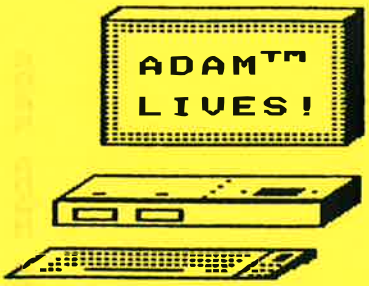
NIBBLES & BITS

the comprehensive monthly newsletter for ADAM users

P.O. Box 37
Oak Hill, WV 25901



AUGust 1987
vol: 2, nmb: 2
single issue: $3.50

This issue includes six SmartBASIC program LISTs and three disassembled Z80 routines.

DESIGNED and PRINTED with the amazing ADAM™ computer (using an Orphanware 64K expander, an Eve Electronics Centronics parallel interface, a Panasonic KX-P1080 dot matrix printer, ShowOFF I, and ShowOFF II).

NIBBLES & BITS

ADAM ADAM ADAM

ADAM ADAM ADAM

DIGITAL EXPRESS

PUBLIC NOTICE

"NIBBLES & BITS" is published monthly by DIGITAL EXPRESS. Individual issues may be purchased for the current month or for a back issue for $3.50; the *premier* issue was July, 1986. The standard subscription rate for one year (12 issues) is $22.00 in the USA, its possessions, and Canada; and, the annual rate is $30.00 in other foreign countries. The standard subscription rate for six months (6 issues) is $12.00 in the USA, its possessions, and Canada; and, the semi-annual rate is $16.00 in other foreign countries. All subscription issues are sent by U.S. mail, FIRST CLASS.

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 (next to last) 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 any issues, please renew early and let us know promptly of any address change. Please include your subscription ID number on the address change notification (you can get a free address change kit from your local US post office).

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

DIGITAL EXPRESS

CREATIVE COMPUTING

for the ADAM

EDITOR'S NOTE

When Coleco officially announced it was to cease production of the ADAM there was an unexpected reaction. The nearly half million ADAM owners didn't all discard their orphaned computers to the family's storage room for cobweb collection. Most of us were first time computer owners and saw this situation as an incentive to explore into and dabble with the high tech orphan. As a result, many programmers (BASIC, LOGO, CP/M and z80) were borne.

At first, third party software development was rather slow and some of the early programs were a little weak. As programmers have refined their skills and expanded their knowledge of the system, commercial software has steadily become more sophisticated. Have you noticed that much of the latest software is written totally in machine code?

One of the first steps in becoming a proficient programmer is to use modular programming, *ie*, acquiring a library of routines that can be used in different programs. In writing **SpritePOWER** and **Clipper**, I accumulated nearly 10K of z80 routines. It occurred to me along the way that many of these routines could be useful to other programmers. As space permits, we'll examine each of them in N&B; in the meantime, I've started grouping the routines and modifying ADAM's ROM based EOS into one sort of universal package. We'll call it GO-DOS, Graphics Oriented Disk Operating System. It supports the use of SmartKEY labels at the bottom of the screen, directory file folders, bit manipulation techniques, graphics windowing, song routines, melodies while files load, drawing graphics, and lots more.

The package is still far from finished; I'm adding new algorithms and refining older ones as we develop other commercial titles. My hope is that other z80 programmers will consider GO-DOS a time-saving convenience for creating their own commercial programs.

One of the first purchasers of **SpritePOWER** asked me what assembler we used to write the 36K program. In truth, I didn't use an assembler. I have yet to find one that contains all the features that I need. Instead, I wrote the entire file using BASIC. I developed the program one block at a time transferring the code to disk by dint of DATA and POKE statements. This method is a little crude, but it gives you programming control that you won't find with any assembler; and, you don't have to learn hex to program in this manner. As you delve deeper into z80, you might want to try this block by block programming technique.

Solomon Swift
EDITOR

N&B NEWS

□□□ Our eleventh commercial program, **Clipper**, is now completed. This machine code program introduces the exciting concept of "clip art" to ADAM. It allows the user to draw and edit the small hi-res pictures and it also permits you to "capture" graphics from SmartPAINT hi-res pictures. It shows you how to use the files in your own programs and, of course, the program features all the graphics enhancements of our second generation of DIGITAL EXPRESS software.

□□□ Our seventh collection of public domain SmartPAINT picture files is now complete. And, REEDY SOFTWARE has just released a PD package of SmartPAINT pictures entitled ART GALLERY 2. All the files in their volume are drawings of Smurf-like characters. You now have 110 public domain pictures to choose from. All these files can be used in your own programs and viewed, edited, and printed (with a compatible dot matrix) with our **ShowOFF I** graphics design package. Each volume includes a baker's dozen (13) pictures.

□□□ There are three distinct origens of SmartPAINT pictures: those drawn with SmartPAINT (from ShowOFF I), those drawn with other computers and converted to SmartPAINT format, and those that are converted from digitized photographs. The following subscribers (in alphabetical sequence) have made significant contributions to this rapidly growing collection of graphics pictures:

David Carmichael
Doug Glenn
Kevin Lindquist
Dave McIntosh
Alan Neeley
Bryan Payton
Jack Reedy
Lee Smith
Anthony Yulo

□□□ We have replaced our PD volume entitled "CP/M test/music" with "CP/M demo carts". This newer package has eleven programs including the graphic system tester, music samples, the Coleco demo cartridge, a Coleco copy program, and several others. This package does require the 64K expander. If you ordered the earlier version in the last few weeks we should have sent you this latter one instead. If you've purchased the earlier volume from us, we'll update it for you for a $2.00 processing fee plus shipping and handling (to waive the shipping expense, you may send a return mailer with sufficient postage or order the update with a product order).

ADAM NEWS

□□□ Several weeks ago Orphanware announced the completion of their 256K memory expander for ADAM. Now they've released a 512K memory card. With this kind of memory access ADAM now has the potential to perform all the sophisticated tasks that today's latest, high-priced machines can.

They have four memory cards for ADAM, the MX-64 ($42), the MX-128 ($115), the MX-256 ($165) and the MX-512 ($285). When ordering the three larger boards you'll need to return your Orphanware PIA2 parallel interface or order the Orphanware Addresser (it costs 17.50). And, the three larger boards also come with CP/M programs to allow for up to a 494K ramdisk, correctly init 320K disk drives, parallel print drivers, and other patches. Orphanware also offers a variety of trade-in options for price discounts. For a detailed catalog and shipping cost information you may send an SASE to:

Orphanware
P.O. Box 324
Canal Fulton, OH 44614

GJMG Enterprises (authors of QUICKOPY) are also working on a new operating system that will support these powerful memory expansions. And, some of our upcoming DIGITAL EXPRESS programs may support the chips, as well.

□□□ REEDY SOFTWARE has just released another superb game. **STAGE FRIGHT**, written by Mike McCauley, is an extensive text adventure in which you play the role of an actor (or actress) trapped in an abandoned theater. See our review in this issue.

□□□ WALTER'S SOFTWARE has recently released two new programs. **LIBRARIAN** is a utility for keeping track of all your other programs. It reads the filenames from disks or data packs and stores them in one large SmartFILER compatible file. With SmartWriter you can search by file or volume name. You can also add comments with room for over 700 files. This could be a tremendous time-saver when you have a lot of files. The price is only $19.95.

Their other release is entitled **DOUBLE DISK FORMATTER**. If you have two disk drives, the program will format a disk in each drive <u>at the same time</u>. The program includes other features and will also work with one disk drive. It's just $9.95 on disk and $10.95 on data pack.

□□□ D.L. DECKER ENTERPRISES has Coleco game controller extension cables in good supply. Each cable is 10 feet long and is straight (not coiled). The price is only $9.95 per cable, or $18.95 for a set of two cables.

□□□ H & R Corporation has refurbished Coleco disk drives for just $99.50 plus shipping. The price includes the drive, power supply, and operating instructions. It does NOT include the ADAMnet cable or the Disk Manager. Orphanware will make ADAMnet cables from 3 to 6 feet (you specify length) for just $2.50 plus shipping. And, we have added the Disk Manager to our public domain library. H&R guarantees the drives for 30 days. This could be an excellent opportunity to get your first drive or to add a second.

□□□ American Design Components has a full page ad on page 59 of the September issue of *Family Computing*. About half of the ad is devoted to ADAM products. The products include both software and hardware. We ordered one of the "like new" disk drives for $199.00. Indeed, it looks and operates just like a new one. The only thing missing is the original outer box. And, it arrived in less than a week!!

□□□ Terry Fowler of gHAAUG has Coleco data drives for $9.95 plus three dollars shipping. This is a better price than the large surplus electronics warehouses are currently offering.

□□□ Orphanware now repairs ALL components for the ADAM computer. The service center is at a different location than the main company. See this month's BULLETIN BOARD for the address.

□□□ Here are two local users groups that will now mail their informative groups' newsletter nationally for only five dollars.

Puget Sound ADAM Network
22607 SE 322
Kent, WA 98042

Omaha ADAM Users Club
809 West 33rd Avenue
Bellevue, NE 68005

□□□ **HIGHLIGHTS**, the bimonthly newsletter of the 1986 ADAM Users Group, is no longer FREE. As the membership has grown over the past couple of years, they've found it necessary to charge $15 for annual memberships. The price is very reasonable as each issue is packed with tips, news, and reviews for ADAM and the new Sega and Nintendo game systems.

