

# NIBBLES & BITS

The comprehensive monthly  
newsletter for ADAM users



November 1986  
vol: 1, nmb: 5  
SINGLE ISSUE: \$3.50

THE N&B STAFF . . . . .	2
PUBLIC NOTICE . . . . .	2
DISCLAIMER. . . . .	2
EDITOR'S NOTE . . . . .	3
N&B NEWS. . . . .	3
ADAM NEWS . . . . .	4
EXPANDING YOUR SYSTEM . . . . .	5
ADAM USERS FORUM. . . . .	5
BIT BY BIT	
BASIC DECISIONS. . . . .	7
BASIC LOOPS. . . . .	7
BYTE-SIZED BASIC	
POKES TO PLAY WITH (part 5). . . . .	8
SmartBASIC 2.0 FEATURES. . . . .	8
NUMBER CONVERSION PROGRAMS . . . . .	9
BASIC ANIMATION. . . . .	9
HACKER'S DELIGHT	
HACKER'S NOTES . . . . .	12
DATA/REM EXTRA SPACE Patch . . . . .	12
DELETE to STATUS CHANGE. . . . .	12
New SmartBASIC V1.0 Bootstrap. . . . .	13
GETTING INTO CP/M 2.2	
BOOKS TO READ. . . . .	20
THE BUILT-IN COMMANDS (part 2) . . . . .	20
ADAM PRODUCT REVIEWS. . . . .	21
ADAM ACCESS . . . . .	25
BULLETIN BOARD. . . . .	27
LOCAL ADAM USERS GROUPS . . . . .	28
PRODUCT LIST. . . . .	28
PRODUCT ORDER FORM. . . . .	30
SOFTWARE EXCHANGE . . . . .	31
SWIFT POLL BALLOT . . . . .	31

This issue includes 8 SmartBASIC program LISTs and 7 assembly language lists.

COLECOVISION, ADAM, SmartBASIC, and SmartWriter are registered trademarks of Coleco Industries, Inc.  
CP/M 2.2 is a registered trademark of Digital Research, Incorporated.

NIBBLES & BITS is printed in the USA. Copyright (c) 1986 by DIGITAL EXPRESS, INC. All rights reserved.



## THE N&B STAFF

### FOUNDER:

Vernon L. Whitman, Sr.

### EDITOR-IN-CHIEF:

Dr. Solomon Swift

### DESIGN EDITOR:

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 month or for 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. Please include your subscription ID number on address change notifications.

## 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. Unless otherwise stated, all correspondence shall be considered as "open to public review".

## EDITOR'S NOTE

## N&amp;B NEWS

The holiday season is upon us again. It was Christmas 1983 when I got my first ADAM. Back then the ADAM sold for just under \$800. At first, I was extremely disappointed. There was absolutely no hardware available and virtually no software (worth mentioning). For more than a year, it collected dust in our storage room.

Then a collegiate acquaintance (who later founded DATA DOCTOR) began importuning me to give it another try. Reluctantly, after three months, I capitulated. Some of what I found was predictable.

The big league hardware and software manufacturers wouldn't even consider an out of production computer. And, Coleco refused, for more than two years, to release any technical information; and, they were inordinately slow in processing orders. Coleco had orphaned it. Then the industry leaders ignored it. Other computerists criticized it. The mainstream computer users continued to chase the latest razzle-dazzle kludgeboxes. Finally, Coleco tried to assassinate it.

Despite all this, the computer still thrived, due solely to the assiduous efforts of thousands of devoted ADAMites all around the world. Well over one hundred local users groups support ADAM throughout the USA, Canada, Australia, and the United Kingdom. Scores of software developers work around the clock developing new, exciting programs. The competition between a half dozen hardware manufacturers continues to bring you high quality, low cost, industry standard peripherals. And, several companies produce informative, ADAM specific newsletters.

Today, support for ADAM has never been stronger. There has never been, nor will there probably ever be again, a 64K computer as potentially powerful as the ADAM is. And, at the current list price of \$199.95 it is one of the best buys that you will ever come across. I, for one, am very proud to say: "Yes — I am an ADAM user".

In many households, mine included, the ADAM is used to some extent by every family member. So, as Christmas draws near, be sure to consider ADAM products as gifts. Expand your system; tap more deeply into its potential; and support your favorite ADAM companies.



Dr. Solomon Swift  
EDITOR-IN-CHIEF

→We have just completed Intel-LOAD V2.0. It is an exact parallel of Intel-LOAD V1.0, except that it designed specifically for SmartBASIC V2.0. Now you can speed load your V2.0 programs. The price is the same. The only limitation is that it only supports STD MEM programs (those not employing the extended memory). See our product list this month.

→Some of you have asked what happened to Intel FONTS (mentioned in the August issue). We've combined it into another, still inchoate, package ShowOFF III. The ShowOFF series is designed to do just that, show off your system.

→If you're new to NIBBLES & BITS, you may be surprised to learn that every single issue of our newsletter is written, designed, and printed with an ADAM computer. This particular issue was printed entirely with two new DEI packages, ShowOFF I and II.

→We've just released ShowOFF I: The Graphics Design Package. All the graphics in this issue were created and printed with this software. It includes the most sophisticated high resolution graphics design program ever developed for ADAM, SmartPAINT. With it, you can create, edit, save, and print graphics screens. And, you can even convert PaintMASTER picture files (Strategic Software) to be used by SmartPAINT (with an included program patch). And, it includes an advanced version of BlockPAINT (September issue). We're very excited about ShowOFF I. You will be too! See the ADAM ACCESS department in this issue for more details.

→Intel-LOAD V2.0 and ShowOFF I: The Graphics Design Package are completed and ready now! ShowOFF II: Word Processing Enhancements and ShowOFF III: BASIC Graphics Utilities should be ready for release around 1/1/87.

→We have selected the winner of HACKER'S CONTEST #3. This ADAM hacker received a \$10.00 prize and a three month extension to his NIBBLES & BITS subscription. The winning hacker is:

Randal Bondi of Allison Park, PA

## ADAM NEWS

→Here we go again ... another contest. This is the biggest one yet. The first place prize is \$100.00 plus a one year subscription extension. Second place gets \$25.00 plus a one year extension. And, third place gets a free one year extension. This contest begins immediately and valid entries **must** be postmarked prior to 3/1/87. The contest is for the best judged picture file designed with SmartPAINT (from ShowOFF I). Entries will be judged on creativity and attention to detail. All entries will be considered as public domain and the top five (decisions of the judges are final) pictures will be printed in the April issue. There is no limit to the number of entries you can submit and all entries must be submitted on data pack or disk. Get ShowOFF I today and tap into the fascinating realm of graphics design.

→We have re-organized our public domain libraries. The BASIC volumes are segregated into to four categories: utilities, games, graphics/sound, and math/finance. We've also lowered the price to \$5.95 each and increased the size (25 files or 100K). And, we've added some of Coleco's PD's (these are not offered in exchange for BASIC programs), including ADAMLINK II. In the near future we'll start a PD library for CP/M. As a final note, we have now released BlockPAINT version 1.7 into the public domain. You may want to contribute it to another company's public domain library.

→When Luke founded DEI, he had planned to switch to bulk rate postage for NIBBLES & BITS beginning with the 1/1/87 issue. The current administration is adamantly opposed to bulk rate mailing. We feel that this lower class postage would demean the newsletter and be unsatisfactory to most of our subscribers. Consequently, we're forced to go up on the annual subscription rate (to maintain FIRST CLASS delivery throughout the USA). The ONLY rate that's changing is the standard annual (12 issue) subscription. Effective 1/1/87 the annual rate will be \$22.00. If you prefer, you may renew before that date at the current rate of \$18.00. We sincerely hope that we don't lose any of you when it does come time to renew. Thank you, in advance, for understanding.

→REEDY SOFTWARE is continuing their trend toward high quality, low cost games for ADAM. The ENTERTAINMENT PACK 1 and Mage Quest (reviewed in this issue) are two of the most professionally appealing BASIC intellectual games ever written for the ADAM (or any other computer for that matter).

→Another creator of excellent BASIC games is Bob Tarnowski of Mr. T Software. See our review in this issue of his superb, multi-player TriviaPac I.

→Some of the local ADAM users groups produce their own newsletters. Most of these are monthly and contain information both for their local members (meeting times, etc.) and ADAM users at large. Many of these newsletters are available nationally for a small fee. Each of the groups listed below offer such publications. Please take a couple of minutes to write for more information to these worthy ADAM support groups.

gHAAUG  
Terry Fowler  
Route 2, Box 2756  
Pearland, TX 77581

KAUG  
David E. Carmichael  
1325 North Meridian, Apt. 201  
Wichata, KS 67203-4637

The Paper Peripheral  
Paul Pappas  
2623-A Yanceyville Street  
Greensboro, NC 27405-4407

IEAUG  
Wendy L. Ball  
P.O. Box 0  
Rialto, CA 92376

→We would like to publicly thank Terry Fowler of gHAAUG for his munificent aid in expanding our public domain volumes. Thanks also for the positive review of NIBBLES & BITS in the August issue of your newsletter.

