 140 PRINT: PRINT: ON k% GOSUB 1000,2000,3000,4000,5000: GOTO 100
1000 f1 = PEEK(17950)-128: f2 = PEEK(17954): ct = f1*64+f2
1010 PRINT " current tone value = ";ct: PRINT
1020 INPUT " enter new tone (0-1023): ";nt
1030 IF nt < 0 OR nt > 1023 GOTO 1020
1040 hi% = nt/64: lo% = nt-(64*hi%)
1050 POKE 17950,hi%+128: POKE 17954,lo%: GOTO 4000
2000 cv = 159-PEEK(17958)
2010 PRINT " current volume = ";cv: PRINT
2020 INPUT " enter new volume (0-15): ";vo
2030 IF vo < 0 OR vo > 15 GOTO 2020
2040 POKE 17958,159-vo: GOTO 4000
3000 du = (PEEK(17963)*256)+PEEK(17962)
3010 PRINT " current duration = ";du: PRINT
3020 PRINT " enter new duration"
3030 INPUT " (1 - 65535): ";nd
3040 IF nd < 1 OR nd > 65535 GOTO 3030
3050 hi% = nd/256: lo% = nd-(256*hi%)
3060 POKE 17962,lo%: POKE 17963,hi%: GOTO 4000
4000 PRINT CHR$(7): RETURN
5000 POKE 17950,143: POKE 17954,17: POKE 17958,144
5010 POKE 17962,128: POKE 17963,7: TEXT
5020 PRINT " program terminated.": END
```

```
  10 REM simple presidential sequence quiz
  20 REM demonstates use of true randomization
  30 REM and randomizing data techniques
 100 TEXT: SPEED  = 255: nb% = 40: DIM rk$(nb%),pd$(nb%),du$(nb%)
 110 INVERSE: PRINT " SIMPLE PRESIDENTIAL QUIZ ": NORMAL: PRINT
 120 PRINT: PRINT " one moment please . . ."
 130 GOSUB 10000: FOR x = 1 TO 5: READ menu$(x): NEXT
 140 FOR x = 0 TO 10: READ rate$(x): NEXT
 150 POKE 16149,255: POKE 16150,255
 160 score = 0: ct = 1
 500 HOME: PRINT: PRINT " Which option do you prefer?": PRINT
 510 FOR x = 1 TO 5: PRINT " ";x;" = ";menu$(x): NEXT
 520 POKE 64885,0
 530 kp% = PEEK(64885): IF kp% <> 0 GOTO 550
 540 ct = ct+1: ON ct > 32768 GOTO 10100: GOTO 530
 550 IF kp% < 49 OR kp% > 53 THEN  PRINT CHR$(7);: GOTO 520
 560 xx = RND(-ct)
 570 kp% = kp%-48: IF kp% = 5 GOTO 10100
 580 lo = 10*kp%-9: hi = lo+9: q1 = hi-lo+1
 700 HOME: PRINT " ";: INVERSE: PRINT " easy review page ": NORMAL
 710 PRINT: PRINT: PRINT " ";: INVERSE: PRINT " # ";: NORMAL
 720 PRINT " ";: INVERSE: PRINT " president": NORMAL
 730 PRINT: FOR x = lo TO hi
 740 PRINT " ";: IF x < 10 THEN  PRINT " ";
 750 PRINT x;": ";pd$(x): NEXT
 760 VTAB 20: PRINT " press any key to continue";
 770 GET key$
1000 FOR quest = 1 TO 10
1010 q2 = INT(RND(1)*q1)+lo: IF rk$(q2) = "used" GOTO 1010
1020 HOME: PRINT: PRINT " #";quest
1025 PRINT " Who was the ";rk$(q2)
1030 PRINT " US president?": PRINT: PRINT
1040 FOR x = 1 TO nb%: du$(x) = pd$(x): NEXT: ff = FRE(0)
1050 cr = INT(RND(1)*5)+1
1060 FOR x = 1 TO 5
1070 IF x = cr THEN  pt$ = pd$(q2): GOTO 1200
1080 q3 = INT(RND(1)*q1)+lo: IF q3 = q2 GOTO 1080
1090 IF q3 = 24 GOTO 1080
1100 IF du$(q3) = "taken" GOTO 1080
1110 pt$ = pd$(q3): du$(q3) = "taken"
1200 PRINT " ";x;"= ";pt$: NEXT x
1300 PRINT: PRINT: PRINT " select with number keys . . ."
1310 PRINT: PRINT " ";
1320 GET ans$: ans% = VAL(ans$)
1330 IF ans$ = CHR$(27) THEN  RUN
1340 IF ans$ < "1" OR ans$ > "5" THEN  PRINT CHR$(7);: GOTO 1320
1350 INVERSE: PRINT ans%: NORMAL: PRINT: IF ans% = cr GOTO 1500
1400 PRINT " Sorry . . . incorrect!!!": point = 0
1410 PRINT " ";pd$(q2)
1420 PRINT " was the ";rk$(q2);" president."
1430 FOR x = 63 TO 43 STEP -1
1440 POKE 17954,x: PRINT CHR$(7);: NEXT
1450 POKE 17954,17: GOTO 1600
```