□□□ E&T SOFTWARE will begin publication of their new ADAM newsletter, **ADAM ALIVE**, in the next few weeks. Their new periodical will cover many aspects of ADAM with a concentration on software usage. And, subscribers will get discounts on their extensive line of ADAM products. See this month's BULLETIN BOARD for their address.

EXPANDED MEMORY

by Patricia J. Herrington

The 64K card is a memory expander for ADAM which allows more RAM (Random Access Memory). It was originally produced by Coleco as an add-on. When installed, it effectively doubles useable memory. What does this mean for you? Well ... that depends on what you use your ADAM for. Here are some advantages:

(1) When you use SmartWriter, the 64K card increases your workspace, allowing you to write, STORE, and GET longer files.

(2) When downloading messages with your modem, you needn't close and open files as often. If you are using an information service such as CompuServe, this saves on those on-line charges.

(3) When using ADAMCalc, you can use the 64K as a print buffer, according to the instructions in your manual. This means you can print out one spreadsheet at the same time as you're working on another!

(4) Now that the public domain interpreter, SmartBASIC 2.0, is readily available with its special extended memory mode (EXTMEM), you can write longer BASIC programs. You have over 90K of workspace.

(5) In CP/M, the 64K card is accessible as another drive (drive M). Using CP/M is one of the single strongest reasons people have bought 64K cards. A portion of that extra memory is actually set aside to act as though it were a separate, physical drive. But since whatever you load from it is really already in memory, it's loaded in a flash. You don't have to wait for the tape to spin, and it's actually faster than a disk drive! Which brings us to ...

(6) If you don't have a disk drive (or even if you do), a 64K card gives you speed you can't possibly get from tape, whenever you can use it as a separate drive. Now you can use your 64K card from SmartBASIC 1.0 (the version that came with your ADAM) as an extra drive (d7). There are now two commercial packages available that allow this ... something that was unheard of just last year. Walter's Software produces RAMDISK and DIGITAL EPXRESS has just recently came out with TurboDISK.

(7) New applications continue to crop up. For instance, Backup+ 3.0 (a superlative copy utility by MMSG Software) uses the expanded memory as a copy buffer. That means it can load 102K at a time, which, in turn, means you don't have to keep swapping media.

OVER THE PHONE LINES

by David E. Carmichael

Hello Fellow ADAMites. What I will be trying to do here is to help bring ADAM users together through the use of COMPUTER TELECOMMUNICATIONS. You may ask what is COMPUTER TELECOMMUNICATIONS, right? Well it is the use of your computer to contact another computer user. Thanks to work a few years back there are a few standards set up that allow owners of different brands of computers to TALK to each other.

One of the standards that was devised is called: American Standard Code for Information Interchange, "ASCII" for short. This is a set of codes that consists of numbers ranging from 0 to 127. And, each number represents an alpha-numeric character, punctuation mark, or control code. With these codes your ADAM may now talk to other computers via a hardware device called a MODEM. More on how these devices work later.

I know that some of you that have read the above are saying to yourself that you already know this! But the best place to start learning is with the elementary information.

For you advanced users of telecommunications I would like to suggest that you give the national system known as AMERICAN PEOPLE LINK a try (1-800-524-0100)! I make this suggestion for a number of reasons. Two of these are this system's low on-line non-prime time rates and the way this systems bills. This is the only national system that, to my knowledge, you do not need to give out personal banking numbers to (such as charge card or checking account) in order to be able to use it. You have the option of cash pre-payment!

If you run a local BBS please send me your phone number along with a log-on ID that I could use; due to the high costs of long distance calls, an ID would save me time. Also let me know the times your system is opperational. I will then be able to try out your system and do a write up later.

The world of COMPUTER TELECOMMUNICATIONS is a little slow right now. This is, in part, due to the fact that users of these systems seem to have lost the art of speaking their minds on topics and just use these services as a way to get FREE software. So when you do LOG-ON to your local or national BBS system, PLEASE take the time to start a topic or to reply to a topic that is being discussed! These topics need not be just computer based. One of my local BBS's hotest topics is the MIDDLE EAST conflicts. You never know what will come up.

So, till next time HAPPY COMPUTING!

TIDBITS

§§§ It's usually a good idea to include an SASE when writing to a mail order company for information or their catalog. If you don't, many companies will not reply. An SASE is a Self - Addressed, Stamped Evelope (usually #10 size will suffice).

§§§ ADAM's Operating System was developed by InfoSoft.

§§§ Wrapping unused monitor or TV cable in loose loops and fastening with a rubber band can sometimes reduce screen ghosts.

§§§ If you have a disk drive, you should clean the drive head every other month. You can get a standard disk drive cleaning kit at any computer supply store.

§§§ If you've used different color ribbons (red, blue, green, etc.) for your dot matrix printer, you may have noticed that these don't seem to last as long as standard black ribbons. This is because most companies don't include a re-inking pad inside the case for colored ribbons. The ribbon will most likely expend the ink supply long before the nylon wears out.

§§§ One of the most common problems with disk drives is consistent "write" errors; the drive reads files, but it won't write anything to the disk. The problem is usually caused by a bad photo sensor.

The drive behaves as if you have a write - protect tab on the disk. The write enable photo sensor is just inside the front of the drive on the left side. Repairing drives should be done by qualified technicians; but, if you find yourself in a tight situation, you may be tempted to remove the cover. There are six screws on the bottom of the drive. After you remove them, you'll need to remove the four screws that hold the top in place. Now you can see the write enable photo sensor. It has a green and a black wire coming from it. Try holding a flashlight to the sensor and writing to a disk. Sometimes this will provide temporary help. If you're skilled at electronics repair, you might also try inserting a micro - toggle switch on the black wire. This information is provided for those who are competent in electronics repair. Always turn the power off BEFORE performing electrical repairs. NEVER turn the power on while electrical components are exposed.

WISH LIST

Last month we asked readers to submit their *wish lists* of topics to be covered in upcoming issues of N&B. Some of the following items will be covered in this and/or future issues.

□□□ HOW TO CREATE INVERSE FONTS FOR ADAM'S 40 COLUMN TEXT MODE. See this month's HACKER'S DELIGHT department.

□□□ SHOW A PROGRAM OR PATCH THAT ALLOWS BASIC INPUT FROM THE KEYPAD AND THE KEYBOARD. See this month's HACKER'S DELIGHT department.

□□□ DISCUSS HOW ADAM's VARIOUS TONAL VALUES CORRESPOND TO ACTUAL MUSICAL NOTES. This will be explained in an upcoming issue.

□□□ EXPLAIN HOW TO PLAY MUSICAL TUNES WHILE FILES LOAD INTO MEMORY. This will be revealed in an upcoming issue.

□□□ SHOW HOW TO FIX THE SmartFILER EXTRA LINE FEED BUG IN PRINTING. SHOW HOW TO CHANGE THE SCREEN COLOR OF SmartFILER. We'll LIST a program that performs both of these functions next month.

□□□ INCLUDE MORE ARTICLES ON CP/M AND ADAMCalc. From time to time we have columns on CP/M; we may begin including these as regulars. We encourage our readers to submit articles. If you have information regarding a particular piece of software, please pass it along. We always give full credit to the author.

□□□ EXPLAIN HOW TO USE ADAM'S MULTI-COLOR MODE. We had planned to start a short series of articles on this topic this month. Due to the research involved, it will be another month or two before we publish them.

□□□ SHOW AN ML ROUTINE WHICH PARALLELS BASIC'S SCRN FUNCTION THAT WILL WORK WITH HI-RES GRAPHICS. We'll cover this one in an upcoming issue.

□□□ WRITE A PROGRAM THAT WILL CONVERT A BASIC PROGRAM INTO A MACHINE LANGUAGE PROGRAM. These type of compilers are extremely complex and require many, many hours of testing. It is unlikely that we'll include such a program in the newsletter. But, with the rapid rate at which third party software is now being released, it is possible that one could be written in the next year or two.

□□□ WRITE A GOOD PROGRAM LINE RENUMBERING UTILITY. We'll reveal a z80 routine that accomplishes this task in an upcoming issue.

BIT BY BIT

ASCII CODES

(part 2)

Last month we discussed that every operation ADAM performs is based on the simple concepts of "yes" and "no" (or "true" and "false"). We correlate these two possibliities as a logical one and a logical zero. Considering each value as a BIT, the conditions are grouped into a quantity referred to as a BYTE or WORD.

Using the binary system of math, each byte can represent a value ranging from 0 thru 255, inclusive. The bytes are further grouped into individual codes. One of these is the American Standard Code for Information Interchange or ASCII (pronounced "as-key"). With each byte value representing a different letter of the alphabet, number digit, special symbol, or control function the code is employed universally on personal computers. (Some large, older computers use a different system called EBCDIC).

The best aspect of ASCII is its standardization. This allows ASCII files to easily be transferred from one brand of computer to another via modem and standardized disk operating systems.