→ → → HAPPY THANKSGIVING ← ← ←

## EXPANDING YOUR SYSTEM

### PRINTER ALTERNATIVES

By far the most frequently added non-Coleco peripherals to ADAM are a second printer and a 64K expander. Over the next 3 or 4 months we'll take a look at some of the available printer options.

Three items are required to make this expansion: the interface, the cable, and, of course, the printer. Let's discuss interfaces first.

An interface is an electronic device which permits the Central Processing Unit (the computer brain, ie, the Z80 on ADAM) to communicate with peripherals. All communication with the CPU is handled via an internal component called the bus, ie, ADAMNET. The keyboard, ADAM printer, video chip, etc. are each connected to this bus. ADAM includes five additional ADAMNET connectors. The disk drive modular plug on the left side of the memory console is interchangeable with the keyboard plug on the front. Under the memory console cover are three bus expansion slots. And, on the right side of the console, under the lift-up door, is a bus extender.

All of ADAMNET's connectors are nonstandard. Many companies employ this technique in the hopes that you'll be more inclined to purchase their own peripherals. Third party developed interfaces simply allow you to convert from nonstandard to the much more popular connectors so that you have a wide range of peripherals to choose from.

The two most common bus interfaces are the RS232 serial interface and the Centronics parallel interface. The RS232 is most frequently employed for telecommunications (via modem). And, the parallel interface is generally used by dot matrix printers.

Both Orphanware and Eve Electronics offer these standard interfaces. The quality from both of these staunch ADAM supporters is very high and their prices are reasonable. We've listed the addresses of each company in previous issues.

Essentially, there are five different types of computer printers. Daisywheel printers, like the ADAM printer, are used to produce true letter quality documents. The remaining four classes of printers each use a matrix of dots to generate near letter quality results. Impact dot matrix printers, like we use, print by means of tiny pins on the printing head. Centronics parallel interfaced impact dot matrix printers are, by far, the most popular genre. Their combination of fast print speed and low cost is, indeed, nonpareil.

Thermal printers work by passing a heated plate across a special paper. Ink jet printers squirt tiny (typically about 50 microns) droplets of ink on the paper. Laser printers, which resemble photocopiers in many respects, pinpoint microscopic beams of coherent light to construct the patterns of dots.

## 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.

### WHERE TO GET DISK DRIVES

I recently became interested in purchasing a disk drive for my ADAM computer, and have found them to be a very scarce item. If you are aware of where one might still be purchased, please let me know.

Scott D. Pellingier  
39 Micahill Lane  
Levittown, PA 19056

**IN RESPONSE:** Coleco disk drives are, indeed, scarce. Orphanware and NIAD try to keep some in stock (addresses listed in previous issues). Also, Don Perlman usually has some on hand. He also does disk drive repairs. His address is:

620 Harmon Cove  
Secaucus, NJ 07094

MACHINE LANGUAGE MODE

How do I get ADAM to go into the machine language or assembly language mode? Do I need additional software?

Joseph M. Quinn  
6665 Timbers Drive  
Mobile, AL 36609

**IN RESPONSE:** The Apple II series computers include a **monitor mode** which permits direct programming of machine code in hex format. ADAM, however, doesn't include this feature. We include assembly language lists primarily for educational purposes. It's much easier to think of machine code in terms of mnemonics rather than just a collection of numbers. The BASIC programs in the HACKER'S DELIGHT department include the machine code in DATA statements so that you can make practical use of the assembly language.

CP/M FORMATTING

When I purchased my ADAM I also got CP/M 2.2. I have since realized that for someone who mainly uses the computer for entertainment and family oriented programs, CP/M 2.2 is not needed. I have several data packs which I converted to the CP/M format. I am now unable to use them in SmartBASIC and am unable to reformat or initialize them in SmartBASIC. How can I reformat a data pack to standard EDS format?

Lawrence P. Zoia  
637 Spring Street  
East Bridgewater, MA 02333-1804

**IN RESPONSE:** The easiest way to thoroughly reformat a data pack is to use one of the popular media copy utilities. Use a blank DDP as the source medium and the CP/M formatted DDP as the destination. Copy all 255 blocks. Then CATALOG each medium.

ADAMCalc WINDOWING

Using the windowing feature of ADAMCalc lets you see up to six different parts of the spreadsheet at one time. Is there any way to print it out that way?

Stan von Helms  
Route 3, Box 161-J  
Hudson, WI 54016

**IN RESPONSE:** Regrettably, we do not know of a method of accomplishing this directly from ADAMCalc. If anyone reading this message has discovered a technique for this, please let us know.

BOOTING BASIC FROM DISK

I would like to know if there is a way to change BASIC on disk so that it will RUN the HELLO program as a turnkey on the BASIC disk.

Fitzroy E. Ryan  
Box 1098  
New York, NY 10156

**SPECIAL RESPONSE:** For the benefit of all readers, we've modified our written response. Just after SmartBASIC is loaded into RAM, it performs a search for a turnkey program. This program, which must be named HELLO, should therefore automatically RUN when it is on the SmartBASIC tape or disk. However, this is one of SmartBASIC's more annoying bugs. Even if you backup BASIC to disk, it still searches for the HELLO program in the first tape drive. Address 16641 in SmartBASIC controls the drive that the HELLO search is performed on. There are many programs that allow you to change the value at that address on the actual tape or disk. However, this is merely a trade off. A much more useful alternative is to actually correct SmartBASIC.

With regard to Mr. Ryan's request, this is precisely what we've done. A rather significant portion of the HACKER'S DELIGHT this month involves the development of a new bootstrap routine for SmartBASIC V1.0. With this routine the HELLO search will be performed on whatever drive BASIC is booted from.



**BIT BY BIT****BASIC LOOPS****BASIC DECISIONS**

Computers are particularly useful in performing repetitive tasks and making logical decisions. The **IF ... THEN** command is used by BASIC to make decisions within a program. The format is:

**IF** condition **THEN** do something

The condition is usually an arithmetic equation. If the condition exists, it is said to be **TRUE**. If the condition does not exist, it is said to be **FALSE**. When the condition is true, the part of the statement following **THEN** is executed. Consider this example to see how it works.

```
10 INPUT "Enter any number? "; nu
20 IF nu < 10 THEN PRINT "It's less than 10."
30 IF nu > 10 THEN PRINT "It's greater than 10."
40 IF nu = 10 THEN END
50 PRINT: GOTO 10
```

If you enter the number "10", the program will **END**. Any other number will cause a message to be displayed, then the program will repeat itself. Line #50 makes the repetition happen. The **PRINT** statement without a parameter simply prints a blank line. Do you see how **IF** can be used to make a decision?

Did you notice that line #50 contained two separate **BASIC** commands? You can chain several commands together on a single program line (to save memory) in this manner provided the commands are separated by colons (:). However, commands following an **IF ... THEN** statement (on the same line) will be executed **ONLY** if the condition is **TRUE**.

One of the more common uses of **IF ... THEN** is to cause the program to branch to a different line number based on the condition. Next month we'll include a **BASIC LIST** of an arithmetic quiz that reveals the benefit of this technique.

A loop is simply an operation that repeats itself. Usually this repetition continues until some certain condition is met. Consider this example.

```
10 PRINT "HELLO";
20 GOTO 10
```

In this simple program, the screen will fill with "HELLOs". The loop will continue until you press **CNTRL-C**. This particular type is referred to as an **endless loop**.

Here's another type of loop. This one is a parallel of the way loops are accomplished in machine code.

```
10 x = 0
20 x = x + 1
30 IF x < 3000 THEN GOTO 20
40 PRINT "loop done."
```

**BASIC** includes the **FOR ... NEXT** command to facilitate usage of loops. The following program is an exact parallel of the one above.

```
10 FOR x = 1 TO 3000: NEXT
20 PRINT "loop done."
```

Loops are typically used for one of two purposes: **forced program delays** or **repetitive calculations**. The two examples above caused program delays. This type of pause is used most often in music (for tempo, durations, etc.).

You can implement these forced delays with reasonable precision (even in **BASIC**). On **ADAM** a loop that counts from "1" to "750" lasts one second. One that counts from "1" to "1500" lasts two seconds, and so on.



## **BYTE-SIZED BASIC**

### POKES TO PLAY WITH

The LIST command:

Four addresses control the extra spaces added by the BASIC interpreter. The LIST command reveals these extra spaces. The default value of each address is "32".

Address 13357 adds the extra space after a semi-colon (;). If you POKE a zero into 13357 you can eliminate this unnecessary space.

Address 13349 adds the extra space after a comma (,). If you POKE a zero into 13349 you can eliminate this unnecessary space.

Address 13423 controls the extra space after BASIC commands. Ultimately, this is the culprit that causes the dreaded DATA and REM space bump.