```
1500 PRINT " That's absolutely correct!!!": point = 10
1510 GOSUB 10200
1600 score = score+point: perc = (score/quest)*10
1610 PRINT: PRINT " ";perc;"% out of ";quest
1620 PRINT: PRINT " press any key to continue";
1630 rk$(q2) = "used"
1640 GET key$: IF key$ = CHR$(27) THEN  RUN
1650 NEXT quest: HOME: PRINT
2000 INVERSE: PRINT " category: ";: NORMAL
2010 PRINT " ";menu$(kp%): PRINT
2020 INVERSE: PRINT " correct:  ";: NORMAL
2030 PRINT " ";score/10: PRINT
2040 INVERSE: PRINT " missed:   ";: NORMAL
2050 PRINT " ";10-score/10: PRINT
2060 INVERSE: PRINT " grade:    ";: NORMAL
2070 PRINT " ";perc: PRINT: PRINT: PRINT
2080 INVERSE: PRINT " rating:   ";: NORMAL: PRINT
2090 PRINT: PRINT " ";rate$(score/10)
2100 IF perc < 90 GOTO 2200
2110 FOR y = 1 TO 5: GOSUB 10200: NEXT
2200 VTAB 22: PRINT " press any key to continue";
2210 GET key$: RUN
5000 DATA 1st,George Washington,2nd,John Adams
5010 DATA 3rd,Thomas Jefferson,4th,James Madison
5020 DATA 5th,James Monroe,6th,John Quincy Adams
5030 DATA 7th,Andrew Jackson,8th,Martin Van Buren
5040 DATA 9th,William Henry Harrison,10th,John Tyler
5050 DATA 11th,James Knox Polk,12th,Zachary Taylor
5060 DATA 13th,Millard Fillmore,14th,Franklin Pierce
5070 DATA 15th,James Buchanan,16th,Abraham Lincoln
5080 DATA 17th,Andrew Johnson,18th,Ulysses Simpson Grant
5090 DATA 19th,Rutherford Birchard Hayes,20th,James Abram Garfield
5100 DATA 21st,Chester Alan Arthur,22nd,Grover Cleveland
5110 DATA 23rd,Benjamin Harrison,24th,Grover Cleveland
5120 DATA 25th,William McKinley,26th,Theodore Roosevelt
5130 DATA 27th,William Howard Taft,28th,Woodrow Wilson
5140 DATA 29th,Warren Gamaliel Harding,30th,Calvin Coolidge
5150 DATA 31st,Herbert Clark Hoover,32nd,Franklin Delano Roosevelt
5160 DATA 33rd,Harry Truman,34th,Dwight David Eisenhower
5170 DATA 35th,John Fitzgerald Kennedy,36th,Lyndon Baines Johnson
5180 DATA 37th,Richard Milhous Nixon,38th,Gerald Rudolph Ford
5190 DATA 39th,Jimmy Carter,40th,Ronald Wilson Reagan
6000 DATA 1st thru 10th,11th thru 20th,21st thru 30th
6010 DATA 31st thru 40th,exit the quiz
7000 DATA you need a lot of practice,try studying the list
7010 DATA practice!!!,think before you answer
7020 DATA keep practicing,better luck next time
7030 DATA practice makes perfect,not too bad
7040 DATA looking good!!!,a very nice score!!!!!
7050 DATA GREAT!!!  A perfect score!!!
10000 FOR x = 1 TO nb%: READ rk$(x),pd$(x): NEXT: RETURN
10100 TEXT: PRINT "program terminated.": END
10200 FOR x = 15 TO 1 STEP -1
10210 POKE 17954,x: PRINT CHR$(7);: NEXT
10220 POKE 17954,17: RETURN
```

HACKER'S DELIGHT

BIT CONFIGURATIONS (part 2)

In the August issue we briefly touched on bit configurations (in conjunction with examining video register one). Many of the more interesting aspects of machine language programming involve an understanding of bit configurations, eg, designing fonts, creating sprites, machine code decisions, masking, etc.

Each byte consists of eight bits. A "set" bit has a value of logical one. A "reset" bit has a value of logical zero.

The eight bits (of any byte) are numbered by sequential integers from the rightmost bit. The rightmost bit is numbered "0". The leftmost bit is numbered "7". Each of these numbers represents a power of two. For example, if all bits are reset (logical zero) except the leftmost (#7) bit, then that byte has a value of 2 to the 7th power or 128.

Understanding bit configurations is just as important to machine code endeavors as comprehending the hexadecimal system is. Here are a few examples for you to experiment with. Please take the time to become familiar with this concept.

SPECIAL NOTE: Using bit configurations in designing fonts and sprites for the monitor screen is often called "bit mapping". And, using bit configurations on dot-matrix printers is usually referred to as "bit image graphics".

10000001 =
$2^7 + 2^0 =$
128 + 1 =
129

01000010 =
$2^6 + 2^1 =$
64 + 2 =
66

11011011 =
$2^7 + 2^6 + 2^4 + 2^3 + 2^1 + 2^0 =$
128 + 64 + 16 + 8 + 2 + 1 =
219

11110000 =
$2^7 + 2^6 + 2^5 + 2^4 =$
128 + 64 + 32 + 16 =
240

00001111 =
$2^3 + 2^2 + 2^1 + 2^0 =$
8 + 4 + 2 + 1 =
15

11111111 =
$2^7 + 2^6 + 2^5 + 2^4 \quad 2^3 + 2^2 + 2^1 + 2^0 =$
128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 =
255

00000000 =
0

THE DIRECTORY (part 2)

In the August issue we mentioned that each directory entry is allocated a 26-byte slot of file information. The 13th byte in each slot is the file's attribute.

The EOS uses these attribute bytes for screening filenames. The program on the next two pages (pages 14 and 15) allows you to change file attributes. The list below explains each bit of the attribute byte.

BIT #7 (set value = 128)
* delete protect
(this bit is set with LOCK)

BIT #6 (set value = 64)
* write protect
(when set, prevents appending)

BIT #5 (set value = 32)
* read protect
(when set, file can not be read)

BIT #4 (set value = 16)
* default setting
(has no effect on attribute)

BIT #3 (set value = 8)
* display protect
(when set, prevents normal display)

- continued on page 16 -

```
  10 REM file attribute changer
 100 LOMEM :34000: POKE 16149,255: POKE 16150,255
 110 POKE 61412,0: POKE 61413,0
 200 DATA 62,8,17,235,255,33,160,253,205,204,252,50,255,107,201
 210 FOR x = 27600 TO 27614: READ ml: POKE x,ml: NEXT
 220 p1 = 27600: p2 = 27601: p3 = 27609
 230 DATA tape one,disk one: FOR x = 1 TO 2: READ dv$(x): NEXT
 300 DATA delete protect,write protect,read protect
 310 DATA default setting,display protect,file deleted
 320 DATA unlock protect,directory end
 330 FOR x = 7 TO 0 STEP -1: READ aa$(x): NEXT
 500 TEXT: VTAB 2: HTAB 2: INVERSE: PRINT " file attribute changer "
 510 NORMAL: VTAB 6: PRINT " Which drive?": PRINT
 520 FOR x = 1 TO 2: PRINT " ";x;" = ";dv$(x): NEXT
 530 PRINT: PRINT " [escape] = exit program"
 540 GET key$: ON key$ = CHR$(27) GOTO 10000: k% = VAL(key$)
 550 IF k% < 1 OR k% > 2 THEN  PRINT CHR$(7);: GOTO 540
 560 dv% = 8/k%: POKE p2,dv%
 600 HOME: PRINT " current device: ";dv$(k%): PRINT
 610 PRINT " 1 = catalog device": PRINT " 2 = enter filename"
 620 PRINT " 3 = change devices/exit"
 630 GET opt$: opt% = VAL(opt$)
 640 IF opt% < 1 OR opt% > 3 THEN  PRINT CHR$(7);: GOTO 630
 650 ON opt% GOTO 1000,2000,500
1000 cd% = PEEK(16821): POKE 16821,dv%
1010 HOME: PRINT " for a catalog,"
1020 GOSUB 11000: IF go% = 27 GOTO 600
1100 HOME: PRINT CHR$(4);"catalog": POKE 16821,cd%
1110 PRINT: PRINT " press any key for menu . . ."
1120 GET go$: GOTO 600
2000 HOME: PRINT: PRINT " * end filename with file type!"
2010 PRINT " * press [return] when done.": VTAB 10
2020 INPUT " enter filename: ";fi$
2100 le% = LEN(fi$): IF le% <= 11 GOTO 2200
2110 PRINT CHR$(7): PRINT " filename too long!!"
2120 PRINT " try again . . ."
2130 PRINT: PRINT: GOTO 2020
2200 lt$ = RIGHT$(fi$,1): IF lt$ = "A" OR lt$ = "a" GOTO 2300
2210 IF lt$ = "H" OR lt$ = "h" OR lt$ = CHR$(2) GOTO 2300
2220 PRINT CHR$(7): PRINT " end filename with file type!"
2230 GOTO 2120
2300 HOME: PRINT " to verify filename,"
2310 GOSUB 11000: ON go% = 27 GOTO 600: GOSUB 11500
2320 f2$ = fi$+CHR$(3): le% = le%+1
2330 FOR x = 1 TO le%: POKE 65514+x,ASC(MID$(f2$,x,1)): NEXT
2340 POKE p3,204: CALL p1
2350 IF PEEK(27647) = 0 GOTO 3000
2400 IF PEEK(27647) <> 22 GOTO 2500
2410 HOME: PRINT " missing ";LEFT$(dv$(k%),4);"!!!"
2420 PRINT: PRINT " press any key for menu . . ."
2430 GET go$: GOTO 600
2500 HOME: PRINT " filename not found!!!": GOTO 2420
```