To store the letters of the name ADAM, the computer simply takes the input (from keyboard, disk, etc.) and converts it to ASCII binary code. If you PEEKed those memory addresses from BASIC you would see the decimal equivalent of the code. The decimal and binary ASCII values for "ADAM" are:

A = 65 = 01000001
D = 68 = 01000100
A = 65 = 01000001
M = 77 = 01001101

BYTE-SIZED BASIC

POKES TO PLAY WITH

(part 14)

The Vertical HPLOT limit:

Have you ever noticed that SmartBASIC won't allow you to HPLOT at the vertical coordinate of 159 in the HGR mode. This is another of the interpreter's minor bugs. To correct it, just:

```
POKE 25940, 160
```

Correcting HPLOT for 40 column TEXT:

In all the excitement of Ben Hinkle's introduction of a 40 column TEXT option for SmartBASIC, we overlooked that the HTAB command is setup for 32 columns. Here's how to adjust it for 40 columns.

after executing your 40 column text command,

```
POKE 26198, 39
```

after returning to 32 columns,

```
POKE 26198, 31
```

BASIC 2.0 FEATURES

With SmartBASIC 1.0 the random access file commands don't work correctly. Because it would require major rewriting of the interpreter, we will probably not fix this SB 1.0 bug. On the other hand, BASIC 2.0 offers significant improvements in this area.

All the file access commands work as they should. However, there is a limitation with WRITEing to and READing from random access files. The length of the file can NOT be greater than 32767 bytes (32K). If you try to exceed this limit, you'll get a "RANGE ERROR".

For some unknown reason the entire procedure of random access is interwoven with the BASIC INTEGER function. Since the largest integer the interpreter can handle is 32767, the file size is likewise limited. This is still far better than the SB 1.0 implementation of the file access feature.

Thanks to Ed Jenkins of E & T SOFTWARE for pointing out this minor bug.

HI-RES SHAPES

As we discussed last month, the shape table for hi-res shapes are stored in a machine code notation format (not in bit - image format, as sprite and font shape tables are stored). Each vector notation represents two or three movements with and/or without HPLOTting. It is MUCH easier to design shapes using a program designed specifically for the task than it is to do so mentally. "SHAPEMAKER", carried in our PD library, which was written by Guy Cousineau contains an example of such a program.

Plotting the vectors in a hi-res shape table is analogous to drawing with an imaginary pen. This pen may be put down and moved to draw and it may be lifted and moved without drawing. There are four "pen up" (lifted) movements and four "pen down" (draw) movements. This means that there are eight possible vectors or three BIT combinations for each type of movement ($2^3 = 8$). These are as follows:

LIFT/UP = 000
LIFT/RIGHT = 001
LIFT/DOWN = 010
LIFT/LEFT = 011
DRAW/UP = 100
DRAW/RIGHT = 101
DRAW/DOWN = 110
DRAW/LEFT = 111

A vectored shape table byte consists of three sections: the first three bits (0, 1, and 2), the second three bits (3, 4, and 5), and the last two bits (6 and 7). Either LIFT or DRAW vectors may be used in the first two of these sections. ONLY LIFT vectors can be used in the upper (two bit) section. This is due simply to the fact that the four bit combinations which represent a LIFT movement can each be stored as a two bit value (00, 01, 10, 11). Notice that the third bit is SET (a logical one) for the four DRAW vectors.

Let's examine the decimal equivalents of the possible binary vector values for each section of a byte.

Without plotting (LIFTED):

MOVE	sct 1	sct 2	sct 3
up	0	0	0
right	1	8	64
down	2	16	128
left	3	24	192

With plotting (DRAW):

MOVE	sec 1	sct 2
up	4	32
right	5	40
down	6	48
left	7	56

One of the most important considerations in using hi-res shape tables is that points are PLOTTED FIRST and then the pen is moved. If you don't keep this in mind, your shapes will be distorted.

Now, suppose you want to make three vectored moves: (1) plot and then move right, (2) plot and then move up, and (3) move left without plotting. Let's construct the notation byte. Just get the values from the two tables above.

The value for the first section (first move) is "5". The value for the second section (second move) is "32". Since the third move is LIFTed, we'll use the third section of the byte. The value for it is "192". Now, we just add the three values: 5 + 32 + 192 = 229. It's that simple.

Next month we'll go into more detail on constructing an entire shape table and discuss how to use BASIC's shape commands.

BASIC CONVERSIONS

This is the first in a new series of articles designed to help you convert BASIC programs between SmartBASIC V1.0, SmartBASIC V2.0, and AppleSoft BASIC. Many programs can be typed on the keyboard or downloaded from a BBS data library that will RUN perfectly well in any of these versions of BASIC. But, the vast majority will have to be modified to some extent. These alterations consist primarily of OS and interpreter PEEKs, POKEs, and CALLs. Solely to frustrate those interested in such conversions, some progrmmers even use CALLs vice BASIC commands.

You should keep in mind that the Apple and ADAM use different CPU's or computer brains. This means that machine language routines, such as music algorithms, will have to be completely re-written.

TWO versions of SmartBASIC 2.0:

Also, SmartBASIC 2.0 comes with two versions, *ie*, STDMEM and EXTMEM. When you first boot SB 2.0, you are in STandarD MEMory. If you have the 64K card, you can use the EXTMEM command to allow for a 90K workspace. Among other minor changes the EXTended MEMory interpreter, completely revises the lower 16K of RAM (Random Access Memory). You can go back to STandarD MEMory with the STDMEM command. Before using either command, you must have the SB 2.0 medium in the current drive.

You can determine if SB 2.0 detected the 64K card by PEEKing address 16788. If the value is zero, you can NOT use the STDMEM and EXTMEM commands. If the value is 255, you CAN use the two commands. Address 16789 is also used as a flag byte. This one determines which version is currently in RAM. If the value at the address is zero, you are in STDMEM. If the value is 255, you are in EXTMEM. Also, an easy way to determine if you're using SB 1.0 or SB 2.0 is to PEEK (259). If the value is 210, you are using SB 2.0. If the value is 195, you are using SB 1.0.

CURRENT DRIVE:

With SB 1.0, you can determine the current drive with PEEK (16821). With SB 2.0, PEEK (16781) reveals the current drive. Here are the drive values:

DRIVE	SB 1.0	SB 2.0
disk 1	4	4
disk 2	5	5
tape 1	8	8
tape 2	24	24
ramdisk	26	205

TEXT MARGINS:

MARGIN	ASoft	SB 1.0	SB 2.0
width	33	17198	17322
bottom	35	17199	17323
top	34	17201	17325
left	32	17202	17326

The Apple II series computers only come with one character width option for TEXT mode, *ie*, 40 columns. With SmartBASIC, you have two options for TEXT mode, *ie*, the standard 32 columns and 40 columns. There are several patches available for SmartBASIC that will let you use ADAM's 40 column TEXT mode (see our patch at the bottom of page 18 of the 12/86 issue). You will most likely want to use the 40 column mode for RUNning AppleSoft programs on ADAM.

The values of the margin addresses are not all absolute values on ADAM as they are on the Apple. The left margin is absolute; the default (start up) left margin is "1". The top margin is one less than the actual value; the default top margin is zero. The bottom margin value is the number of lines down from the top; the default bottom margin is "23". The line width is the number of rows over from the left margin; the default line width is "30". In 40 column TEXT mode, the default line width is "39". SmartBASIC performs calculations based on these values. Thus, you should not POKE in values that are too low or too high; the interpreter could crash. For example, don't set the top margin value higher than the bottom margin value.

THE HOME MARGINS:

With SmartBASIC you can also change the margins for the HOME command. The TEXT command will reset these four address values. This permits you to easily create a windowing effect by temporily changing the HOME clear values.

MARGIN	SB 1.0	SB 2.0
lines	16993	16957
right	16994	16958
top	16995	16959
left	16996	16960

THE SPEED VALUE:

All three BASIC's use the same values for the SPEED of screen printing, *ie*, 0 (slowest) thru 255 (fastest). The only difference is the location at which the value is stored. You can PEEK the address to determine the current SPEED value, or you can POKE in a new value to change the current SPEED. The default value is 255.

Asoft	SB 1.0	SB 2.0
241	16129	1628

HACKER'S DELIGHT

DATA PROCESSING

Over the previous eight months, we laid a fairly solid foundation for using the z80 data transfer op codes. With BASIC's PEEK and POKE commands, it was easy to demonstrate those functions with corresponding BASIC routines. The z80's data processing op codes, however, are more difficult to simulate with BASIC because most of them involve the manipulation of individual bits.

There are four general categories of this class of op code. ARITHMETIC OPERATIONS involve adding or subtracting byte values. BIT MANIPULATIONS change or test the logical status bits. LOGICAL OPERATIONS involve conjunctive and disjunctive bit changes. BIT SHIFT OPERATIONS shift the bits of a byte to the left or right with an option to include the Carry Flag.

This month we'll take a look at the CPL (ComPLiment) instruction. It is considered an arithmetic operation. Complimenting is simply reversing the logical status of a bit. If it was SET (logical one), it is changed to RESET (logical zero); and, vice versa. This instruction compliments the value stored in the accumulator (register A).