Address 16148 controls the space used to format line numbers. If you POKE a zero into this address, the LIST will look exactly like the SmartWriter version of your program. Be sure to POKE a "32" back into the address before entering more program lines.

#### DATA and REM space fix:

Whenever you save a program or edit a program line containing a DATA or REM statement, BASIC adds an extra space after the command. It doesn't take much to push your DATA and REMarks into oblivion. There are at least three ways to correct this problem.

Intel-BEST 3.3 corrects the bug by patching a machine language routine over address 13423 (mentioned above). This routine checks each command to see if it is either of these two. If not, it adds a space.

One alternative is to change the interpreter's parameter check routine for DATA and REM. This routine, which also handles all quotes ("), occupies addresses 15814 through 15901. Ben Hinkle (The Hacker's Guide To ADAM), was the first to reveal four POKES to this routine which eliminate the dreaded extra space.

The POKES are as follows:

POKE 15824, 216  
POKE 15830, 8  
POKE 15831, 55  
POKE 15832, 19

This fix is incompatible with Intel-BEST 3.3. The two patches combine to eliminate ALL spaces after DATA and REM.

Both of these fixes do have a limitation. They only correct programs entered after the patch. If you have SAVED a program that already has several extra spaces added, you have to manually delete them from each program line. In the HACKER'S DELIGHT department this month, we have a patch that corrects the problem thoroughly.

### SMARTBASIC 2.0 FEATURES

Enhanced features:

Probably one of the most coveted features of this newer version of the interpreter is its use of the 64K RAM expander in BASIC. Two new BASIC OS commands have been added. **EXTMEM** allows you to access the memory card. **STDMEM** takes you back to standard RAM. Each of these commands read information from the BASIC medium, thus you **must** have it in the current drive before executing either command.

Another new BASIC OS command is **MERGE**. This one allows you to chain programs together in RAM. Its use is a parallel of the LOAD command.

Version 2.0 also handles program lines with errors differently. The interpreter inserts a new command at the beginning of any program line that contains a syntax error.

This new command is a frowning face and a bell, ie, CHR\$(2) and CHR\$(7). The command is a parallel of the standard REM command. To remove the command and correct the error, you need to erase the command in editing.

SmartBASIC 2.0 also corrects the DATA and REM space bump bug. We'll reveal more of the new interpreter's features next month.



## Version 2.0 POKES:

SmartBASIC 2.0 uses less memory for the interpreter (447 bytes) because more machine code subroutines are employed. This is particularly true of the graphics mode routines. The following screen color POKES use the same master color code values as version 1.0 (mentioned in our July issue).

17184 (TEXT background)  
17240 (TEXT NORMAL)  
17251 (TEXT INVERSE)

16783 (HGR window)  
24784 (GR window)

24695 (GR and HGR background)  
24847 (GR and HGR TEXT)

Address 153 controls the FLASH speed. In version 1.0 address 159 does this.

Address 17437 controls the cursor blink rate. In version 1.0 address 17291 does this.

Address 1628 contains the current SPEED value. The version 1.0 equivalent is address 16129.

NUMBER CONVERSION PROGRAMS

The two programs on the next page allow you to easily convert between numbering systems. The program on the top of page 10 will display the equivalent in bases 2 through 16 of a decimal number (0 - 65535).

Just type the program and enter a decimal number at the prompt. Instantly it will display each equivalent in formatted fashion.

The decimal system, the one we commonly use, is a base 10 system. The base 2 system is referred to as the binary system. The base 8 system is called the octal system.

Alphabetic nomenclature is used to represent digits greater than nine for bases larger than the standard decimal. The most common of these is the hexadecimal, base 16, system.

Line numbers 310 through 420 calculate each digit of the equivalents. Line numbers 430 through 480 determine and PRINT each formatted number.

The program on the bottom of page 10 allows you to convert between standard decimal numbers and their two-byte machine code equivalents. Line numbers 210 through 220 calculate the decimal to high/low byte conversion. Line number 340 calculates and PRINTs the high/low byte to decimal conversion.

BASIC ANIMATION

The program LISTed on page 11 is a simple application of low resolution graphics animation. The game, towers of hanoi, is a computer version, of an ancient logic game.

The game starts with 5 different sized disks on the left of three pegs. The object is to move all of the disks to the rightmost peg. You can only move one disk at a time and you can only place smaller disks atop larger ones. The least number of possible moves is 31.

Line #100 sets the GR screen colors. Line #110 corrects the GR color table. Line numbers 200 through 230 draw the pegs and initial disk set up.

Line numbers 300 through 340 draw the numbers above each peg. Line numbers 400 through 540 constitute the "move from" module. Line numbers 600 through 780 control the "move to" aspect. The sound routine occupies line numbers 900 through 920.

The program uses the double dimensioned peg variable to keep track of disk locations.

peg(1,x) for the left peg  
peg(2,x) for the middle peg  
peg(3,x) for the right peg

The  $\times$  element signifies the positions (1 - 5) on a peg. For example, peg(1,3) would indicate that a disk occupies the middle position on the left disk. IF the value of peg(1,3) is a one, then the disk at that position is the smallest disk.

```
10 REM decimal to base convertor
100 DIM sym$(16),dig(16),pow(16)
110 DATA 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F
120 FOR x = 0 TO 15: READ sym$(x): sp$ = sp$+" ": NEXT
130 POKE 17059,5: POKE 17115,23
200 TEXT: INVERSE: VTAB 2: HTAB 2
210 PRINT " Decimal to Base Convertor ": NORMAL
220 VTAB 6: INPUT " enter decimal number: ";dn$
230 dn = INT(VAL(dn$)): IF dn < 0 OR dn > 65535 GOTO 220
300 HOME: VTAB 2: PRINT " decimal (base 10) = ";dn
310 VTAB 4: FOR ba = 2 TO 16: PRINT " base ";ba;": ";
320 IF ba < 10 THEN PRINT " ";
330 d2 = dn: conv$ = "": count = 0
340 FOR x = 0 TO 15: pow(x) = ba^x: NEXT x
400 FOR x = 15 TO 0 STEP -1
410 dig(x) = INT(d2/pow(x)): d2 = d2-(dig(x)*pow(x))
420 conv$ = conv$+sym$(dig(x)): NEXT x: IF ba = 2 GOTO 460
430 FOR y = 1 TO 15: zero$ = MID$(conv$,y,1)
440 IF zero$ <> "0" GOTO 460
450 count = count+1: NEXT y
460 IF count = 0 THEN done$ = conv$: GOTO 480
470 done$ = LEFT$(sp$,count)+RIGHT$(conv$,16-count)
480 PRINT done$: NEXT ba
500 VTAB 20: PRINT " again (Y or N)? ";
510 GET ag$: IF ag$ = "y" OR ag$ = "Y" GOTO 200
520 TEXT: PRINT " program terminated.": END
```

```
10 REM decimal/integer byte convertor
100 TEXT: PRINT " 1 = decimal to integer bytes"
110 PRINT " 2 = integer bytes to decimal"
120 PRINT " 3 = exit program": VTAB 5: PRINT " ?";
130 GET key$: key% = VAL(key$)
140 IF key% < 1 OR key% > 3 GOTO 130
150 TEXT: ON key% GOTO 200,300,160
160 PRINT " program terminated.": END
200 VTAB 2: INPUT " enter decimal number: ";dec$
210 dec = INT(VAL(dec$)): IF dec < 0 OR dec > 65535 GOTO 200
220 hi% = dec/256: lo% = dec-(hi%*256)
230 VTAB 6: PRINT " low order byte = ";lo%
240 PRINT " high order byte = ";hi%: GOTO 400
300 VTAB 2: INPUT " enter low order byte: ";lo$
310 lo% = VAL(lo$): IF lo% < 0 OR lo% > 255 GOTO 300
320 VTAB 4: INPUT " enter high order byte: ";hi$
330 hi% = VAL(hi$): IF hi% < 0 OR hi% > 255 GOTO 320
340 VTAB 10: PRINT " decimal equivalent = ";hi%*256+lo%
400 VTAB 20: HTAB 2: PRINT "again (Y or N)? ";
410 GET ag$: ON ag$ = "y" OR ag$ = "Y" GOTO 100: TEXT: GOTO 160
```

```

10 REM towers of hanoi
100 POKE 18607,11: POKE 18633,17: POKE 18711,27
110 GR: FOR x = 0 TO 15: POKE 18781+x,x: NEXT
200 COLOR = 7: FOR x = 29 TO 31: HLINE 2,36 AT x: NEXT
210 FOR x = 8 TO 30 STEP 11: VLINE 18,29 AT x: NEXT
220 COLOR = 6: FOR x = 1 TO 5: HLINE 8-x,8+x AT 2*x+18
230 peg(1,x) = x: NEXT: move = 1
300 COLOR = 3: VLINE 10,14 AT 8
310 FOR x = 10 TO 14 STEP 2: HLINE 18,20 AT x
320 HLINE 29,31 AT x: NEXT
330 VLINE 10,12 AT 20: VLINE 12,14 AT 18
340 VLINE 10,14 AT 31
400 PRINT " #";move;" : "
410 INPUT " move disk from which peg? ";mf$
420 mf = INT(VAL(mf$)): IF mf < 1 OR mf > 3 GOTO 400
500 offset = 0: FOR x = 1 TO 5: IF peg(mf,x) = 0 GOTO 540
510 offset = peg(mf,x): center = 11*mf-3: peg(mf,x) = 0
520 COLOR = 1: HLINE center-offset,center+offset AT 2*x+18
530 COLOR = 7: PLOT center,2*x+18: z = x: GOTO 600
540 NEXT x: PRINT " no disk on peg #";mf: GOTO 400
600 INPUT " move disk to which peg? ";mt$
610 mt = INT(VAL(mt$)): IF mt < 1 OR mt > 3 GOTO 600
700 check = 0: FOR x = 5 TO 1 STEP -1
710 IF check = 0 AND peg(mt,x) = 0 THEN check = x
720 NEXT x: IF check = 5 GOTO 750
725 IF peg(mt,check+1) >= offset GOTO 750
730 PRINT " only smaller disks on larger!": COLOR = 6
735 peg(mf,z) = z
740 HLINE center-offset,center+offset AT 2*z+18: GOTO 400
750 COLOR = 6: middle = 11*mt-3
760 HLINE middle-offset,middle+offset AT 2*check+18
770 peg(mt,check) = offset: IF mt = 3 AND check = 1 GOTO 800
780 move = move+1: GOSUB 900: GOTO 400
800 HOME: PRINT " you solved the puzzle in "
810 PRINT " ";move;" moves!!!"
820 FOR y = 1 TO 5: GOSUB 900: NEXT: END
900 POKE 17963,2: FOR x = 15 TO 1 STEP -1
910 POKE 17954,x: PRINT CHR$(7);
920 NEXT: POKE 17954,17: POKE 17963,7: RETURN