```
 3000 HOME: VTAB 2: HTAB 2: INVERSE: PRINT " filename: ";: NORMAL
 3010 PRINT "  ";fi$: PRINT: INVERSE
 3020 ab = PEEK(64940): HTAB 2: PRINT " attribute value: ";
 3030 NORMAL: PRINT "  ";ab: PRINT: PRINT
 3100 FOR x = 7 TO 0 STEP -1: PRINT " ";x;": ";: INVERSE
 3110 PRINT " ";aa$(x);":": NORMAL: NEXT
 3120 GOSUB 12000: GOSUB 12500
 3200 VTAB 16: PRINT " * enter # (1 - 7) to change"
 3210 PRINT "   the value of a bit"
 3220 PRINT " * enter '8' to change the ";LEFT$(dv$(k%),4);
 3230 PRINT " * enter '9' for menu without "
 3240 PRINT "   an attribute change"
 3300 VTAB 23: PRINT " enter a number . . ."
 3310 GET nu$: nu% = VAL(nu$)
 3320 IF nu$ < "1" OR nu$ > "9" THEN  PRINT CHR$(7);: GOTO 3310
 3330 IF nu% = 9 GOTO 600
 3340 IF nu% = 8 GOTO 4000
 3400 IF ab$(nu%) = "on" THEN  ab = ab-2^nu%: GOTO 3420
 3410 ab = ab+2^nu%
 3420 POKE 64940,ab: GOSUB 12000: GOSUB 12500
 3430 VTAB 4: HTAB 22: PRINT ab: GOTO 3310
 4000 HOME: PRINT " one moment please . . ."
 4010 IF PEEK(64940) = 0 THEN  POKE 64940,16
 4020 POKE p3,207: CALL p1
 4030 IF PEEK(27647) = 0 GOTO 4200
 4100 PRINT CHR$(7): PRINT " can not write the change to"
 4110 PRINT " ";dv$(k%);"!!!": PRINT
 4120 GOTO 4210
 4200 HOME: PRINT: PRINT " attribute changed,"
 4210 PRINT " press any key for menu . . ."
 4220 GET go$: GOTO 600
10000 TEXT: PRINT " program terminated."
10010 POKE 61412,203: POKE 61413,86: END
11000 VTAB 4: PRINT: PRINT " press [return] to continue ..."
11010 PRINT " press [escape] for menu ..."
11020 GET go$: go% = ASC(go$)
11030 IF go% <> 13 AND go% <> 27 THEN  PRINT CHR$(7);: GOTO 11020
11040 RETURN
11500 HOME: PRINT " one moment please . . .": RETURN
12000 FOR x = 0 TO 7: ab$(x) = "off": NEXT: ba = ab
12010 FOR x = 7 TO 0 STEP -1
12020 IF ba >= 2^x THEN  ba = ba-2^x: ab$(x) = "on"
12030 NEXT: RETURN
12500 FOR x = 7 TO 0 STEP -1: VTAB 14-x
12510 HTAB 23: PRINT ab$(x): NEXT: RETURN
```

BIT #2 (set value = 4)
 * delete status
 (when set, file deleted)

BIT #1 (set value = 2)
 * unlock protect
 (when set, BASIC can not UNLOCK)

BIT #0 (set value = 1)
 * directory end
 (when set, indicates end of filenames)

THE PRIMARY COMMAND TABLE

Addresses 272 through 817 in SmartBASIC V1.0 constitute the primary command table. This table contains BASIC words such as, GOTO, REM, LIST, etc. Along with the ASCII spelling of each word is other vital information. The format for the table is as follows:

1 byte:	command parsing token
2 bytes:	vector to parameter check(s)
1 byte:	number of letters in the BASIC word
1-7 bytes:	ASCII spelling of the BASIC word

The parsing token is used in the binary coded version of a program. To determine the execution vector of a particular command, multiply the token number by two and add that product to 6421 (for SmartBASIC V1.0).

The lists on the next three pages show each primary BASIC command, its execution address, and its parameter check address(es). Commands that don't have parameter checks can be executed simply by CALLing the execution address.

The program on page 20 will print the data for the SmartBASIC V1.0 primary command table. For Intel-BEST 3.3, change the value of 'ex' (line# 150) to 65378. For SmartBASIC V2.0, change the value of 'ex' to 7238 and the '817' in line# 900 to and '812'.

HOW TO SAVE AN HGR SCREEN

The program at the top of page 21 shows you how to save an HGR screen in RAM. It uses previously discussed EOS routines. The program draws a hires picture and then saves it in RAM. You'll notice that recalling the saved picture is considerably faster than the original drawing.

You may find these routines most useful in storing and retrieving pictures on data pack or disk.

To store:
1. draw picture on HGR screen
2. CALL 27600
3. BSAVE (filename), A27648, L10240

To retrieve:
1. BLOAD (filename), A27648
2. HGR
3. CALL 27624

With either procedure YOU MUST SET LOMEM to 37888 or higher at the beginning of the program!!!

PLAYING WITH FONTS
(part 1)

As mentioned in the premier issue, addresses zero through 2047 in VRAM define the shape of each font (in 8-byte sets) in TEXT mode. The last byte in each set is normally set to zero; this is to allow for vertical spacing. However, you can set this byte to a '255' to create underscoring.

The program on the bottom of page 21 shows you an easy to understand technique for creating this underscoring of the INVERSE fonts. The program on the top of page 22, moreover, demonstates a superior machine code algorithm. It INSTANTLY converts the INVERSE fonts to underscored. The assembly language of this routine is detailed in asmb#18 on page 23. This routine is independent of its RAM address, so that you can store it anywhere you that your situation warrants.

In this routine BC is used to keep track of the remaining number of unchanged fonts. DE indicates that only one byte's value will be changed in each pass of the main LOOP. HL is used to store the current VRAM address.

The 'XOR A' instruction in line #4 is a Z80 trick for resetting the accumulator and the flags register. It is similar to 'LD A, $00'.

Each of the control registers must be PUSHed before CALLing the EOS routine. Then, they are retrieved (in reverse order) to restore the control values.

The relative jump in line #20 repeats the LOOP. A signed displacement of 233 is equivalent to 23 bytes backwards (256 - 233 = 23).

SmartBASIC V1.0 PRIMARY COMMAND TABLE

TOKEN	COMMAND	EXECUTION	PARAMETER ROUTINE(S)				
1		6247	15020				
2	GOSUB	8427	15756				
3	GOTO	8342	15756				
4	INPUT	8957	15543				
5	LET	6247	15020				
6	NEXT	8811	15567				
7	PRINT	7854	15580				
8	READ	9499	15574				
9	REM	8419	15817				
10	FOR	8557	14991	15963	14875	15093	
11	IF	7705	14947	15035			
12	DATA	8419	15814				
13	DIM	6942	15574				
14	ON	8381	14875	15209			
15	ONERR	8114	15991	15756			
16	STOP	6378					
17	RETURN	8477					
18	END	6047					
19	DEF	8244	15125				
20	CLEAR	8141					
21	RESUME	8313					
22	NEW	6356					
23	POP	8493					
24	RUN	6159	15232				
25	LIST	7407	15243				
26	TRACE	6336					
27	NOTRACE	6341					
28	DEL	7555	15247				
29	CALL	10042	14875				
30	CONT	6387					
31	CLRERR	8109					
32	GET	9378	15364				
33	POKE	10104	14875	15939	14875		
34	RESTORE	9482					
35	HOME	11090					
36	DRAW	11358	14875	14976			
37	XDRAW	11412	14875	14976			
38	FLASH	11050					
39	INVERSE	11055					
40	NORMAL	11060					
41	TEXT	11065					
42	GR	11070					
43	HGR	11075					
44	HGR2	11080					
45	HLIN	11170	14875	15939	14875	15977	14875
46	VLIN	11219	14875	15939	14875	15977	14875
47	HPLOT	11487	15102				
48	PLOT	11139	14875	15939	14875		
49	HTAB	11320	14875				
50	VTAB	11330	14875				
51	SHLOAD	11085					
52	RECALL	11764	15364				
53	STORE	11756	15364				
54	WAIT	10126	14875	15939	14875	14969	
55	SPEED	10832	15926	14875			
56	ROT	11459	15926	14875			
57	SCALE	11473	15926	14875			
58	COLOR	11099	15926	14875			
59	HCOLOR	11119	15926	14875			
60	IN	12084	15950	14875			
61	PR	12058	15950	14875			
62	HIMEM	11010	15911	14875			
63	LOMEM	10870	15911	14875			
64	BREAK	6346					
65	NOBREAK	6351					
7	?	7854	15580				
66	&	10164	15817				

Intel-BEST 3.3 PRIMARY COMMAND TABLE

TOKEN	COMMAND	EXECUTION	PARAMETER ROUTINE(S)				
1		6247	15020				
2	GOSUB	8427	15756				
3	GOTO	8342	15756				
4	INPUT	8957	15543				
5	LnB	27407					
6	NEXT	8811	15567				
7	PRINT	7854	15580				
8	READ	9499	15574				
9	REM	8419	15817				
10	FOR	8557	14991	15963	14875	15093	
11	IF	7705	14947	15035			
12	DATA	8419	15814				
13	DIM	6942	15574				
14	ON	8381	14875	15209			
15	ONERR	8114	15991	15756			
16	STOP	6378					
17	RETURN	8477					
18	END	6047					
19	DEF	8244	15125				
20	CLEAR	8141					
21	RESUME	8313					
22	NEW	27459					
23	POP	8493					
24	RUN	6159	15232				
25	LIST	7407	15243				
26	LINE	1075	15756				
27	NV	1062	15926	14875			
69	NS	222	15926	14875			
28	DEL	7555	15247				
29	CALL	10042	14875				
30	CONT	6387					
31	CLRERR	8109					
32	GET	9378	15364				
33	POKE	10104	14875	15939	14875		
34	RESTORE	9482					
35	HOME	11090					
36	DRAW	11358	14875	14976			
37	XDRAW	11412	14875	14976			
38	FLASH	11050					
39	INVERSE	11055					
40	NORMAL	11060					
41	TEXT	11065					
42	GR	11070					
43	HGR	11075					
44	HGR2	11080					
45	HLIN	11170	14875	15939	14875	15977	14875
46	VLIN	11219	14875	15939	14875	15977	14875
47	HPLOT	11487	15102				
48	PLOT	11139	14875	15939	14875		
49	HTAB	11320	14875				
50	VTAB	11330	14875				
51	V2	65333	15926	14875			
52	V3	65346	15926	14875			
53	T1	172	15926	14875			
54	T2	6423	15926	14875			
67	T3	197	15926	14875			
68	OFF	64851					
55	SPEED	10832	15926	14875			
56	ROT	11459	15926	14875			
57	SCALE	11473	15926	14875			
58	COLOR	11099	15926	14875			
59	HCOLOR	11119	15926	14875			
60	V1	65320	15926	14875			
61	PR	12058	15950	14875			
62	HIMEM	11010	15911	14875			
63	LOMEM	10870	15911	14875			
38	F	11050					
35	H	11090					
39	I	11055					
40	N	11060					
7	?	7854	15580				
41	T	11065					

SmartBASIC V2.0 PRIMARY COMMAND TABLE

TOKEN	COMMAND	EXECUTION	PARAMETER ROUTINE(S)				
1		7069	15519				
2	GOSUB	9146	16255				
3	GOTO	9069	16255				
4	INPUT	9690	16042				
5	LET	7069	15519				
6	NEXT	9516	16066				
7	PRINT	8548	16079				
8	READ	10185	16073				
9	REM	9138	16323				
10	FOR	9262	15490	16472	15374	15592	
11	IF	9042	15446	15534			
12	DATA	9138	16320				
13	DIM	7761	16073				
14	ON	9103	15374	15708			
15	ONERR	8814	16500	16255			
16	STOP	7195					
17	RETURN	9191					
18	END	6868					
19	DEF	8944	15624				
20	CLEAR	8841					
21	RESUME	9013					
22	NEW	7173					
23	POP	9207					
24	RUN	6987	15731				
25	LIST	8175	15742				
26	TRACE	7163					
27	NOTRACE	7168					
28	DEL	8304	15746				
29	CALL	10704	15374				
30	CONT	7204					
31	CLRERR	8809					
32	GET	10072	15863				
33	POKE	10766	15374	16448	15374		
34	RESTORE	10168					
35	HOME	11723					
36	DRAW	11950	15374	15475			
37	XDRAW	11978	15374	15475			
38	FLASH	11688					
39	INVERSE	11693					
40	NORMAL	11698					
41	TEXT	11703					
42	GR	11708					
43	HGR	11713					
44	HGR2	11718					
45	HLIN	11801	15374	16448	15374	16486	15374
46	VLIN	11822	15374	16448	15374	16486	15374
47	HPLOT	12019	15601				
48	PLOT	11769	15374	16448	15374		
49	HTAB	11888	15374				
50	VTAB	11898	15374				
51	SHLOAD	7172					
52	RECALL	9138	15863				
53	STORE	9138	15863				
54	WAIT	10788	15374	16448	15374	15468	
55	SPEED	11470	16435	15374			
56	ROT	11999	16435	15374			
57	SCALE	12009	16435	15374			
58	COLOR	11739	16435	15374			
59	HCOLOR	11754	16435	15374			
60	IN	12581	16459	15374			
61	PR	12551	16459	15374			
62	HIMEM	11648	16420	15374			
63	LOMEM	11508	16420	15374			
64	0	7172					
65	0	7172					
7	?	8548	16079				
66	&	10826	16323				
67	7,2	8642	16323				