Suppose the value of a byte is 192; the compliment is 63. If the value of a byte is 15, the compliment is 240. Ergo, from the decimal perspective a CPL instruction subtracts the previous accumulator value from 255 (the result supplants the original accumulator value). For better understanding, consider these two examples.

```
11000000 = 192
00111111 =  63

00001111 =  15
11110000 = 240
```

There are three typical uses of CPL: advanced arithmetic, inverting bit image graphics, and changing the value of a status byte. Our True Inverse Fonts routine this month illustrates the use with bit image graphics. Let's take a look at the status byte concept.

Suppose you are writing a graphics design program and the user has the option of lifting the pen (stylus) or of putting it down to draw. Your pen movement routine will need to be able to determine the pen status (up or down). You could set aside a certain byte that determines this. For example, when the pen is down the status byte value would be "255"; when it is up the byte value would be "0". You could toggle the status (based on another user input) with the CPL op code. This sort of toggle status is rather common.

The CPL instuction does NOT require an operand (the accumulator is implied). The decimal value for CPL is "47". The hex value is "2F".

TRUE INVERSE FONTS

One of the minor shortcomings with ADAM's 40 column mode is that you can not distinguish INVERSE fonts from NORMAL fonts. This is due to the fact that the second set of fonts is not a true inverse bit image. The 32 column mode uses the same characters for INVERSE and just changes the color values. In 40 column mode, you can only use one SET and one UNSET color.

The program LISTed at the top of page 12 corrects this problem. After RUNning it, you can CALL 27600 to invert the bit image of the second group of font shapes. The machine code specifics of the routine are explained in ASMB # 53 (at the bottom of page 12).

This algorithm consists of three basic components: read current bit image values from VRAM, invert them, and then write the new values back to VRAM. The CPL op code is employed for the bit reversals. We have discussed the VRAM read and write table routines in previous issues. Addresses 27612 thru 27626 perform the bit image reversal. HL is setup with the starting RAM address and DE is setup with the byte count. The byte pointed to by HL is LoaDed into the accumulator, complimented, and then put back at the same address.

SHOW DELETED FILES

When the Operating System DELETEs a file, it only changes the attribute byte for the filename in the directory. If you store a file afterwards which is of equal or smaller size, the OS then replaces the DELETEd file. The program LISTed in the middle of page 12 patches the BASIC CATALOG command to allow you to see these "ghost" files along with the others. There is one minor difference.

To indicate that a file is LOCKed, CATALOG prints an asterisk to the left of the file size. Our patch uses an INVERSE letter "D" in the same position to indicate a DELETEd file. With our STATUS patch (LISTed on page 14 of the 11/86 issue) you can even recover these DELETEd files.

PUFF

Pages 16 thru 23 of this issue LIST an action - packed, arcade - style game entitled PUFF. In addition to being a lot of fun to play, it reveals NUMEROUS programming tricks. The game uses sprites for quick animation and includes some appealing sound effects ranging from a crash sound to a "CHARGE" melody.

It includes on - line instructions, so we won't go into a lot of detail on the program this month. By the way, this DIGITAL EXPRESS public domain contribution is included in our commercial package SpritePOWER.

Most of the input is through the game controllers. Line numbers 20000 thru 20710 include the instructions. To save some time on your first draft of typing, you might want to omit these lines. If you do, just change line number 20000 to: 20000 RETURN.

MoreKEYS

Over the past few months several readers have asked that we include a BASIC patch that will permit input from either the keyboard or the game controller keypad. MoreKEYS is our solution. The BASIC program that creates the z80 patch is LISTed on page 13.

This program could be particularly useful in typing programs with a lot of numerical DATA values. You can still use the keyboard as before and you can still use the PDL function in your programs. In addition to the standard number keys, we've included eight other features, these are:

```
joystick up      = up arrow
joystick right   = right arrow
joystick down    = down arrow
joystick left    = left arrow
right trigger    = <return>
left trigger     = <HOME>
asterisk (*)     = period (.)
sharp (#)        = comma (,)
```

Two z80 routines are needed. One reads the input from keyboard or game controller. The other debounces the game controller after input so that you don't get a long string of values for each input. ASMB # 54 (below) and ASMB # 55 (page 14) detail the routines. We will elaborate on using the EOS GC input function next month. There is one minor limitation with this patch; it stops the cursor from blinking.

CONTINUING ezFILER

The two programs on page 15 create the bit image graphics file used by the ezFILER bootstrap routine. We'll LIST the bootstrap and go into the technical aspects of how the ezFILER program works next month.

The "BootPic File Maker" program allocates directory space for a 5K file which will be named "BootPic". When completed, you'll see this picture upon pulling the reset switch; it will stay there while SmartBASIC loads into RAM. This feature is generally considered more appealling than just having the standard blank screen while BASIC loads. In fact, this trick can also be used for any medium that will contain the SmartBASIC program.

You MUST INIT the medium that you want use FIRST (use the default 1K directory size). Then, RUN this program. Press <return> to create the file in the directory. You can press any other key to abort.

The program entitled "BootPic Writer" writes the bit image data to the medium in the space set aside by the previous program. Here's how to use it.

FIRST, use the Pix.MGR program (LISTed on page 16 of the 2/87 issue and also included in our **ShowOFF I** package) to store an HGR screen. Use the "HGR file option" (don't store it in SmartPAINT format). After it's stored, enter LOMEM: 40000 in immediate mode. Then BLOAD the file. Now, RUN this program (BootPic Writer). Insert the medium that contains "BootPic" (created above) and press <return> to write the DATA to the file.

Both these programs are set up for use with the first disk drive. You can change the second data element (from a "4") in line # 110 in "File Maker" and in line # 100 in "Writer" to accomodate your system.

TITLE (asmb#54):

MoreKEYS

(debounce game controller subroutine)

```
addr:  Label:   Value(s):          Op Code:          Comment:

58521  DEBNCE   245                PUSH AF           ;store accumulator value
58522  check    221, 33,138, 65     LD   IX, 16778    ;set index pointer
58526             62,  2,          LD   A,  2        ;prep for controller #1
58528           205, 62,253        CALL 64830        ;read game controller
58531  keypad    58,146, 65        LD   A,  (16786)  ;get keypad value
58534           254, 15,           CP   15           ;check for no keypress
58536            32,240            JR   NZ, -16      ;if keypress, then "check"
58538  rgttrg    58,145, 65        LD   A,  (16785)  ;get right trigger value
58541           254,  0            CP   0            ;check for release
58543            32,247            JR   NZ, -9       ;if NOT release, "check"
58545  lfttrg    58,144, 65        LD   A,  (16784)  ;get left trigger value
58548           254,  0            CP   0            ;check for release
58550            32,247            JR   NZ, -9       ;if NOT release, "check"
58552  joystk    58,143, 65        LD   A,  (16783)  ;get joystick value
58555           254,  0            CP   0            ;check for center
58557            32,247            JR   NZ, -9       ;if NOT center, then check
58559  done      241               POP  AF           ;get incoming accumulator
58560            201               RET               ;exit subroutine
```

TITLE (asmb#55):