```



## HACKER'S DELIGHT

### HACKER'S NOTES

One of the vital steps in learning to develop machine code programs is studying pertinent applications. This month's HACKER programs are salient in two respects. They are involved specifically with improving SmartBASIC. And, our new BASIC bootstrap routine is the most extensive Z80 program we've listed thus far.

If you are annoyed by a particular aspect of BASIC, let us know. If there is some aspect that you'd like to see patched, we'd like to give it a try.

Although our current concentration is on improving BASIC, we still have several other projects underway. In upcoming issues, we'll elaborate on font and sprite design, sprite usage, and shape table theory. Also, we'll have explanatory articles on creating and using a ramdisk in BASIC. And, we'll develop DiskDOCTOR, our machine code utility.

### DATA/REM EXTRA SPACE PATCH

The two programs on the top of page 14 reveal a true fix for the troublesome DATA and REM space bump bug of SmartBASIC V1.0. The Intel-BEST 3.3 version also bypasses that enhancement's own fix with line numbers 5500 and 5510. The assembly language of each program is explained in **asmb#20** at the top of page 17.

Line number 5030 in each program alters the start of the REM and DATA parameter check routine. Now the routine starts by CALLing our new patch. Line number 5040 corrects the REM entry into the routine.

**Asmb#20** details the patch. The patch is entered at the second byte, label ENTRY. The DE register pair points to the current position in the tokenized version of the program line. If the byte pointed to by DE is a space, the patch skips it and checks the next byte. It continues in this manner until another ASCII value is found or the program line ends.

The greatest benefit of using this fix over others is that it eliminates ALL extra spaces between the command and actual parameter. After executing the patch, try to enter a DATA statement with a lot of extra spaces. The patch automatically eliminates them.

### DELETE to STATUS Change

Last month we presented a program that allowed you to change the attribute byte of filenames in the directory. We didn't reveal one of the program's best features. You can actually use it to recover DELETED filenames.

The EOS doesn't truly delete a file; it merely changes the attribute. Later, though, a newer file that uses the same or a smaller number of blocks for file storage can preempt the DELETED filename's 26-byte directory slot.

This month we've modified the DELETE command for changing attributes directly from BASIC. To do so, we've altered both the BASIC interpreter and a portion of the EOS.

The EOS includes a routine specifically for deleting file entries. It occupies addresses 61774 through 61818. Our patch erases part of the routine.

This is a dramatic change to the delete file function. Line 6030 of the program allows you to recover deleted files by changing the attribute. Line numbers 6040 and 6050 change the spelling of the BASIC command to STATUS. This is to prevent the possibility of confusion.

Now when you STATUS a filename (not DELETE), the value at address 61794 becomes that file's attribute. It doesn't matter if the file is LOCKed, UNLOCKed, or even DELETED.

When BASIC first creates a file, it is given an attribute of 16. Normally, when BASIC DELETes a file, it is given an attribute of 20 (16 + 4). Files LOCKed from BASIC have an attribute of 208 (16 + 128 + 64).

Suppose you want to set the attribute of a filename so that it is LOCKed and UNLOCK protected. Here's all you need to do.

Determine the attribute value:

128 + 16 + 2  
or, 146

Then POKE 61794, 146

Now, STATUS (filename)

That's all there is to it. Generally, you'll find this more convenient than loading a BASIC or machine code utility to perform the task.

Line 6000 of the SmartBASIC V1.0 version changes the POKE limit. Now you can POKE anywhere in RAM. Intel-BEST 3.3 automatically does this for you.

### New SmartBASIC V1.0 Bootstrap

The program on pages 15 and 16 is more than a mere patch. In essence, it is a bona fide (though simple) machine code program. This particular example demonstrates several aspects of Z80/EOS programming. And, it reveals how to create a machine code program in BASIC.

When you pull the computer reset switch, ADAM performs several housekeeping routines. The last of these is a CALL to address 64560 in the EOS. This is the **boot system** routine. The routine searches for a medium in each active drive. If none is found it reverts to SmartWriter. The search begins with the first disk drive, then the second disk drive, then the first tape drive, and finally to the second tape drive.

If a medium is found in any drive, the search stops and block number zero of that medium is read into a 1K buffer starting at address 51200 (\$C800). Then, that address is jumped to. On most BASIC media this block starts with a three byte jump to SmartWriter.

This new bootstrap program has three advantages over the original. First, it provides an immediate screen message (while BASIC is being loaded). Also, this bootstrap includes an error message with options to retry the boot or default to SmartWriter. And, the program will set the current drive as the default. This means that you can use the turnkey (HELLO program) feature on the medium regardless of the drive that it's booted from.

This program includes a lot of numerical DATA. You should take great care in keying in this information. Even one incorrect value could destroy a medium. We recommend that you use the program only on a BASIC backup (not the original). Also, to test the error routine, you may want to transfer the bootstrap to a medium that doesn't contain the SmartBASIC program (BASICPGM). Finally, be sure to store this BASIC program (from pages 15 and 16) before you RUN it.

The program consists essentially of four segments. Both the main (Z80) program and the error routine consist entirely of decimal machine code. The segment to convert the strings to ASCII follows. And, finally is the transfer module. You may omit line numbers 5900 and 10900. We used these lines while testing the program.

Asmb# 21:

The main Z80 program begins by setting the stack pointer. The EOS routine at 64560 Loads the current drive number into register B. Our program Loads B into A and then stores that value at RAM address 65535 (255 \* 256 + 255).

Asmb# 22:

This assembly language list describes the sound routine from lines 5300 and 5310 in the BASIC program. This routine uses the B register both as a counter and as the current note to be played. Line numbers 8 through 12 implement a programmed delay (to hold each note briefly). Line number 13 decrements the value in register B. Then line number 14 decrements B again and repeats the LOOP. The sound routine plays notes 20, 18, 16, 14, etc. A signed displacement of 234 is the same as 22 bytes backwards (256 - 22 = 234).

```
1 LOMEM :28000
50 REM *** for SmartBASIC V1.0 ONLY !!! ***
100 REM *** PatchWORK ***
110 REM >>> simple BASIC enhancements and fixes
4999 REM *** DATA/REM space fix
5000 DATA 19,26,183,200,254,32,40,248,183,201
5010 FOR x = 6346 TO 6355: READ m1: POKE x,m1: NEXT
5020 DATA 0,205,203,24
5030 FOR x = 15815 TO 15818: READ m1: POKE x,m1: NEXT
5040 POKE 335,188
```

```
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
4999 REM *** DATA/REM space fix
5000 DATA 19,26,183,200,254,32,40,248,183,201
5010 FOR x = 6346 TO 6355: READ m1: POKE x,m1: NEXT
5020 DATA 0,205,203,24
5030 FOR x = 15815 TO 15818: READ m1: POKE x,m1: NEXT
5040 POKE 335,188
5500 DATA 62,32,205,218,46
5510 FOR x = 13422 TO 13426: READ m1: POKE x,m1: NEXT
```

```
1 LOMEM :28000
50 REM *** for SmartBASIC V1.0 ONLY !!! ***
100 REM *** PatchWORK ***
110 REM >>> simple BASIC enhancements and fixes
5999 REM *** DELETE to STATUS change
6000 POKE 16149,255: POKE 16150,255
6010 FOR x = 61786 TO 61792: POKE x,0: NEXT
6020 POKE 61793,62: POKE 61794,20
6030 POKE 61412,0: POKE 61413,0
6040 nw$ = "STATUS": FOR x = 1 TO LEN(nw$)
6050 POKE x+20192,ASC(MID$(nw$,x,1)): NEXT
```

```
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
5999 REM *** DELETE to STATUS change
6010 FOR x = 61786 TO 61792: POKE x,0: NEXT
6020 POKE 61793,62: POKE 61794,20
6030 POKE 61412,0: POKE 61413,0
6040 nw$ = "STATUS": FOR x = 1 TO LEN(nw$)
6050 POKE x+20192,ASC(MID$(nw$,x,1)): NEXT
```