```
   10 REM primary command list
   20 REM requires fan-fold paper
   30 REM designed for SmartBASIC V1.0
  100 TEXT: PRINT " insert paper in printer ..."
  110 PRINT: PRINT " - press [RETURN] to start -"
  120 GET key$: IF key$ <> CHR$(13) GOTO 10000
  150 adr = 272: ex = 6421
  200 PR #2: PRINT SPC(24);"Intel-BEST 3.3 PRIMARY COMMAND TABLE": PRINT
  210 PRINT "TOKEN     COMMAND     EXECUTION";
  220 PRINT "     PARAMETER ROUTINE(S)": PRINT
  300 tk$ = STR$(PEEK(adr))
  310 e1 = ex+2*PEEK(adr): e2 = PEEK(e1)+(256*PEEK(e1+1))
  320 p1 = PEEK(adr+1): p2 = PEEK(adr+2): p3 = p1+(256*p2)
  330 le = PEEK(adr+3): cm$ = "": e2$ = STR$(e2)
  340 FOR x = 0 TO le: cm$ = cm$+CHR$(PEEK(adr+3+x)): NEXT
  350 IF cm$ = "" THEN  cm$ = " "
  400 PRINT SPC(3-LEN(tk$));tk$;
  410 PRINT SPC(15-LEN(cm$));cm$;
  420 PRINT SPC(11-LEN(e2$));e2$;
  430 PRINT SPC(5);
  470 IF PEEK(p3) = 0 GOTO 540
  500 FOR x = 0 TO PEEK(p3)-1
  510 x1 = PEEK(p3+1+2*x): x2 = PEEK(p3+2+2*x)
  520 pm$ = STR$(x1+(256*x2))
  530 PRINT SPC(8-LEN(pm$));pm$;: NEXT
  540 PRINT
  900 adr = adr+le+4: IF adr < 817 GOTO 300
 1000 PR #0
10000 TEXT: PRINT " program terminated.": END
```

HACKER'S CONTEST #4

The NIBBLES & BITS Hacker's Contest is a bi-monthly competition. The winner of each contest is randomly selected from the correct responses postmarked within the specified dates. No individual shall be named the winner in three consecutive contests. The winner of each contest shall be awarded ten dollars and a free three month extension to his/her NIBBLES & BITS subscription term. Decisions of the judges are final.

Responses for this contest will be considered valid if, and only if, they are postmarked after September 30, 1986 and prior to December 1, 1986. The winner shall be announced in the December issue of NIBBLES & BITS.

Write a SmartBASIC program (it may include machine code in DATA statements), which will display all of the INVERSE fonts upside down (inverted) on the TEXT screen.