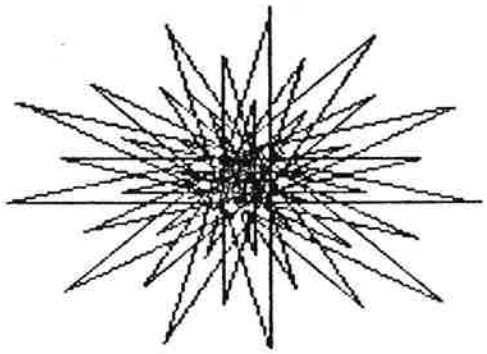
MoreKEYS

(read keyboard or game controller)

```
addr:  Label:  Value(s):          Op Code:          Comment:

58561  RESET   175                XOR  A            ;reset accumulator
58562           50,117,253         LD   (64885), A  ;reset input byte
58565  INLOOP  205,168,252         CALL 64680       ;start keyboard read
58568           58,117,253         LD   A, (64885)  ;get input byte
58571          254,  0             CP   0           ;check for NO keypress
58573           40,  7             JR   Z, 7        ;if NONE, then "READGC"
58575  done    245                PUSH AF          ;store accumulator
58576          175                 XOR  A           ;reset accumulator
58577           50,117,253         LD   (64885), A  ;reset input byte
58580          241                 POP  AF          ;get previous accumulator
58581          201                 RET              ;exit patch
58582  READGC  229                PUSH HL          ;store HL pair
58583          213                 PUSH DE          ;store DE pair
58584          197                 PUSH BC          ;store BC pair
58585          221,229             PUSH IX          ;store IX pair
58587          221, 33,128,65      LD   IX, 16778   ;set index pointer
58591           62,  2             LD   A,  2       ;prep for controller #1
58593          205, 62,253         CALL 64830       ;read game controller
58596  keypad   58,146, 65         LD   A, (16786)  ;get keypad value
58599          254, 15             CP   15          ;check for no keypress
58601           40, 24             JR   Z, 40       ;if no keypress, "rgttrg"
58603          246, 48             OR   48          ;convert to ASCII
58605          254, 58             CP   58          ;check if "*" pressed
58607           32,  2             JR   NZ, 2       ;if not, skip two bytes
58609           62, 46             LD   A,  46      ;if so, change to "."
58611          254, 59             CP   59          ;check if "#" pressed
58613           32,  2             JR   NZ, 2       ;if not, skip two bytes
58615           62, 44             LD   A,  44      ;if so, change to ","
58617  GCdone  205,153,228         CALL 58521       ;debounce game controller
58620          221,225             POP  IX          ;retrieve IX pair
58622          193                 POP  BC          ;retrieve BC pair
58623          209                 POP  DE          ;retrieve DE pair
58624          225                 POP  HL          ;retrieve HL pair
58625           24,204             JR   -52         ;jump to "done"
58627  rgttrg   58,145, 65         LD   A, (16785)  ;get right trigger value
58630          254, 64             CP   64          ;check if pressed
58632           32,  4             JR   NZ, 4       ;if NOT, then "lfttrg"
58634           62, 13             LD   A,  13      ;if so, change to CR
58636           24,235             JR   -21         ;jump to GCdone
58638  lfttrg   58,144, 65         LD   A, (16784)  ;get left trigger value
58641          254, 64             CP   64          ;check if pressed
58643           32,  4             JR   NZ, 4       ;if NOT, then "joystk"
58645           62, 128            LD   A, 128      ;if so, change to <HOME>
58647           24,243             JR   -13         ;jump to GCdone
58649  joystk   58,143, 65         LD   A, (16783)  ;get joystick value
58652          254,  0             CP   0           ;check for center
58654           40, 26             JR   Z, 26       ;if center, then "REPEAT"
58656          254,  1             CP   1           ;check for up
58658           32,  2             JR   NZ, 2       ;if NOT, skip two bytes
58660           62,160             LD   A, 160      ;change to <up arrow>
58662          254,  2             CP   2           ;check for right
58664           32,  2             JR   NZ, 2       ;if NOT, skip two bytes
58666           62,161             LD   A, 161      ;change to <right arrow>
58668          254,  4             CP   4           ;check for down
58670           32,  2             JR   NZ, 2       ;if NOT, skip two bytes
58672           62,162             LD   A, 162      ;change to <down arrow>
58674          254,  8             CP   8           ;check for left arrow
58676           32,  2             JR   NZ, 2       ;if NOT, skip two bytes
58678           62,163             LD   A,  163     ;change to <left arrow>
58680           24,221             JR   -35         ;jump to GCdone
58682  REPEAT  221,225             POP  IX          ;retrieve IX pair
58684          193                 POP  BC          ;retrieve BC pair
58685          209                 POP  DE          ;retrieve DE pair
58686          225                 POP  HL          ;retrieve HL pair
58687          195,197,228         JP   58565       ;jump to INLOOP
```

```
 10 REM BootPic File Maker
100 LOMEM :28000
110 DATA 62,4,33,0,108,17,0,16,1,0,0,205,201,252,50,0,0,201
120 FOR x = 1 TO 18: READ mc: POKE x, mc: NEXT
130 na$ = "BootPic"+CHR$(2)+CHR$(3)
140 FOR x = 1 TO LEN(na$): POKE x+27647, ASC(MID$(na$, x, 1)): NEXT
200 TEXT: PRINT " This program allocates space"
210 PRINT " in the directory for a 5K file";
220 PRINT " which will contain the bit"
230 PRINT " image data for a hi-res"
240 PRINT " picture.": PRINT: PRINT: PRINT
250 PRINT " <return> = create file"
300 GET go$
310 IF go$ <> CHR$(13) THEN  END
320 CALL 1: PRINT PEEK(0)
```



```
  5 REM BootPic Writer
 10 REM This program will write an HGR picture to disk or datapack.
 20 REM It is very useful for making a file to be accessed by
 30 REM a bootstrap routine.
 90 LOMEM :40000: POKE 16149, 255: POKE 16150, 255
100 DATA 62,4,1,0,0,17,2,0,33,0,136,205,246,252,50,0,0,201
110 FOR x = 1 TO 18: READ mc: POKE x, mc: NEXT
120 TEXT: PRINT " Do NOT use this program unless";
130 PRINT " you've read the instructions"
140 PRINT " in the Ø8/87 issue of N&B!!!": PRINT: PRINT
150 PRINT " It can DESTROY a medium!!!": PRINT: PRINT: PRINT
200 PRINT " <return> = write disk"
210 GET go$: IF go$ <> CHR$(13) THEN  LIST: END
300 POKE 7, 2: POKE 11, 136: CALL 1: PRINT PEEK(0)
310 POKE 7, 3: POKE 11, 140: CALL 1: PRINT PEEK(0)
320 POKE 7, 4: POKE 11, 144: CALL 1: PRINT PEEK(0)
330 POKE 7, 5: POKE 11, 148: CALL 1: PRINT PEEK(0)
340 POKE 7, 6: POKE 11, 152: CALL 1: PRINT PEEK(0)
350 PRINT " picture on disk ...": END
```

```
 10 REM PUFF
 20 REM a public domain contribution
 30 REM by DIGITAL EXPRESS
 40 REM AUGust 1987
 50 REM game concept derived from FLY AWAY
 60 REM by Daryl L. Scott (c) 1983 by Microsparc, Inc.
 90 TEXT: HTAB 5: PRINT "one moment please ..."
100 LOMEM :28000: POKE 16149, 255: POKE 16150, 255: POKE 16134, 255
102 POKE 16953, 0: DIM nt(25), du(25), nn(12), dd(12), wn(22), wd(22)
104 POKE 18728, 121: POKE 18729, 0: POKE 18730, 0: POKE 25940, 160
106 lv = 1: rd = 0: eg(1) = 100: eg(2) = 100: sc(1) = 0: sc(2) = 0
110 REM initialize sprites routine
111 DATA 62,200,17,128,0,33,0,31,205,38,253
112 DATA 1,62,5,205,32,253,1,7,6,205,32,253
113 DATA 33,0,216,17,0,56,1,0,4,205,26,253
114 DATA 1,194,1,205,32,253,201
115 FOR x = 27600 TO 27641: READ mc: POKE x, mc: NEXT
120 REM move sprites routine
121 DATA 33,000,000,17,0,0
122 DATA 34,0,212,237,83,2,212,123,61,135,135,50,2,212
123 DATA 79,6,0,33,0,31,9,93,84,33,0,212,1,4,0,205,26,253,201
124 FOR x = 27642 TO 27680: READ mc: POKE x, mc: NEXT
130 REM note to sound chip routine
131 DATA 62,0,211,224,201
132 FOR x = 27681 TO 27685: READ mc: POKE x, mc: NEXT
140 REM notes off slowly routine
141 DATA 6,16,17,0,7,27,122,179,32,251,62,160,144,211,224
142 DATA 62,192,144,211,224,62,224,144,211,224,16,231,201
143 FOR x = 27686 TO 27713: READ mc: POKE x, mc: NEXT
150 REM attract song data
151 DATA 11,2,13,1,11,2,15,1,13,2,15,1,13,2,15,1,11,2,12,2,11,2,13,2
152 DATA 16,4,13,2,15,2,16,2,18,2,20,4,18,1,16,1,14,1,13,1,12,1
153 DATA 11,1,10,4
154 FOR x = 1 TO 25: READ nt(x), du(x): NEXT
160 REM puff sound
161 DATA 62,228,211,224,62,240,211,224,17,0,15,27,122,179
162 DATA 32,251,62,255,211,224,201
163 FOR x = 27714 TO 27734: READ mc: POKE x, mc: NEXT
170 REM charge sound effect data
171 DATA 53,2,40,2,32,2,27,2,32,1,27,4,18,2,14,2,11,2,9,2,11,1,9,5
173 FOR x = 1 TO 12: READ nn(x), dd(x): NEXT
180 REM change background color
181 DATA 1,0,7,205,32,253,201
182 FOR x = 27735 TO 27741: READ mc: POKE x, mc: NEXT
190 REM winner sound effect data
191 DATA 29,2,26,2,28,1,26,1,23,1,26,2,23,1,21,2,23,1,21,1
192 DATA 20,2,21,1,20,1,17,2,20,1,21,1,20,1,21,1,23,2,26,2,29,2,26,4
193 FOR x = 1 TO 22: READ wn(x), wd(x): NEXT
200 REM bit image for sprites
210 DATA 192,240,222,211,210,210,211,222,240,192,0,0,0,0,0,0
215 DATA 0,0,1,251,7,7,251,1,0,0,0,0,0,0,0,0
220 DATA 0,0,128,223,224,224,223,128,0,0,0,0,0,0,0,0
225 DATA 3,15,123,203,075,075,203,123,15,3,0,0,0,0,0,0
230 DATA 3,15,31,31,31,31,15,7,3,1,1,0,0,0,1,1
235 DATA 128,224,240,240,240,240,224,192,128,0,0,128,128,128,0,0
```