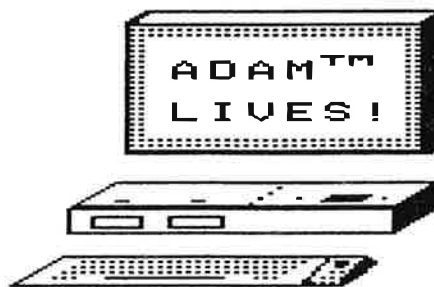
```
5 LOMEM :30000
10 REM new bootstrap for SmartBASIC
20 REM Use ONLY on a backup!!
30 REM Do NOT write this routine to your original SmartBASIC!!
100 REM block write routine
110 DATA 62,4,17,0,0,1,0,0,33,0,108,205,246,252,201
120 FOR x = 27600 TO 27614: READ m1: POKE x,m1: NEXT
300 HOME: PRINT " one moment please..."
499 REM clear bootstrap buffer
500 FOR x = 0 TO 1023: POKE 27648+x,0: NEXT
4999 REM main program
5000 DATA 49,128,209,120,50,255,255
5010 DATA 1,0,0,205,32,253
5020 DATA 1,224,1,205,32,253
5030 DATA 33,0,0,62,3,205,41,253
5040 DATA 33,0,8,62,2,205,41,253
5050 DATA 205,56,253,17,0,4,33,0,0,1,128,0,205,23,253
5060 DATA 33,0,11,62,4,205,41,253
5070 DATA 62,25,17,16,0,33,0,11,205,38,253
5080 DATA 62,244,17,16,0,33,16,11,205,38,253
5090 DATA 62,32,17,0,3,33,0,8,205,38,253
5100 DATA 1,6,7,205,32,253
5110 DATA 17,39,8,33,0,203,1,17,0,205,26,253
5120 DATA 17,98,8,33,17,203,1,28,0,205,26,253
5130 DATA 17,170,8,33,45,203,1,12,0,205,26,253
5140 DATA 17,227,8,33,57,203,1,26,0,205,26,253
5150 DATA 17,4,9,33,83,203,1,24,0,205,26,253
5160 DATA 17,38,10,33,107,203,1,20,0,205,26,253
5300 DATA 6,20,62,128,211,224,120,211,224,62,146,211,224
5310 DATA 17,0,10,27,122,179,32,251,5,16,234,62,159,211,224
5500 DATA 58,255,255,6,1,33,127,203,205,192,252,194,0,202
5510 DATA 1,0,112,33,0,1,205,210,252,194,0,202
5520 DATA 62,1,205,195,252,58,255,255,50,1,65,195,0,1
5900 DATA 205,108,252,48,251
5999 DATA 201,-1
6000 mr = 108*256
6010 READ m1: IF m1 = -1 GOTO 7000
6020 POKE mr,m1: mr = mr+1: GOTO 6010
7000 IF mr = 27891 GOTO 10000
7010 PRINT " DATA error (0-7000)": END
10000 REM error routine
10010 DATA 62,226,211,224,62,240,211,224
10020 DATA 17,0,200,27,122,179,32,251
10030 DATA 17,0,200,27,122,179,32,251,62,255,211,224
10040 DATA 62,32,17,0,3,33,0,8,205,38,253
10050 DATA 17,8,8,33,137,203,1,17,0,205,26,253
10060 DATA 17,132,8,33,154,203,1,24,0,205,26,253
10070 DATA 17,196,8,33,178,203,1,21,0,205,26,253
10080 DATA 17,228,8,33,199,203,1,26,0,205,26,253
10090 DATA 24,20,62,226,211,224,62,240,211,224
10100 DATA 17,0,100,27,122,179,32,251,62,255,211,224
10110 DATA 205,108,252,48,251
10120 DATA 254,49,32,3,195,48,252,254,50,32,220,195,231,252
10900 DATA 205,108,252,48,251
10999 DATA 201,-1
```

```

11000 er = 110*256
11010 READ m1: IF m1 = -1 GOTO 12000
11020 POKE er,m1: er = er+1: GOTO 11010
12000 IF er = 28294 GOTO 15000
12010 PRINT " DATA error (10000-12000)": END
15000 REM print strings
15010 ps = 111*256
15020 wd$ = " SmartBASIC V1.0 ": GOSUB 16500
15030 wd$ = "(tm) Coleco Industries, Inc.": GOSUB 16000
15040 wd$ = "(c) 1983 LMI": GOSUB 16000
15050 wd$ = "bootstrap routine (c) 1986": GOSUB 16000
15060 wd$ = "by DIGITAL EXPRESS, Inc.": GOSUB 16000
15070 wd$ = "one moment please...": GOSUB 16000
15080 wd$ = "BASICPGM"+CHR$(2)+CHR$(3): GOSUB 16000
15090 wd$ = " FILE LOAD ERROR ": GOSUB 16500
15100 wd$ = "Please select an option:": GOSUB 16000
15110 wd$ = "1.  reboot the system": GOSUB 16000
15120 wd$ = "2.  exit to word processor": GOSUB 16000
15900 IF ps = 28641 GOTO 20000
15910 PRINT " String error (15000-16000)": END
16000 w% = LEN(wd$): FOR x = 1 TO w%
16010 POKE ps,ASC(MID$(wd$,x,1))
16020 ps = ps+1: NEXT: RETURN
16500 w% = LEN(wd$): FOR x = 1 TO w%
16510 POKE ps,ASC(MID$(wd$,x,1))+128
16520 ps = ps+1: NEXT: RETURN
20000 HOME: PRINT: HTAB 2
20010 PRINT "Use ONLY on a BASIC backup!!": PRINT
20020 PRINT " Which drive?": PRINT
20030 PRINT " 1 = tape one": PRINT " 2 = disk one"
20040 PRINT: PRINT " any other key to abort ..."
20050 GET key$
20060 IF key$ = "1" THEN dv% = 8: GOTO 21000
20070 IF key$ = "2" THEN dv% = 4: GOTO 21000
20100 TEXT: PRINT " program terminated.": END
21000 POKE 27601,dv%
21010 HOME: PRINT " press [RETURN] to alter..."
21020 GET key$: IF key$ <> CHR$(13) GOTO 20100
21030 HOME: PRINT " one moment please..."
21040 CALL 27600: GOTO 20000

```

The  
Family  
Computer  
System



**TITLE (asmb#20):**  
**DATA/REM Extra Spaces Patch**

<u>Line#:</u>	<u>Label:</u>	<u>Decimal value:</u>	<u>Op Code:</u>	<u>Comment:</u>
1	SKIP	19,	INC DE	; advance one position
2	ENTRY	26,	LD A, (DE)	; get current ASCII byte
3	CHECK	183,	OR A	; check if end of program line
4		200,	RET Z	; RETURN to BASIC if end
5		254, 32,	CP \$20	; check if space
6		40, 248,	JR Z, \$FB	; if so, goto SKIP
7		183,	OR A	; restore flags
8	DONE	201	RET	; if not, RETURN to BASIC

**TITLE (asmb#21):**  
**New SmartBASIC V1.0 Bootstrap**  
 (COLD START)

<u>Line#:</u>	<u>Label:</u>	<u>Decimal value:</u>	<u>Op Code:</u>	<u>Comment:</u>
1	START	49, 128, 209,	LD SP, \$D100	; set stack
2		128,	LD A, B	; get current drive number
3		50, 255, 255	LD (\$FFFF), A	; store drive number