```
  10 REM HGR screen saver routine demonstration
 100 LOMEM :37888
 110 POKE 25431,11: POKE 25471,17: POKE 25568,27: HGR
 200 DATA 1,0,20,17,0,32,33,0,108,197,205,29,253,193
 210 DATA 17,0,0,33,0,128,205,29,253,201
 220 FOR x = 27600 TO 27623: READ ml: POKE x,ml: NEXT
 230 DATA 1,0,20,17,0,32,33,0,108,197,205,26,253,193
 240 DATA 17,0,0,33,0,128,205,26,253,201
 250 FOR x = 27624 TO 27647: READ ml: POKE x,ml: NEXT
 300 FOR x = 0 TO 15: POKE 18765+x,x: NEXT
 500 HCOLOR  = 7
 510 FOR x = 0 TO 254: HPLOT x,0 TO x,9
 520 HPLOT x,150 TO x,159: NEXT
 530 FOR x = 0 TO 160: HPLOT 4,x TO 13,x
 540 HPLOT 246,x TO 255,x: NEXT
 600 FOR x = 2 TO 15: HCOLOR  = x
 610 HPLOT 54,x+20 TO 205,x+20: NEXT
 700 pi = ATN(1)*4: rn = pi/180: HCOLOR  = 15
 710 ra = 20: y1 = 80: FOR x1 = 64 TO 192 STEP 64
 720 FOR point = 0 TO 2*pi STEP rn*4
 730 x2 = ra*SIN(point): y2 = ra*COS(point)
 740 HPLOT x1+x2,y1-y2: NEXT point: NEXT x1
 800 CALL 27600: REM save picture in RAM
1000 HOME: PRINT " 1 = erase picture": PRINT " 2 = draw picture"
1010 PRINT " 3 = exit program"
1020 VTAB 23: GET key$
1030 IF key$ = "1" THEN  HGR: GOTO 1000
1040 IF key$ = "2" THEN  CALL 27624: GOTO 1000
2000 TEXT: PRINT " program terminated.": END
```

```
  10 REM create underscored fonts
  20 REM changes inverse fonts
  30 REM TEXT, GR, or HGR resets
 100 LOMEM :28000
 110 DATA 62,255,17,1,0,33,0,0,205,38,253,201
 120 FOR x = 27600 TO 27611: READ ml: POKE x,ml: NEXT
 130 p1 = 27600: p2 = 27606: p3 = p2+1
 140 POKE 17126,PEEK(17115): REM INVERSE COLOR = NORMAL COLOR
 150 TEXT: PRINT " one moment please ..."
 200 FOR x = 128 TO 255: ad = x*8+7
 210 hi% = ad/256: lo% = ad-(hi%*256)
 220 POKE p2,lo%: POKE p3,hi%: CALL p1: NEXT x
 500 HOME: VTAB 2: HTAB 3: INVERSE
 510 PRINT "UNDERSCORED INVERSE FONTS": NORMAL
 520 VTAB 6: PRINT " This simple technique could "
 530 PRINT " add a nice touch to your "
 540 PRINT " programs!!!"
 550 VTAB 18: HTAB 2: FLASH
 560 PRINT "What do you think?": NORMAL
```

```
  10 REM create underscored fonts
  20 REM changes inverse fonts
  30 REM TEXT, GR, or HGR resets
  40 REM *** SUPER FAST VERSION
 100 LOMEM :28000
 110 DATA 1,128,0,17,1,0,33,255,3,175,185,200
 112 DATA 13,197,1,8,0,9,193,62,255,197,213,229
 114 DATA 205,38,253,225,209,193,24,233
 120 FOR x = 27600 TO 27631: READ ml: POKE x,ml: NEXT
 140 POKE 17126,PEEK(17115): REM INVERSE COLOR = NORMAL COLOR
 200 TEXT: CALL 27600
 500 HOME: VTAB 2: HTAB 3: INVERSE
 510 PRINT "UNDERSCORED INVERSE FONTS": NORMAL
 520 VTAB 6: PRINT " This simple technique could "
 530 PRINT " add a nice touch to your "
 540 PRINT " programs!!!"
 550 VTAB 18: HTAB 2: FLASH
 560 PRINT "What do you think?": NORMAL
```

```
   1 LOMEM :28000
  50 REM *** for SmartBASIC V1.0 ONLY !!! ***
 100 REM *** PatchWORK ***
 110 REM >>> simple BASIC enhancements and fixes
3999 REM *** true RND fix
4000 DATA 229,42,64,63,35,34,64,63,225,201
4010 FOR x = 172 TO 181: READ ml: POKE x,ml: NEXT
4020 POKE 171,0: POKE 11907,201
```