PUFF LIST continued ...

```
240 DATA 0,0,0,1,7,15,15,31,31,15,15,7,1,0,0,0
245 DATA 0,0,0,128,224,240,240,248,248,240,240,224,128,0,0,0
250 DATA 0,0,0,36,73,146,228,157,67,36,18,9,0,0,0,0
255 DATA 0,0,0,146,36,75,158,124,165,146,73,36,0,0,0,0
260 DATA 5,42,85,42,85,171,87,175,87,175,85,42,85,42,21,2
265 DATA 64,168,84,170,84,234,213,234,213,170,85,170,84,170,84,160
270 DATA 0,0,1,1,3,127,63,31,15,7,7,7,15,12,24,16
275 DATA 128,128,192,192,224,255,254,252,248,240,240,240,120,24,12,4
280 DATA 0,0,0,1,2,5,10,212,56,23,10,5,2,1,0,0
285 DATA 0,0,26,165,202,18,35,78,113,137,69,34,145,89,38,24
290 DATA 0,7,31,63,63,127,127,127,127,127,63,63,31,15,3,0
295 DATA 0,192,240,248,252,252,254,254,254,254,252,252,248,240,192,0
300 DATA 0,127,127,96,96,96,127,127,96,96,96,96,96,96,96,0
305 DATA 0,192,224,96,96,96,224,192,0,0,0,0,0,0,0,0
310 DATA 0,96,96,96,96,96,96,96,96,96,96,96,96,127,63,0
315 DATA 0,48,48,48,48,48,48,48,48,48,48,48,48,240,224,0
320 DATA 0,127,127,96,96,96,127,127,96,96,96,96,96,96,96,0
325 DATA 0,224,224,0,0,0,0,0,0,0,0,0,0,0,0,0
330 DATA 0,127,127,96,96,96,127,127,96,96,96,96,96,96,96,0
335 DATA 0,224,224,0,0,0,0,0,0,0,0,0,0,0,0,0
340 DATA 3,15,31,31,31,31,15,7,3,1,1,2,4,2,1,0
345 DATA 128,224,240,240,240,240,224,192,128,0,0,0,0,0,0,128
350 DATA 3,15,31,31,31,31,15,7,3,1,0,0,0,1,2,4
355 DATA 128,224,240,240,240,240,224,192,128,0,128,64,128,0,0,0
360 DATA 3,15,31,31,31,31,15,7,3,1,1,0,0,0,1,1
365 DATA 128,224,240,240,240,240,224,192,128,0,0,128,128,128,0,0
370 FOR x = 0 TO 511: READ mc: POKE x+55296, mc: NEXT
400 DATA 9,15,5,13,3,8,11
405 FOR x = 0 TO 6: READ gc(x): NEXT
410 DATA one player,two players,instructions,exit PUFF
420 FOR x = 1 TO 4: READ m1$(x): NEXT
430 DATA balloon,cotton ball,feather,lint ball
440 DATA paper star,leaf,table tennis ball
450 FOR x = 1 TO 7: READ ob$(x): NEXT
460 REM sprite collision check
461 DATA 205,35,253,58,99,253,230,32,50,255,255,201
462 FOR x = 27742 TO 27753: READ mc: POKE x, mc: NEXT
470 DATA fast gun movement,moderate gun movement,slow gun movement
472 FOR x = 1 TO 3: READ df$(x): NEXT
500 REM attract mode
510 POKE 17059, 241: POKE 17115, 241: POKE 17126, 113: TEXT: CALL 27600
520 FOR x = 0 TO 3: vt = 4: ht = 96+x*16: nu = x+10: co = 2
525 GOSUB 30000: NEXT
530 FOR x = 0 TO 6: vt = 48+x*20: ht = 16: nu = x+3: co = gc(x)
535 GOSUB 30000: NEXT: POKE 64885, 0
550 VTAB 5: HTAB 1: PRINT "contributed by: DIGITAL EXPRESS"
555 VTAB 7: HTAB 10: PRINT "AUGust 1987": INVERSE
560 FOR x = 1 TO 4: VTAB 9+2*x: HTAB 8: PRINT x; " = "; m1$(x): NEXT
565 NORMAL: x = 1: v1 = 200: v2 = 200: v3 = 200
570 c1 = 12: c2 = 4: c3 = 6
```

PUFF LIST continued ...

```
 600 n1 = PDL(13): n2 = PDL(12)
 605 IF n1 = 1 OR n2 = 1 THEN  CALL 27686: CALL 27600: np = 1: GOTO 1000
 610 IF n1 = 2 OR n2 = 2 THEN  CALL 27686: CALL 27600: np = 2: GOTO 5000
 615 IF n1 = 3 OR n2 = 3 THEN  CALL 27686: CALL 27600: GOTO 20000
 620 IF n1 = 4 OR n2 = 4 THEN  CALL 27686: CALL 27600: GOTO 10000
 630 sb = nt(x)
 632 fb = 128: vo = 144: GOSUB 30100
 634 fb = 162: vo = 176: GOSUB 30100
 636 FOR y = 1 TO 55*du(x): NEXT y
 640 x = x+1: IF x = 26 THEN  x = 1
 650 v1 = v1-2: v2 = v2-4: v3 = v3-3: hh = RND(-y*x)*4
 652 IF v1 < 56 THEN  v1 = 200: c1 = INT(RND(-y+x)*12)+3
 654 IF v2 < 56 THEN  v2 = 200: c2 = INT(RND(-x)*12)+3
 656 IF v3 < 56 THEN  v3 = 200: c3 = INT(RND(-y)*12)+3
 660 vt = v1: ht = 88+hh: nu = 15: co = c1: GOSUB 30000
 662 vt = v2: ht = 120+hh: nu = 16: co = c2: GOSUB 30000
 664 vt = v3: ht = 152+hh: nu = 14: co = c3: GOSUB 30000
 670 GOTO 600
1000 GOSUB 31000: POKE 64885, 0
1005 GOSUB 60000: GOSUB 32000: vg = 0: hg = 112: GOSUB 33000
1010 hf = 0: vf = 2*(rd+lv-1): IF vf > 24 THEN  vf = 24
1020 GOSUB 34000: GOSUB 35000: GOSUB 55100
1030 GOSUB 55200: GOSUB 35200
1040 IF PEEK(64885) = 27 THEN  CALL 27686: RUN
1055 IF PDL(13) = 10 THEN  CALL 27686: GOTO 6000
1060 GOSUB 60000: GOTO 1020
5000 GOSUB 31000: POKE 64885, 0
5005 GOSUB 60000: GOSUB 32000: vg = 0: hg = 112: GOSUB 33000
5010 hf = 0: vf = 2*(rd+lv-1): IF vf > 24 THEN  vf = 24
5020 GOSUB 34000: GOSUB 35000: GOSUB 35100
5030 GOSUB 35300: GOSUB 35200
5040 IF PEEK(64885) = 27 THEN  CALL 27686: RUN
5050 IF PDL(13) = 10 OR PDL(12) = 10 THEN  CALL 27686: GOTO 6000
5060 GOSUB 60000: GOTO 5020
5100 HOME: VTAB 22: HTAB 10: PRINT "END OF GAME": POKE 64885, 0
5110 FOR x = 1 TO 750: NEXT
5115 IF PEEK(64885) = 13 THEN  RUN
5120 GOSUB 33200: FOR x = 1 TO 1500: NEXT
5125 IF PEEK(64885) = 13 THEN  RUN
5130 HOME: PRINT: PRINT " press <return> to restart ..."
5140 FOR x = 1 TO 1500: NEXT
5145 IF PEEK(64885) = 13 THEN  RUN
5150 IF sc(1) <> sc(2) GOTO 5160
5155 HOME: VTAB 22: HTAB 10: PRINT "TIED GAME!!!": GOTO 5200
5160 IF sc(1) < sc(2) GOTO 5170
5165 HOME: VTAB 22: HTAB 7: PRINT "player ONE wins!!!": GOTO 5200
5170 IF np = 2 GOTO 5190
5180 HOME: VTAB 22: HTAB 8: PRINT "COMPUTER wins!!!": GOTO 5200
5190 HOME: VTAB 22: HTAB 7: PRINT "player TWO wins!!!"
5200 FOR x = 1 TO 22: sb = wn(x)
5210 fb = 128: vo = 144: GOSUB 30100
5220 fb = 163: vo = 176: GOSUB 30100
5230 FOR y = 1 TO 95*wd(x): NEXT y: NEXT x
5240 CALL 64851: IF PEEK(64885) = 13 THEN  RUN
5250 GOTO 5100
```