**TITLE (asmb#22):**  
**New SmartBASIC V1.0 Bootstrap**  
 (FIRST SOUND ROUTINE)

<u>Line#:</u>	<u>Label:</u>	<u>Decimal value:</u>	<u>Op Code:</u>	<u>Comment:</u>
1	SETUP	6, 20,	LD B, \$14	; load note/counter value
2	LOOP	62, 128,	LD A, \$80	; set-up first sound frequency
3		211, 224,	OUT (\$E0), A	; send byte to sound chip
4		120,	LD A, B	; get second sound frequency
5		211, 224,	OUT (\$E0), A	; send second sound byte
6		62, 146,	LD A, \$92	; get volume
7		211, 224,	OUT (\$E0), A	; send volume
8	DELAY	17, 0, 10,	LD DE, \$00A0	; set delay count
9		27,	DEC DE	; decrement counter
10		122,	LD A, D	; check decrement
11		179,	OR E	; for zero
12		32, 251,	JR NZ, \$FB	; if not, goto DELAY
13		5,	DEC B	; decrement note
14		16, 234,	DJNZ \$EA	; if note not zero, goto LOOP
15	DONE	62, 159,	LD A, \$9F	; get volume off value
16		211, 224,	OUT (\$E0), A	; send volume byte

**TITLE (asmb#23):**  
**New SmartBASIC V1.0 Bootstrap**  
 (INITIALIZE VIDEO CHIP)

<u>Line#:</u>	<u>Label:</u>	<u>Decimal value:</u>	<u>Op Code:</u>	<u>Comment:</u>
1	Vreg#0	1, 0, 0,	LD BC,\$0000	; set-up VRAM graphics mode
2		205, 32, 253,	CALL \$FD20	; set Vreg zero
3	Vreg#1	1, 224, 1,	LD BC,\$01E0	; set-up VRAM display mode
4		205, 32, 253,	CALL \$FD20	; set Vreg one
5	Vreg#3	33, 0, 0,	LD HL,\$0000	; set-up ASCII pointer
6		62, 3,	LD A,\$03	; continue set-up
7		205, 41, 253,	CALL \$FD29	; implement ASCII set-up
8	Vreg#2	33, 0, 8,	LD HL,\$0B00	; set-up screen start
9		62, 2,	LD A,\$02	; continue set-up
10		205, 41, 253,	CALL \$FD29	; set VRAM screen start
11	FONT#1	205, 56, 253,	CALL \$FD38	; load ASCII from ROM
12	FONT#2	17, 0, 4,	LD DE,\$0400	; set VRAM INVERSE start
13		33, 0, 0,	LD HL,\$0000	; load VRAM copy from addr
14		1, 128, 0,	LD BC,\$0080	; load # of fonts to copy
15		205, 23, 253,	CALL \$FD17	; set INVERSE fonts
16	Vreg#4	33, 0, 11,	LD HL,\$0B00	; load VRAM color start
17		62, 4,	LD A,\$04	; continue set-up
18		205, 41, 253,	CALL \$FD29	; set color table pointer
19	COLR#1	62, 25,	LD A,\$19	; get NORMAL color
20		17, 16, 0,	LD DE,\$0010	; load # (* 8) to set
21		33, 0, 11,	LD HL,\$0B00	; load VRAM color start
22		205, 38, 253,	CALL \$FD26	; set NORMAL color
23	COLR#2	62, 244,	LD A,\$F4	; get INVERSE color
24		17, 16, 0,	LD DE,\$0010	; load # (* 8) to set
25		33, 16, 11,	LD HL,\$0B10	; load VRAM color start
26		205, 38, 253,	CALL \$FD26	; set INVERSE color
27	Vreg#7	1, 6, 7,	LD BC,\$0706	; set-up background color
28		205, 32, 253,	CALL \$FD20	; set Vreg seven

**TITLE (asmb#24):**  
**New SmartBASIC V1.0 Bootstrap**  
 (PRINT STRINGS)

<u>Line#:</u>	<u>Label:</u>	<u>Decimal value:</u>	<u>Op Code:</u>	<u>Comment:</u>
1	STRING	17, lo, hi,	LD DE,\$nnnn	; load screen position
2		33, lo, hi,	LD HL,\$nnnn	; load RAM addr of string
3		1, lo, 0,	LD BC,\$nnnn	; load string length
4		205, 26, 253,	CALL \$FD1A	; transfer string to VRAM

**TITLE (asmb#25):**  
**New SmartBASIC V1.0 Bootstrap**  
 (GET SmartBASIC)

<u>Line#:</u>	<u>Label:</u>	<u>Decimal value:</u>	<u>Op Code:</u>	<u>Comment:</u>
1	OPEN	58, 255, 255,	LD A, \$FFFF	; get current drive value
2		6, 1,	LD B, \$01	; set file mode
3		33, 127, 203,	LD HL, \$CB7F.	; get RAM addr of filename
4		205, 192, 252,	CALL \$FCC0	; call EDS open file
5		194, 0, 202,	JP NZ, \$CA00	; if error, goto ERROR
6	READ	1, 0, 112,	LD BC, \$7000	; set file length
7		33, 0, 1,	LD HL, \$0100	; set RAM begin address
8		205, 210, 252,	CALL \$FCD2	; call EDS read file data
9		194, 0, 202,	JP NZ, \$CA00	; if error, goto ERROR
10	CLOSE	52, 1,	LD A, \$01	; set file mode
11		205, 192, 252,	CALL \$FCC3	; call EDS close file

**TITLE (asmb#26):**  
**New SmartBASIC V1.0 Bootstrap**  
 (CORRECT DEFAULT DRIVE)

<u>Line#:</u>	<u>Label:</u>	<u>Decimal value:</u>	<u>Op Code:</u>	<u>Comment:</u>
1	FIX	58, 255, 255,	LD A, \$FFFF	; get current drive value
2		58, 1, 65,	LD (\$4101), A	; load value into 16641
3	BEGIN	195, 0, 1,	JP \$0100	; start SmartBASIC

---

*Christmas 1986*  
 is almost here

. . .

This year, give  
 a present with a  
 future -- give  
**ADAM™ software & hardware!!!**

---

Asmb# 23:

This routine initializes the video chip. In standard TEXT mode (32 column), the first 5 video registers must be set with compatible values. Registers 0, 1, and 7 only require an, easy to calculate, value. For these we have employed the EOS routine at 64800 (\$FD20). It just transfers the value in the Z80 register C to the video register indicated by the value in register B.

The remaining video registers are easier to set with the EOS routine at 64809 (\$FD29). These registers are used to store pointers to the various VRAM tables. The pointers, however, are offset by individual factors. With this EOS routine, you simply store the true VRAM address in the HL register pair. Then set the accumulator to equal the video register to initialize and CALL 64809.

Asmb# 24:

We've employed the EOS routine at 64794 to print strings. There are alternatives, however, this **write table to VRAM** routine is the easiest to set up.

Asmb# 25:

This routine retrieves the BASICPGM from the medium. The program is opened, read, and closed. If an error is encountered at any point (the Z flag is NOT set), the program jumps to the error routine. This particular routine could be used to boot any machine code program.

SmartBASIC begins at address 256 (\$0100). The BASICPGM file is 28K in length. Thus, the value of register B in line 6 is 112 (28 \* 4).

Asmb# 26:

This routine sets the default drive for BASIC and starts the interpreter's processing loop. Address 16641 is used to store the default drive value. We simply transferred the bootstrap drive code to that address (in line 2).



## GETTING INTO CP/M 2.2

### BOOKS TO READ

As you get more involved with CP/M, you'll no doubt want to expand your learning experience beyond our articles and the software manual. This month we've listed five of our favorite CP/M guides.

#### **MASTERING CP/M**

by Alan R. Miller  
published by SYBEX  
(for intermediate to advanced CP/M programmers)

#### **Programming the Z80**

by Rodney Zaks  
published by SYBEX  
(detailed, easy to understand guide to Z80 programming)

#### **8080/Z80 ASSEMBLY LANGUAGE**

by Alan R. Miller  
published by Wiley  
(fast paced review of 8080, Z80 and CP/M programming)

#### **The CP/M Handbook**

by Rodney Zaks  
published SYBEX  
(detailed CP/M guide for beginners)

#### **User-Friendly Guide to CP/M 2.2**

by Tony Bove and Cheryl Rhodes  
published by Reston  
(a comprehensive guide to CP/M)

### THE BUILT-IN COMMANDS

(part 2)

The REN command is used to RENAME files. The sequence of the names is the opposite of SmartBASIC. The format is:

**REN A: NEWNAME=OLDNAME**

The drive suffix is optional. And, you should note that this command only changes the filename; it doesn't alter the file's contents.



---

---

**ADAM PRODUCT REVIEWS**

PRODUCT:	MageQuest
MANUFACTURER:	REEDY SOFTWARE
MEDIA TYPE:	DDP/disk
GRAPHICS/SOUND/DESIGN:	98
INSTRUCTIONS:	97
USEFULLNESS vs PRICE	99
RECOMMENDATION:	highly recommended
PRICE:	\$15.95/13.95
RATED BY:	staff

MageQuest is a graphic adventure written primarily by Brian Miguel and marketed by REEDY SOFTWARE. This excellent BASIC game package synergistically combines the intellectual challenge of a text adventure with the eye-to-hand coordination of an arcade game. The graphics are not extremely complex. However, the animation is very smooth and realistic, and the reaction time is very quick. The sound is also nicely done.