```
   1 LOMEM :28000
  50 REM *** for Intel-BEST 3.3 ONLY !!! ***
  60 REM >>> execute Intel-BEST first <<<
 100 REM *** PatchWORK 3.3 ***
 110 REM >>> simple BASIC enhancements and fixes
3999 REM *** true RND fix
4000 DATA 205,75,0,125,135,79,6,0,33,50,25,9
4010 DATA 126,198,128,35,102,111,34,1,0,205,10,0,201
4020 FOR x = 0 TO 24: READ ml: POKE 10164+x,ml: NEXT
4030 POKE 65484,180: POKE 65485,39
4500 DATA 229,42,64,63,35,34,64,63,225,201
4510 FOR x = 172 TO 181: READ ml: POKE x,ml: NEXT
4520 POKE 171,0: POKE 11907,201
```

TITLE (asmb#17): Locate Directory Entry

Line#:	Label:	Decimal value:	Op-code:	Comment:
1	SETUP	62, nn,	LD A, nn	; load drive code value
2		17, 235, 255,	LD DE, $FFEB	; set ASCII start of filename
3		33, 160, 253,	LD HL, $FDA0	; set location to place entry
4	CALLOS	255, 204, 252,	CALL $FCCC	; CALL EOS locate dir entry
5	ERROR	50, 255, 107,	LD $6BFF, A	; store error code
6	DONE	201	RET	; RETurn to BASIC

TITLE (asmb#18): Instant Underscored Fonts

Line#:	Label:	Decimal value:	Op-code:	Comment:
1	SETUP	1, 128, 0,	LD BC, $0080	; load number of fonts
2		17, 1, 0,	LD DE, $0001	; load number of bytes
3		33, 255, 3,	LD HL, $03FF	; load VRAM start address
4	LOOP	175,	XOR A	; reset accumulator to zero
5		185,	CP C	; check if font count is zero
6		200,	RET Z	; if zero, RETurn to BASIC
7		13,	DEC C	; decrement font count by one
8		197,	PUSH BC	; store current font count
9		1, 8, 0,	LD BC, $0008	; load displacement
10		9,	ADD HL, BC	; add displacement to VRAM addr
11		193,	POP BC	; retrieve current font count
12		62, 255,	LD A, $FF	; load bit mapped 'line'
13		197,	PUSH BC	; store current font count
14		213,	PUSH DE	; store byte count
15		229,	PUSH HL	; store current VRAM address
16	CALLOS	205, 38, 253	CALL $FD26	; CALL EOS write byte to VRAM
17		225,	POP HL	; retrieve current VRAM address
18		209,	POP DE	; retrieve byte count
19		193,	POP BC	; retrieve current font count
20	GOBACK	24, 233	JR $E9	; jump back to LOOP

TITLE (asmb#19): True RND Patch

Line#:	Label:	Decimal value:	Op-code:	Comment:
1	PATCH	229,	PUSH HL	; store current value
2		42, 64, 63,	LD HL, ($3F40)	; load current RND seed
3		35	INC HL	; increment seed value
4		34, 64, 63,	LD ($3F40), HL	; change RND seed
5		225,	POP HL	; retrieve previous value
6	DONE	201	RET	; RETurn from interrupt

TRUE RND PATCH

One of the shortcomings of SmartBASIC is the faulty RND function. With SmartBASIC V1.0 the sequence of random (???) numbers is the same every time a program is RUN. With SmartBASIC V2.0 the sequence of random numbers is the same every time BASIC is booted.

There are a few different ways to simulate true randomization from BASIC. However, it's much more effective to modify BASIC instead.

Addresses 16192 and 16193 point to the seed for the random number generator. We can have BASIC correct RND by continuously changing the seed value.

To do so, we've modified the interrupt routine from the video chip (RAM addresses 102 -171). This routine is an example of a hardware interrupt (mentioned in the September issue). The routine is executed 60 times per second regardless of BASIC operations. Please take a look at the two programs at the bottom of page 22.

Once you patch this routine, BASIC will truly generate random numbers. This can be a vital improvement for games and quizzes.

The Intel-BEST 3.3 version relocates the 'T1' command first because it occupies the addresses we need to use. The execution start is now at address 10164.

Asmb#19 (on page 23) details the assembly language of the patch. All it does is get the current seed value, increment it, and put the new value back. This way the seed actually changes 3600 per minute.

The BASIC program also makes another change. The NEW command CALLs a routine which resets program pointers. The routine spans from 11892 to 11919. The last portion of the routine resets the default RND seed. By POKEing a 201 into 11907, the routine RETurns before resetting the RND seed.

Each month, for a while, we'll be adding more patches to SmartBASIC (and Intel-BEST 3.3). The individual programs are numbered so that you can use them separately or merge them into one larger program of several patches.

With standard SmartBASIC, you may want to name the file of patches "HELLO" so that BASIC will automatically run the program. By the same logic, you may want to name to Intel-BEST 3.3 version "FIRST".

GETTING INTO CP/M 2.2

IN PERSPECTIVE

Last month we discussed an easy way to make a backup copy of CP/M. This is vital. Once you've made the backup, put the original away and work ONLY with the backup.

When working with any sophisticated piece of software you need to master the fundamentals first. As obviously simple as this sounds, most of us aspiring hackers too often allow ourselves to be overcome with eagerness. Admittedly, having a firm background in SmartBASIC and some experience with Z80 encoding will augment your understanding of CP/M. However, you'll find this powerful operating system to be most useful if you "conquer" it in a logical step by step process. For many of you, these first few articles will only be review. In due time, though, we'll delve into the more rewarding aspects, such as modifying CP/M and creating machine code programs.

THE BUILT-IN COMMANDS

The DIR command is used to display a list of the files on a specific storage medium. Simply typing DIR and pressing [RETURN] will display the filenames on the current drive. To see the filenames in another drive without changing the default drive, follow DIR with a space, the drive label (A, B, C, or D), and a colon. Then press [RETURN]. For example,

```
DIR B:  [RETURN]
```

will display the files in drive B. You can now type DIR (without a drive label) to see the directory for the default drive.

To change drives, simply enter the drive label, a colon, and then press [RETURN]. Now you can enter DIR or whatever command you prefer on the most recently selected drive.

If there are no files in the directory, you'll get the message "NO FILE". The DIR command does not reveal whether or not a data pack or disk has the operating system stored on it (as by dint of SYSGEN).

The DIR command can also be used to search for a specific filename. For example,