PUFF LIST continued ...

```
 6000 HOME: VTAB 22: HTAB 10: PRINT "GAME PAUSED": POKE 64885, 0
 6010 FOR x = 1 TO 750: NEXT
 6020 IF PEEK(64885) = 13 GOTO 6100
 6030 GOSUB 33200: FOR x = 1 TO 1500: NEXT
 6040 IF PEEK(64885) = 13 GOTO 6100
 6050 HOME: PRINT: PRINT " press <return> to continue ..."
 6060 FOR x = 1 TO 1500: NEXT
 6070 IF PEEK(64885) = 13 GOTO 6100
 6080 GOTO 6000
 6100 CALL 27686: GOSUB 33200: ON np = 1 GOTO 1020: GOTO 5020
10000 TEXT: PRINT " Type NEW before programming."
10010 POKE 16953, 95: POKE 16134, 3: END
20000 POKE 17059, 23: POKE 17115, 23: TEXT
20010 PRINT " You and your opponent are in"
20020 PRINT " a room with an opening in the"
20030 PRINT " ceiling.  A gust of wind blows";
20040 PRINT " lightweight objects into this"
20050 PRINT " room.": PRINT
20060 PRINT " When an object (there are 7)"
20070 PRINT " drifts too close to the floor"
20080 PRINT " the sensor blowers push it"
20090 PRINT " back up. As it gets too close"
20100 PRINT " to the ceiling the incoming"
20110 PRINT " breeze propels it downward"
20120 PRINT " again.": PRINT
20130 PRINT " You and your opponent each"
20140 PRINT " have an air gun.  The goal is"
20150 PRINT " is to push the objects into"
20160 PRINT " your opponent's wall.  The"
20170 PRINT " points awarded are directly"
20180 PRINT " proportional to the distance"
20190 PRINT " from the floor at the point"
20200 PRINT " of impact.": PRINT
20210 HTAB 5: PRINT "press any key for more"; : GET go$: CALL 27686
20300 HOME: PRINT " To start with, you have enough";
20310 PRINT " energy in your gun for 100 air";
20320 PRINT " blasts.  When the object hits"
20330 PRINT " your opponent's wall you are"
20340 PRINT " awarded 10 additional blasts"
20350 PRINT " (not to exceed 100).": PRINT
20360 PRINT " The joysticks control air gun"
20370 PRINT " movement.  Either trigger"
20380 PRINT " shoots an air blast.  If you"
20390 PRINT " exhaust your supply of shots,"
20400 PRINT " you are virtually powerless.": PRINT
20410 PRINT " During the game, you can"
20420 PRINT " suspend action by pressing the";
20430 PRINT " '*' on the game keypad.": PRINT
20440 PRINT " With the single player option"
20450 PRINT " you vie against ADAM.  The"
20460 PRINT " computer plays rather well.": PRINT: PRINT: PRINT
20470 HTAB 5: PRINT "press any key for more"; : GET go$: CALL 27686
```

PUFF LIST continued ...

```
20500 HOME: PRINT " Player one has the air gun"
20510 PRINT " on the left wall.  The second"
20520 PRINT " player has the gun on the"
20530 PRINT " right wall.": PRINT
20540 PRINT " As the game progesses, the"
20550 PRINT " incoming breeze gets"
20560 PRINT " stronger; and, the sensor"
20570 PRINT " blowers are more powerful.": PRINT
20580 PRINT " The closer an object is to"
20590 PRINT " your gun, the stronger your"
20600 PRINT " air blast will be.  Also,"
20610 PRINT " you have twice the normal"
20620 PRINT " blast potential with more than";
20630 PRINT " 75 shots remaining.  And, if"
20640 PRINT " the object hits the right spot";
20650 PRINT " on your air gun, the object"
20660 PRINT " will be pushed in the opposite";
20670 PRINT " direction.": PRINT
20680 PRINT " Enjoy PUFF ..."
20700 VTAB 24: HTAB 5: PRINT "press any key for menu";
20710 GET go$: CALL 27686: GOTO 500
30000 REM move sprite
30010 POKE 27643, vt: POKE 27644, ht: POKE 27646, nu: POKE 27647, co
30020 CALL 27642: RETURN
30100 POKE 27682, fb: CALL 27681: POKE 27682, sb: CALL 27681
30110 POKE 27682, vo: CALL 27681: RETURN
30200 HPLOT 16, x TO 20, x: HPLOT 236, x TO 240, x: RETURN
31000 GOSUB 57000: POKE 25431, 1: POKE 25471, 16: POKE 25568, 240: HGR
31010 FOR x = 0 TO 31: HCOLOR  = 4: GOSUB 30200: NEXT
31020 FOR x = 32 TO 63: HCOLOR  = 5: GOSUB 30200: NEXT
31030 FOR x = 64 TO 95: HCOLOR  = 2: GOSUB 30200: NEXT
31040 FOR x = 96 TO 127: HCOLOR  = 10: GOSUB 30200: NEXT
31050 FOR x = 128 TO 159: HCOLOR  = 8: GOSUB 30200: NEXT
31100 HCOLOR  = 15: HPLOT 24, 0 TO 112, 0: HPLOT 231, 0 TO 128, 0
31110 HCOLOR  = 12: FOR x = 32 TO 228 STEP 8
31120 HPLOT x, 155 TO x-4, 159: HPLOT x, 155 TO x+4, 159
31130 NEXT: RETURN
32000 f1 = 72: f2 = 72: CALL 27600: GOSUB 32100: GOSUB 32200: RETURN
32100 REM move player one nozzle
32110 vt = f1: ht = 21: nu = 1: co = 15: GOTO 30000
32200 REM move player two nozzle
32210 vt = f2: ht = 219: nu = 2: co = 15: GOTO 30000
33000 rd = rd+1: IF rd > 7 THEN  rd = 1: lv = lv+1
33005 vt = vg: ht = hg: nu = rd+2: co = gc(rd-1): GOSUB 30000
33010 HOME: VTAB 22: HTAB 12: PRINT "LEVEL "; lv
33020 rx$ = "ROUND "+STR$(rd)+" ("+ob$(rd)+")"
33030 VTAB 24: HTAB 16-LEN(rx$)/2: PRINT rx$;
33100 FOR x = 1 TO 12: sb = nn(x)
33110 fb = 130: vo = 146: GOSUB 30100
33120 fb = 164: vo = 178: GOSUB 30100
33130 FOR y = 1 TO 100*dd(x): NEXT y: NEXT x: CALL 64851
```

PUFF LIST continued ...

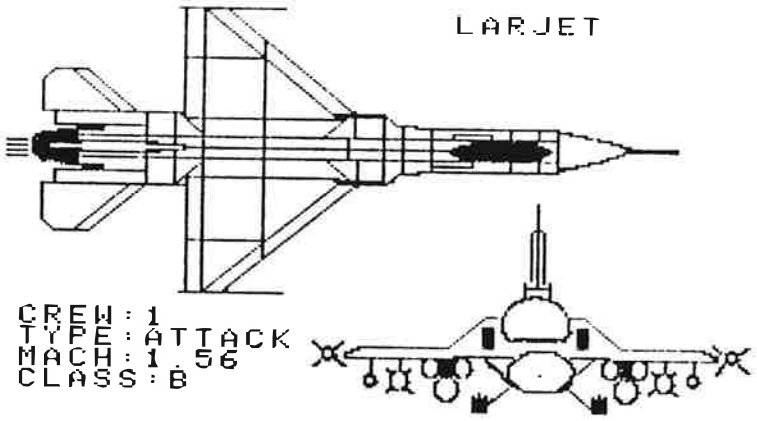
```
33200Ø HOME: VTAB 22: PRINT " SHOTS:"; eg(1)
33210Ø PRINT " SCORE:"; sc(1)
33220Ø VTAB 22: HTAB 18: PRINT "SHOTS:"; eg(2)
33230Ø VTAB 23: HTAB 18: PRINT "SCORE:"; sc(2): RETURN
34000Ø vg = vg+vf: hg = hg+hf
34002 IF vg < Ø THEN  vg = Ø: vf = 2*(rd+lv-1): IF vf > 24 THEN  vf = 24
34004 IF vg > 136 THEN  vf = -(rd+lv-1): vg = 136: GOSUB 61000
34006 vt = vg: ht = hg: co = gc(rd-1): nu = rd+2: GOSUB 30000
34010 ON hg < 19 GOTO 34500: ON hg > 223 GOTO 34600
34020 RETURN
34500 GOSUB 40100: GOSUB 40000: sc(2) = sc(2)+sc
34505 IF eg(1) = Ø GOTO 34520
34510 eg(2) = eg(2)+10: IF eg(2) > 100 THEN  eg(2) = 100
34520 CALL 27600: POP: ON np = 1 GOTO 1005: GOTO 5005
34600 GOSUB 40100: GOSUB 40000: sc(1) = sc(1)+sc
34605 IF eg(2) = Ø GOTO 34520
34610 eg(1) = eg(1)+10: IF eg(1) > 100 THEN  eg(1) = 100
34620 CALL 27600: POP: ON np = 1 GOTO 1005: GOTO 5005
35000 IF PDL(5) = Ø THEN  RETURN
35005 IF PDL(5) = 1 THEN  f1 = f1-e1: GOTO 35020
35010 IF PDL(5) = 4 THEN  f1 = f1+e1: GOTO 35020
35015 RETURN
35020 IF f1 < 2 THEN  f1 = 2
35030 IF f1 > 142 THEN  f1 = 142
35040 IF hg < 34 AND ABS(vg-f1) < 6 THEN  GOSUB 63000
35050 GOSUB 32110: jw = 1: GOTO 35250
35100 IF PDL(4) = Ø THEN  RETURN
35105 IF PDL(4) = 1 THEN  f2 = f2-e2: GOTO 35120
35110 IF PDL(4) = 4 THEN  f2 = f2+e2: GOTO 35120
35115 RETURN
35120 IF f2 < 2 THEN  f2 = 2
35130 IF f2 > 142 THEN  f2 = 142
35140 IF hg > 200 AND ABS(vg-f2) < 6 THEN  GOSUB 63100
35150 GOSUB 32210: jw = 1: GOTO 35250
35200 IF PDL(7) = Ø AND PDL(9) = Ø THEN  RETURN
35201 IF eg(1) = Ø THEN  RETURN
35202 HCOLOR  = 15: HPLOT 40, f1+6 TO 46, f1+6: CALL 27714
35204 POKE 16777, 128: HPLOT 40, f1+6 TO 46, f1+6
35205 q1 = 1: IF eg(1) > 75 THEN  q1 = 2
35206 eg(1) = eg(1)-1: IF eg(1) < Ø THEN  eg(1) = Ø
35208 VTAB 22: HTAB 8: PRINT eg(1); " ";
35209 tx = ABS(f1-vg): IF tx > 12 THEN  RETURN
35210 IF hg < 60 THEN  hf = hf+4*q1: GOTO 35250
35220 IF hg < 100 THEN  hf = hf+2*q1: GOTO 35250
35230 IF hg > 180 THEN  RETURN
35240 hf = hf+1*q1
35250 IF hf > 16 THEN  hf = 12
35252 IF hf < -16 THEN  hf = -12
35254 IF jw = 1 THEN  jw = Ø: RETURN
35256 IF hf = Ø THEN  hf = -1: IF RND(1) < .5 THEN  hf = 1
35258 GOTO 34000
```