The game starts with your mage, the onscreen figure, in the lowest level of Enteon's dungeons. You move your mage from room to room searching for the ward, powerful stolen weapon, on each of the nine levels. When you find a ward, you are transported to the next level. Along the way, you'll acquire various spells and encounter evil beasts. Each beast has two versions, fast animation and slow animation. The spells are for use on the fast beasts. If a beast captures you, the game ends. Also, some rooms are locked. Here, you need to search for the key before you can enter.

Once you've mastered all nine levels, which requires considerable skill, you can continue the fun with REEDY's SOLO adventures, single level adventures. The medium includes three of these. REEDY SOFTWARE also offers continual support of the package via new SOLO adventures. These are purchased separately for use with MageQuest.

MageQuest is well worth the money. Even if you're not an avid game player, this impressive package will provide many, many hours of enjoyment.

# YES...ADAM

---

---

PRODUCT:	BACKUP+ 3.0
MANUFACTURER:	MMSG
MEDIA TYPE:	DDP/disk
GRAPHICS/SOUND/DESIGN:	96
INSTRUCTIONS:	97
USEFULLNESS vs PRICE	97
RECOMMENDATION:	highly recommended
PRICE:	\$34.00/33.00
RATED BY:	staff

BACKUP+ 3.0 is a full featured media copy utility. With the standard ADAM, it uses a 40K copy buffer. And, with the 64K expander, it employs a 102K copy buffer.

Some of the copy features include: copy individual blocks, copy range of blocks, copy files only, and selective file copy. It also includes INIT, format disk, catalog display or print, recover deleted files, and block status options.

MMSG also offers a lower priced version, BACKUP 3.0 (no plus), which has fewer features. MMSG's dedication to ADAM is expressed not only in their quality products; but, also in their continual updating service. We are so impressed by this package, that we now use it for our own (business scale) media duplication. BACKUP+ 3.0 is, indeed, a very useful piece of software.

PRODUCT:	SmartBASIC BONANZA
MANUFACTURER:	MARTIN CONSULTING
MEDIA TYPE:	DDP/disk
GRAPHICS/SOUND/DESIGN:	84
INSTRUCTIONS:	85
USEFULLNESS vs PRICE	82
RECOMMENDATION:	recommended
PRICE:	\$34.95
RATED BY:	staff

When this package was first introduced (over two years ago), it was rather impressive for ADAM. However, within the last year or so we ADAMites have enjoyed the benefits of the explosion of technical ADAM information. No longer is software for ADAM stigmatized by the euphemistic phrase good for ADAM. Unfortunately, the developers of this package haven't offered updates to reflect currently available technical knowledge.

The package consists of 18 BASIC programs. It includes a text adventure, a few graphics and sound programs, a couple of games, and a few utilities. Today, the value of this package as entertainment is greatly diminished. However, it still maintains a certain educational merit for novice programmers.

PRODUCT:	TriviaPac I
MANUFACTURER:	Mr. T SOFTWARE
MEDIA TYPE:	DDP/disk
GRAPHICS/SOUND/DESIGN:	98
INSTRUCTIONS:	93
USEFULLNESS vs PRICE	95
RECOMMENDATION:	highly recommended
PRICE:	\$22.00
RATED BY:	staff

TriviaPac I is a fun-to-play trivia challenge. One to four players are allowed. It uses color and sound very favorably. It controls input professionally and presents pleasing graphics screens.

The package contains 1200 questions (8 sets of 150 questions) in 6 categories. The categories are: MUSIC & STAGE, TV & SCREEN, TIMES & EVENTS, PEOPLE & PLACES, ART, LITERATURE, & SCIENCE, and SPORTS & GAMES.

When you answer a question correctly, ADAM marks that category. When all six categories are marked, you get to choose your WINNING CATEGORY. The first correct answer in a WINNING CATEGORY gets his/her name in the HALL OF FAME as the winner.

PRODUCT:	JEOPARDY
MANUFACTURER:	Coleco
MEDIA TYPE:	DDP/disk
GRAPHICS/SOUND/DESIGN:	99
INSTRUCTIONS:	90
USEFULLNESS vs PRICE	99
RECOMMENDATION:	highly recommended
PRICE:	PUBLIC DOMAIN
RATED BY:	staff

We ADAMites are very fortunate to have this package available at public domain prices. It is probably the BEST game that Coleco ever developed. We now carry it as a PD volume. The only flaw with the program is that the data pack version is very slow.

Since the only instructions are onscreen, we'll explain a few of the features. CNTL-V at almost any point will toggle the background music. STORE allows you to store a partially completed game. UNDO takes you back to the previous menu. WILDCARD toggles between the player graphics with the scores and the answer board.

One to three players are allowed. When asked if you are using a question pack, you can use a trick to quiz on the game designers. To do so, press CNTL + up arrow, CNTL + right arrow, CNTL + down arrow, and finally CNTL + left arrow.

If the winner's score is high enough, it goes into CHAMPION'S HALL OF FAME. And, he/she can have the ADAM printer print a winner's certificate (ADAM dollar). Returning champions can continue their score in new games.

To promote this outstanding public domain volume, we're offering a \$10.00 DEI purchase credit to the FIRST FIVE players (must be a N&B subscriber) who send in a winner's certificate verifying a score higher than Dr. Swift's 635600. His certificate is on the next page. We've even included two high scores on our JEOPARDY volumes that will make it easier for you -- you do NOT have to get the JEOPARDY disk or data pack from us to be a winner though.



## INTRODUCING . . .

The most comprehensive graphics design software ever written for ADAM™,

### ShowOFF I: The Graphics Design Package

The most amazing aspect of ShowOFF I is the power that you control. This is one piece of software that really taps into ADAM's potential.

It includes two primary programs, BlockPAINT 2.3 and SmartPAINT. BlockPAINT 2.3 is a specially enhanced version of the program we LISTed in our September issue. The newer version is fully compatible with files created by version 1.7. And, you can PRINT the low resolution graphics screens on the ADAM printer or your compatible dot matrix printer -- directly from within BlockPAINT.

The prize of this package is SmartPAINT, the high resolution graphics design package. You have 40 paint brushes and 14 colors to choose from. It will automatically draw circles, triangles, quadrangles, and lines for you. It even includes an automatic pen movement option.

SmartPAINT gives you two automatic border selections. You have access to a FAST full screen TEXT editor -- put text right next to your graphics. All 96 standard fonts are included, plus 22 special characters and symbols. You can even rotate any of the characters or paint brushes.

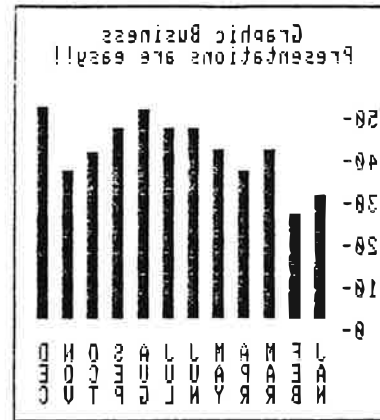
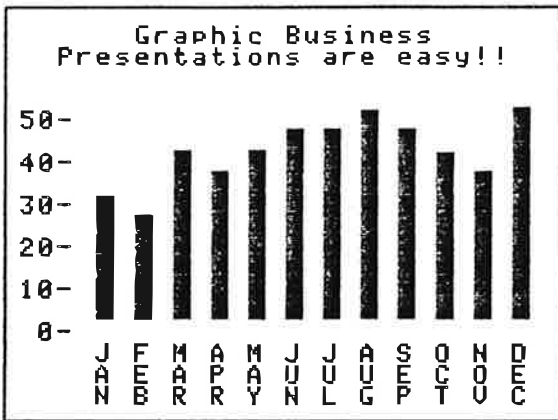
Several powerful color options are included. The monochromatic option instantly converts the entire drawing to your specified color. The inverse monochromatic option instantly changes all foreground colors to black and the background to the color of your choice. Another option will instantly change the background screen to your preference without changing any other colors. Still another option, allows you to instantly change all of one color to the new color of your choice.

Of course, both BlockPAINT and SmartPAINT include SAVE and LOAD picture options. And, both support any Epson FX compatible printer (the Panasonic KX series, the Seikosha SP series, the Star Micronics NX-10 and SG-10, etc). The printer can be connected to ADAM via either the Orphanware PIA2 or the Eve Electronics SP-1.