```
DIR B:EXAMPLE.TXT  [RETURN]
```

will search for the EXAMPLE.TXT file on drive B (without changing default drives). If the filename is not found, the "NO FILE" message will be displayed.

LOCAL ADAM USERS GROUPS

NORTH CAROLINA

Tri-Angle ADAM Users
Gary E. Hill
L-5 Oak Grove
Chapel Hill, NC 27514

Triangle ADAM Users
Paul Pappas
2623-A Yancyville Street
Greensboro, NC 27405-4407

OHIO

Mutual ADAM Users Group
Matt Esterak
412 Bettie Street
Akron, OH 44306

Lake Erie ADAM Users
Jonathan Fligner
2110 West 36th Street
Lorain, OH 44503

OREGON

Oregon ADAM Users
Craig Frerichs
1928 West Burnside, #309
Portland, OR 97209

PENNSYLVANIA

The (717) ADAM Users
Steve Chamberlain
120 East 4th Street
Bloomsburg, PA 17815

TENNESSEE

Midsouth ADAM Users
Roger Burford
Lot 142, NAS MHP
Millington, TN 38053

SWIFT POLL TALLY #1

From 1063 SWIFT POLL ballots (received 7-1 thru 9-30) we've computed the rankings listed below. In general, only about 25 software titles were named consistently. The totals within this elite group were fairly close.

You may be interested in how we attained these results. We assigned points for each position on a ballot. A program ranked in first place any ballot received 10 points, second place received 9 points, and so on. Then the points were tallied for each program. The rankings are, thus, directly proportional to the final score for each piece of software.

These results do not necessarily reflect the opinion of the editor or any staff member. Moreover, the results do not necessarily reflect the rankings on any single ballot.

1. ADAMCalc (c) Coleco
2. Jeopardy Coleco PD
3. CP/M 2.2 (c) Coleco/ (tm) DRI
4. VIDEOTUNES (c) FUTUREVISION
5. SmartFILER (c) Coleco
6. JKL Utilities (c) Overpriced Software
7. AUTOAID (c) FUTUREVISION
8. SmartTRIX (c) DATA DOCTOR/ (c) DEI
9. HACKER'S GUIDE I (c) Ben Hinkle
10. Intel-BEST 3.3 (c) DEI

As you can see, these "top ten" have each been around for at least a year. The only exception is our Intel-BEST. As proud as we are of this package, we must realistically admit that it is included in OUR poll because so many of OUR subscribers have purchased it. Another characteristic that probably is reflective of our readership is that a large portion of the named programs were programming utilities.

As you'll notice we've revised the SWIFT POLL BALLOT. This, hopefully, will encourage responses from a wider spectrum of available software titles.

Please note that you may submit THREE ballots for each three-month tally. At the time of each tally, a $25.00 cash prize winner shall be randomly selected from all ballots for that particular tally period.

Also, we've extended the valid time limit for each ballot. The deadline is now the last day of the third month of the tally period.

ADAM PRODUCT REVIEWS

PRODUCT:	HACKER'S GUIDE TO ADAM: VOLUME TWO
MANUFACTURER:	Ben Hinkle
MEDIA TYPE:	book (DDP/disk option)
GRAPHICS/SOUND/DESIGN:	95
INSTRUCTIONS:	97
USEFULLNESS vs PRICE:	99
RECOMMENDATION:	highly recommended
PRICE:	$13.00
RATED BY:	staff

Ben Hinkle, a tenth grade student, and his father, Peter, have released their third book. "THE HACKER'S GUIDE: VOLUME ONE" was primarily a continuation of their first book, "Info for ADAM Explorers". This latest guide for ADAM hackers is essentially independent of the previous books.

"THE HACKER'S GUIDE: VOLUME TWO" is a detailed guide to SmartBASIC. Indeed, this is the most elaborate explanation of SmartBASIC in any single publication. Every major and most minor routines are described. This book should be an indispensable aid in exploring the inner workings of the BASIC interpreter.

The first chapter gives an overview of how the interpreter works. The next nine chapters cover every address from 256 through 27406 -- in eye-opening detail. Chapter 11 itemizes more than a dozen changes to SmartBASIC. Some of these include 40 column text mode, formula substitution for GOTO and GOSUB, single keypress macros, and four sprite commands.

"THE HACKER'S GUIDE TO ADAM" is a book that you won't want to put down. And, once you've read it, you'll find that it is also a great reference material (all addresses are listed numerically). If you don't already have it, order it TODAY!

SPECIAL NOTE: The formula substitution fix for GOTO and GOSUB on page 79 has a couple of typos. Here is the corrected version.

```
10 DATA 0,0,0,205,3,39,68,77
20 FOR x=0 TO 7: READ d: POKE 8342+x,d: POKE 8437+x,d: NEXT
30 POKE 15756, 195: POKE 15757, 27: POKE 15758, 58
```

YES...ADAM

PRODUCT:	THE SPANISH VOCABULARIAN
MANUFACTURER:	MARATHON COMPUTER PRESS
MEDIA TYPE:	DDP/disk
GRAPHICS/SOUND/DESIGN:	93
INSTRUCTIONS:	93
USEFULLNESS vs PRICE:	95
RECOMMENDATION:	highly recommended
PRICE:	$18.50
RATED BY:	staff

This is the first commercial foreign language program for ADAM users. It is menu driven, uses SmartKEYs, uses color nicely, and has an electronic dictionary mode.

In drill and practice format you can quiz on any portion of the included 1600 word vocabulary. You can quiz on English to Spanish, or vice versa. And, utilities are included that allow you to create and alter vocabulary files. You can add up to 1700 more words. Whether you want to brush up on Spanish or study along with a scholastic curriculum, you'll find THE SPANISH VOCABULARIAN to be a very practical purchase.

PRODUCT:	The ADAM Resource DIRECTORY (2nd ed)
MANUFACTURER:	Keith Burrows
MEDIA TYPE:	book
GRAPHICS/SOUND/DESIGN:	97
INSTRUCTIONS:	N/A
USEFULLNESS vs PRICE:	95
RECOMMENDATION:	highly recommended
PRICE:	$14.95
RATED BY:	staff

This 1986 edition of the Resource DIRECTORY is 111 pages packed with reference material. Some features include spriral binding, appealing graphics, a programmer's tips section, and three times the information of the 1985 edition.

Hundreds of retailers are listed. Categories include computer magazines, ADAM publications, CP/M software info, ADAM users groups, list of ADAM users, list of scores of books, and modem usage tips.

Most of the questions that you already have today and those you'll formulate tomorrow concerning the expansion of your ADAM are answered in this single publication. This is an impressive effort at keeping ADAM alive that is well deserving of your support.

BULLETIN BOARD

ADAM RESOURCE DIRECTORY

New 1986 Edition
P.O. Box 90
Seelyville, IN 47878

Attn: ADAM OWNERS
"PROGRAM A MONTH"
Send SASE to WATT
2601-B Marietta Avenue
Kenner, LA 70062

Looking for local ADAMites
Mr. W.J. Lites
970 Palermo Road
Titusville, FL 32780

ADAM Software & Supplies
Marathon Computer Press
P.O. Box 68503
Virginia Beach, VA 23455

THE ADAM HACKER'S GUIDE:

Volumes One and Two
Ben Hinkle
117 Northview Road
Ithaca, NY 14850

ADAM Software
FUTUREVISION
P.O. Box 34
North Billerica, MA 01862

surplus parts for ADAM
H&R CORPORATION
401 East Erie Avenue
Philadelphia, PA 19134
(215) 426-1708

LARGE SELECTION OF ADAM PRODUCTS
M.W. RUTH COMPANY
510 Rhode Island Avenue
Cherry Hill, NJ 08002

PRODUCT LIST

DEI SOFTWARE

■■■ Intel-BEST 3.3
dynamic enhancement to SmartBASIC -- makes over 3 dozen changes

$24.95 STANDARD PRICE
$18.95 SUBSCRIBER DISCOUNT PRICE

■■■ Intel-LOAD
converts BASIC programs to LOAD up to 12 times faster -- stays in RAM plus onscreen help

$15.95 STANDARD PRICE
$11.95 SUBSCRIBER DISCOUNT PRICE

DEI HARDWARE SUPPLIES

■■■ DEI blank disks
Single-sided, double-density, reliable quality

$1.25 (each) or $11.95 (for 10) STANDARD PRICE
$1.19 (each) or $9.95 (for 10) SUBSCRIBER PRICE

■■■ DEI ADAM printer ribbons
just like the ones that came with your ADAM

$5.50 (each) or $15.50 (for 3) STANDARD PRICE
$4.95 (each) or $13.45 (for 3) SUBSCRIBER PRICE

ATTENTION: Until 12/15/86 we're GIVING a FREE blank, unformatted disk with any DEI purchase. BE SURE to mention on the order form that you'd like to take advantage of this gift offer!!!

DEI PAPER SUPPLIES

███ adhesive labels
white, tractor-feed, fan-fold, 3 1/2 x 15/16, single column

$2.95 (for 500) STANDARD PRICE
$2.25 (for 500) SUBSCRIBER DISCOUNT PRICE

$5.50 (for 1000) STANDARD PRICE
$3.95 (for 1000) SUBSCRIBER PRICE

███ blank white paper
tractor-feed, fan-fold, 9 1/2 x 11, 20# wt., clean edge, 250 sheets

$5.95 STANDARD PRICE
$5.45 SUBSCRIBER DISCOUNT PRICE

DEI EZ-REFERENCE GUIDES

███ EZ #101
approximately 700 NUMERIC Z80 instructions: decimal, hex, op codes, operands, 9 full-size pages (FREE shipping)

$2.50 (each) STANDARD PRICE
1.95 (each) SUBSCRIBER DISCOUNT PRICE

███ EZ #102
approximately 700 ALPHABETIC Z80 instructions: decimal, hex, op codes, operands, 9 full-size pages (FREE shipping)

$2.50 (each) STANDARD PRICE
1.95 (each) SUBSCRIBER DISCOUNT PRICE

DATA DOCTOR SOFTWARE

███ SmartBEST V1.0
the popular SmartBASIC enhancement

$18.95 STANDARD PRICE
$16.95 SUBSCRIBER DISCOUNT PRICE

███ SmartTRIX I
a set of 10 programmer utilities (including two extremely nice sprite design programs) and a 68 page manual

$34.95 STANDARD PRICE
$29.95 SUBSCRIBER DISCOUNT PRICE

███ STRATEGY STRAIN I
a set of 9 computer classics selected for their intellectual challenge (graphics, sound, SmartKEYS)

$24.95 STANDARD PRICE
$18.95 SUBSCRIBER DISCOUNT PRICE

███ QUIKFAX QUEST I
three academic quizzes (U.S. capitals, world capitals, elements of chemistry)

$24.95 STANDARD PRICE
$19.95 SUBSCRIBER DISCOUNT PRICE

COLECO PRODUCTS

(limited quantities)

███ SmartLOGO (DATAPAK ONLY)
Coleco's version of the popular language

$34.95 STANDARD PRICE
$27.95 SUBSCRIBER DISCOUNT PRICE

███ CP/M 2.2 (DATAPAK ONLY)
version of the popular operating system configured for ADAM

$34.95 STANDARD PRICE
$27.95 SUBSCRIBER DISCOUNT PRICE

███ SmartFILER (DISK ONLY)
Coleco's popular general purpose database

$19.95 STANDARD PRICE
$14.95 SUBSCRIBER DISCOUNT PRICE

███ DISK VERSIONS NOW AVAILABLE. Unless otherwise noted, all software is available on disk or data pack.

███ All DEI datapaks and disks are warrantied to be free from defects in material in workmanship. If the storage medium proves defective, return it to DEI for repair or replacement (at DEI's descretion).

██████ The prices listed above are effective 10/1/86 through 12/15/86

PRODUCT ORDER FORM

YOUR NAME: ______________________________

ADDRESS: ______________________________

CITY: ______________ STATE: ___ ZIP: _______

PHONE NUMBER: ____________________________

SUBSCRIPTION ID NUMBER: ____________________

< ITEM/QUANTITY/MEDIA > < PRICE >

<__________________________> $<___.__>

<__________________________> $<___.__>

<__________________________> $<___.__>

<__________________________> $<___.__>

<__________________________> $<___.__>

SUBTOTAL: $____.__

SHIPPING: ____.__ (inside contiguous USA: $2.50; elsewhere: $4.00)

TAX: ____.__ (WV residents only -- 5%)

OTHER: ____.__

OTHER: ____.__ (subscription/renewal)

TOTAL: $____.__

To order: complete this form, and send check or money order (US FUNDS) to:

DIGITAL EXPRESS, INC.
Route one, Box 29 - G
Oak Hill, WV 28086

SOFTWARE EXCHANGE

Our first four public domain libraries are completed. Each BASIC PD library contains over 70K of programs. Each library includes two instruction files which can be read or printed from SmartWriter. All BASIC programs are speed-RUN. Most of the programs are controlled from a "ramdisk" written primarily in machine code.

Each library is available on datapak or disk for $7.95.

To get a free copy of a specific library: (1) contribute an original program, (2) send a signed statement that the program is not copyrighted, (3) send the program on a data pack, (4) request the specific library that you want in return, and (5) include return postage and a mailer or $2.50 for shipping. "FAMILY COMPUTING" programs are not accepted.

The first four BASIC volumes are: B-1.0, B-2.0, B-3.0, and B-4.0. Below is list of the programs on the B-2.0 volume.

```
VOLUME TITLE:  N&B B-2.0            FREE BLOCKS:  163

BOOT        :S  1   DIRECTORY  :S  1   HELLO       :A  1   ml.obj      :H  3
GoHACKER    :H  2   HackerDISK :H  6   BASICPGM    :M  1   Sparkle     :H  2
Numbers     :H  2   IntDump    :H  1   Average     :H  1   ODDorEVEN   :H  1
AutoCenter  :H  1   TrueRND-1  :H  1   TrueRND-2   :H  1   TrueRND-3   :H  1
FLASHplay   :H  1   VolNameChg :H  2   INITrecov   :H  2   EOS-INIT    :H  2
Blocks      :H  1   Geometric  :H  2   TicTacADAM  :H  8   MathQuiz    :H  5
HiResPix-1  :H  2   DateCALC   :H  8   SoundDemo   :H  2   Etch&Draw   :H  3
DummyTurn   :H  1   Poker.BIG  :H 16   READ-1.WPR  :H  6   READ-2.WPR  :H  5
```

SWIFT POLL BALLOT

As a NIBBLES & BITS subscriber, you are invited to submit the following SWIFT POLL ballot. You may submit no more than THREE ballots during the tally period ending December 31, 1986. Valid entries must include your subscription ID number and may be duplicated, if you prefer.

To complete, just list your favorite software title in the categories of your choice. You may list different favorites on each ballot. The results of this particular tally period will be published in the January issue.

YOUR NAME: ______________________________ SUBSCRIPTION ID NUMBER: __________________

Your favorite COLECO title: ____________________________
Your favorite Public Domain title: ______________________

Copyrighted 3rd party titles:

Your favorite media/copy utility: ______________________
Your favorite game (cart, disk, or DDP): _______________
Your favorite BASIC enhancement: _______________________
Your favorite tutorial book: ___________________________
Your favorite CP/M software: ___________________________
Your favorite educational title: _______________________
Your favorite miscellaneous utility: ___________________
Your favorite miscellaneous title: _____________________

DIGITAL EXPRESS, INC.
Route one, Box 29-6
Oak Hill, WV 25901

--- NEW ADDRESS ---