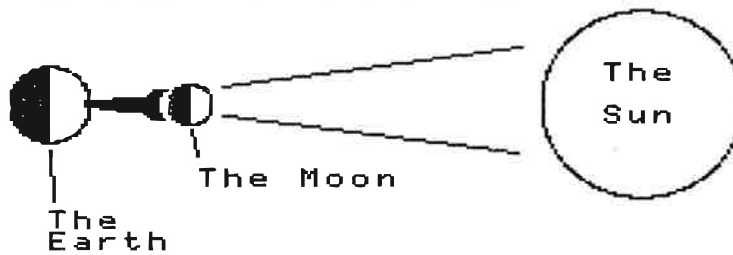
All the graphics in this issue were printed with SmartPAINT. With the extensive, easy to follow instruction manual you can begin creating impressive graphics in hardly no time at all!

There are even a few BONUSes. The package includes a patch to PaintMASTER (by STRATEGIC SOFTWARE) which allows you to convert its files to SmartPAINT format. You can even use your own graphics created independent of SmartPAINT. And, you can use SmartPAINT pictures in your own programs.

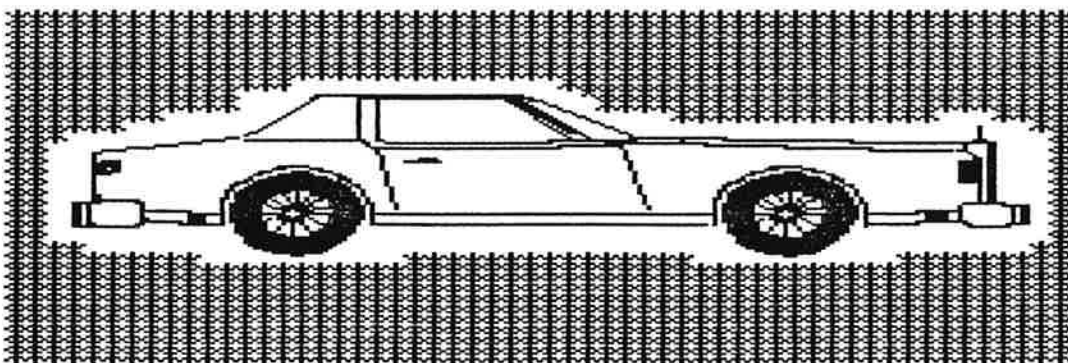
Don't miss out on this one. It's destined to be a bestseller. The standard non-subscriber price is only \$29.95. NIBBLES & BITS subscribers can get it (on DDP or disk) for only \$24.95. However, if you order it before 1/1/87, as a subscriber, you can get this powerful graphics design package (written specifically for ADAM) for ONLY \$19.95. This special offer will NEVER be repeated. Get ShowOFF I today, and SAVE . . .



During a SOLAR ECLIPSE, the moon's shadow is cast on a portion of the earth.



MODERN TRANSPORTATION



SHOWOFF I

The Graphics Design Package



**BULLETIN  
BOARD**

ADAM Software  
**REEDY SOFTWARE**  
10085 60th Street  
Alto, MI 49302

ADAM Software  
**Mr. T Software**  
7316 Northway  
Hanover Park, IL 60107

Looking for local ADAMites  
**Earl R. Kiesler**  
3310 Milford Road  
Indianapolis, IN 46236

ADAM Software  
**Murdock Games**  
2645 Wilbanks Circle  
Dallas, GA 30132

ADAM Software  
**ALPHA-ONE LTD**  
1671 East 16th Street, Suite 146  
Brooklyn, NY 11229  
(718) 336-7612

ADAM Hardware and Software  
**E & T SOFTWARE**  
P.O. Box 821242  
Dallas, TX 75382-1242  
(214) 340-8913

ADAM Hardware and Software  
**STEVE JACOBY**  
Box 11205  
Clearwater, FL 33516

PRINTING SERVICES - NEWSLETTERS, ETC.  
Ted Gocal, Gannon University  
121 West 7th Street  
Erie, PA 16541  
Compuserve ID# 75226,226

Media Backup Utilities  
**MMSG**  
P.O. Box 1112  
Broomfield, CO 80020-8112

ADAM Software  
**ADAMagic**  
1634 North Thompson Drive  
Bay Shore, NY 11706

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

Suffered LIGHTNING damage  
**ADAMLAND**  
795 Garfield  
Lander, WY 82520

120 Public Domain Disks  
**Barry Wilson**  
1566 Wood Lake Drive  
Chesterfield, MO 63017

CP/M Software for ADAM  
**ELLIAM ASSOCIATES**  
24000 Bessemer Street  
Woodland Hills, CA 91367

Advanced Utilities  
**Overpriced Software**  
P.O. Box 89  
Farmington, CT 06034

Programming Instruction  
**National A-Club**  
P.O. Box 15068  
Chevy Chase, MD 20815

ADAM Hardware and Software  
**THE ADAM DEPOT**  
419 Ridway Avenue  
Johnsonburg, PA 15845

**LOCAL ADAM  
USERS GROUPS**

**PRODUCT LIST**

**TEXAS**

ADAM Users of EL Paso  
Dick Lewin  
6308 Falling Star  
El Paso, TX 79912

Houston AUG  
Thomas Rutan  
1805 14th Avenue North  
Texas City, TX 77590

**VIRGINIA**

Central Virginia AUG  
Thomas J. Kelly  
Route 664, 3B  
Earlysville, VA 23508

Norfolk ADAM Group  
Gerald M. Steen  
1000 Rockbridge Avenue, #144  
Norfolk, VA 23508

Washington D.C. AUG  
Jim Tyson  
1811 St. Roman Drive  
Vienna, VA 22180

**WASHINGTON**

Puget Sound ADAM Network  
Valorie Zimmerman  
22607 SE 322nd Street  
Kent, WA 98042

Dave Sandahl  
USNH, Box 2844  
FPO Seattle, WA 98778

**WISCONSIN**

CCCAUG  
Denny Sekmer  
(414) 259-9954

**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 V1.0  
converts BASIC 1.0 programs to LOAD up to 12 times  
faster -- stays in RAM plus onscreen help

\$15.95 STANDARD PRICE  
\$11.95 SUBSCRIBER DISCOUNT PRICE

■■■ ShowOFF I  
powerful graphics design package -- includes a  
variety of print options

\$29.95 STANDARD PRICE  
\$24.95 SUBSCRIBER DISCOUNT PRICE

■■■ Intel-LOAD V2.0  
converts BASIC 2.0 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

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 warranted 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 discretion).

■■■■■■ The prices listed above are effective  
11/15/86 through 12/15/86

## PRODUCT ORDER FORM

Your name _____
Address _____
City _____ State _____
ZIP _____ Phone _____
ID Number _____

PRODUCT	QNTY	MEDIA	PRICE

Subtotal: \_\_\_\_\_  
 Shipping: \_\_\_\_\_ (inside contiguous USA: \$2.50; elsewhere: \$4.00)  
 Tax: \_\_\_\_\_ (WV residents only: 5%)  
 Other: \_\_\_\_\_  
 Other: \_\_\_\_\_ (subscription/renewal)  
 Total: \_\_\_\_\_

***- Thank you for your support!!! -***

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 25901

If you're a new subscriber, please answer these questions:

1. How long have you used an ADAM™ computer?
2. What topics would you like to see discussed in NIBBLES & BITS?
3. Would you like to contribute articles, reviews or programs?
4. Briefly, describe your system . . .

## SOFTWARE EXCHANGE

We now have 5 BASIC public domain volumes and 6 Coleco public domain volumes. Each BASIC volume (requires SmartBASIC) includes two instruction files which can be read or printed from SmartWriter. All BASIC programs are speed-RUN. Most of the BASIC programs (except the UTILities volume) are controlled from a RAMDISK written primarily in machine code.

To get a FREE copy of a specific BASIC volume: (1) contribute an original program, (2) send a signed statement that the program is not copyrighted, (3) send the program on DDP or disk, (4) request the specific volume that you want in return, and (5) include return postage and a mailer or \$2.50 for shipping. Below is the directory for the N&Bgames01 volume.

The BASIC volumes are N&Bgames01, N&Bgames02, N&Butil01, N&Bmath01, N&Bgraph01. The Coleco volumes are Jeopardy, Troll's Tale, SubRoc, Pinball Construction with Hardhat Mac, Pinball Games Set#01, and ADAMLink II. Each volume may be purchased on DDP or disk for ONLY \$5.95.

VOLUME TITLE: N&Bgames01		FREE BLOCKS: 28					
BOOT	:S 1	DIRECTORY	:S 1	HELLO	:A 1	m1.obj	:H 3
GoHACKER	:H 2	HackerDISK	:H 6	BASICP&M	:M 1	JackPot	:H 8
HangMan	:H 4	Cannons	:H 4	Missiles	:H 6	Robot	:H 3
RobotII	:H 3	RobotHunt	:H 4	BattleShip	:H 9	Surround	:H 3
BlackJack	:H 7	TicTacToe	:H 5	EvilOne	:H 5	OTquiz.BIG	:H 22
Baloo.BIG	:H 7	Cube.BIG	:H 13	READ-1.WPR	:H 6	READ-2.WPR	:H 7

## 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: \_\_\_\_\_