PUFF LIST continued ...

```
35300 IF PDL(6) = 0 AND PDL(8) = 0 THEN  RETURN
35301 IF eg(2) = 0 THEN  RETURN
35302 HCOLOR  = 15: HPLOT 216, f2+6 TO 210, f2+6: CALL 27714
35304 POKE 16777, 128: HPLOT 216, f2+6 TO 210, f2+6
35305 q2 = 1: IF eg(2) > 75 THEN  q2 = 2
35306 eg(2) = eg(2)-1: IF eg(2) < 0 THEN  eg(2) = 0
35308 VTAB 22: HTAB 24: PRINT eg(2); " ";
35309 tx = ABS(f2-vg): IF tx > 12 THEN  RETURN
35310 IF hg > 180 THEN  hf = hf-4*q2: GOTO 35250
35320 IF hg > 130 THEN  hf = hf-2*q2: GOTO 35250
35330 IF hg < 60 THEN  RETURN
35340 hf = hf-1*q2: GOTO 35250
40000 REM collision sound effect
40010 FOR x = 0 TO 2: POKE 27736, x*2+2: CALL 27735: POKE 27724, 255
40020 POKE 27715, 228+x: CALL 27714: NEXT: POKE 27724, 15
40030 POKE 27736, 1: CALL 27735: POKE 27715, 228: RETURN
40100 sc = (5-INT(vg/32))*10: HOME
40110 VTAB 22: HTAB 12: PRINT sc; " points": RETURN
55100 rj = RND(1): IF rj < -.15 THEN  RETURN
55110 ab = vg-f2: IF ABS(ab) < 8 THEN  RETURN
55120 IF ab < 1 THEN  f2 = f2-8: GOTO 35120
55130 f2 = f2+10: GOTO 35120
55200 IF hg > 180 AND ABS(ab) < 8 AND hf > 0 GOTO 35301
55201 IF hg > 200 AND ABS(ab) < 8 GOTO 35301
55202 IF hg = 112 AND rj > .3 AND ABS(ab) < .8 GOTO 35301
55204 IF eg(1) = 0 AND ABS(ab) < .8 AND rj > .3 GOTO 35301
55205 IF rj > .21 THEN  RETURN
55210 IF ABS(vg-f2) > 8 THEN  RETURN
55220 IF hg < 30 THEN  RETURN
55230 IF hg > 120 AND rj > .2 GOTO 35301
55240 IF eg(2) < 25 AND rj > .05 THEN  RETURN
55250 IF rj > .01 THEN  RETURN
55260 GOTO 35301
57000 POKE 17059, 27: POKE 17115, 27: TEXT
57010 VTAB 6: HTAB 4: PRINT "Player ONE difficulty?": VTAB 8
57020 FOR x = 1 TO 3: HTAB 4: PRINT x; " = "; df$(x): NEXT
57030 kp = PDL(13): IF PEEK(64885) = 27 THEN  RUN
57040 IF kp >= 1 AND kp <= 3 GOTO 57060
57050 GOTO 57030
57060 CALL 27686: e1 = (4-kp)*3
57070 POKE 17059, 241: POKE 17115, 241: IF np = 1 THEN  RETURN
57100 HOME: VTAB 6: HTAB 4: PRINT "Player TWO difficulty?": VTAB 8
57110 FOR x = 1 TO 3: HTAB 4: PRINT x; " = "; df$(x): NEXT
57120 kp = PDL(12): IF PEEK(64885) = 27 THEN  RUN
57130 IF kp >= 1 AND kp <= 3 GOTO 57150
57140 GOTO 57120
57150 CALL 27686: e2 = (4-kp)*3: RETURN
```

PUFF LIST continued ...

```
60000 IF eg(1) = 0 AND eg(2) = 0 THEN  POP: GOTO 5100
60010 IF eg(1) = 0 AND sc(2) > sc(1) THEN  POP: GOTO 5100
60020 IF eg(2) = 0 AND sc(1) > sc(2) THEN  POP: GOTO 5100
60030 RETURN
61000 hv = hg+8: IF hv < 32 THEN  hv = 32
61010 HCOLOR  = 15: IF hv > 224 THEN  hv = 224
61020 HPLOT hv, 154 TO hv, 148: POKE 16777, 128
61030 CALL 27714: HPLOT hv, 154 TO hv, 148: RETURN
62000 POKE 27724, 50: FOR x = 0 TO 4: POKE 27715, 224+x
62010 CALL 27714: NEXT
62020 POKE 27715, 228: POKE 27724, 15: RETURN
63000 CALL 27742: IF PEEK(65535) <> 32 THEN  RETURN
63010 hf = hf+8: GOTO 62000
63100 CALL 27742: IF PEEK(65535) <> 32 THEN  RETURN
63110 hf = hf-8: GOTO 62000
```



PRODUCT:	"THE BEATLES" & "POTPOURRI"
MANUFACTURER:	VideoSongs
MEDIA TYPE:	DDP / disk
GRAPHICS/SOUND/DESIGN:	96
INSTRUCTIONS:	n/a
USEFULNESS vs. PRICE:	96
RECOMMENDATION:	highly recommended
PRICE:	$11.50 / $9.50
RATED BY:	N&B staff

These two packages "THE BEATLES" and "POTPOURRI" are two excellent collections of songs to be used with the VideoTunes program (by FutureVision). The "POTPOURRI" volume comes with 18 all time favorites. These include "Honey", the "Baby Elephant Walk", and the "Star Spangle Banner". "THE BEATLES" comes with 16 songs including "Yesterday", "A Hard Days Night", and "Let It Be".

All the tunes are very nicely done and sound GREAT. These two packages show off both the sound capabilities of ADAM and the composition potential of VideoTunes. Each volume offers plenty of pleasurable listening. The individual price of each volume is very reasonable and VideoSongs (the manufacturer) also offers a special price if you purchase both at the same time: $15.50 for two disks or $18.50 for two data packs.

PRODUCT:	STAGE FRIGHT
MANUFACTURER:	Reedy Software
MEDIA TYPE:	DDP / disk
GRAPHICS/SOUND/DESIGN:	93/93/98
INSTRUCTIONS:	94
USEFULNESS vs. PRICE:	96
RECOMMENDATION:	highly recommended
PRICE:	$15.95 / $13.95
RATED BY:	N&B staff

STAGE FRIGHT is Reedy's latest release. In this rather complex text adventure you play the role of an actor or actress (you choose) trapped in an abandoned theater. To start with your goal is to get out of the theater using clues (some very subtle) and tools (that you find and "take" along the way). Upon certain accomplishments, you are entertained with melodies and graphics displays. When you've completed this round (which is no small feat), you can go back in and find a treasure. In the third (final) round you are sent back in to perform a rescue.

To move within the theater, you use the arrow keys on the keyboard. To travel up or down steps, you just press the <home> key and the corresponding arrow key. Many of the Word Processing function keys are used for assorted controls. To do things such as picking up a knive, opening a door, or eating a hotdog (?), you just type in the command. And, you can store up to SIX partially completed games on a single medium. Humor pervades the entire game with subtle and obvious puns (some are clues); it's very challenging and a lot of fun too.

And ... STAGE FRIGHT even comes with a "cheat sheet". But even this requires some decoding of the enigmatic tips. If you're new to text adventures, the complexity of this one could be a little frustrating at first. But, intermediate and advanced gamers will take an immediate liking to it. STAGE FRIGHT is a *very good* value for the price.

Clipper

by DIGITAL EXPRESS

Clipper, the latest commercial software release from DIGITAL EXPRESS, introduces the exciting concept of "clip art" to ADAM. This totally machine code program is enhanced with ALL the sophisticated graphics that you get with ALL of our second generation of ADAM software. You will see a graphics title screen as soon as you pull reset. A simple melody plays while the program loads into memory. Of course, you'll see the SmartKEYs at the bottom of the screen just like Coleco's software.

Clipper allows you to draw (background and foreground), edit, store, and retrieve clip art files (64 pixels across by 64 pixels down). You can also "capture" clip art from hi-res graphics screens stored in SmartPAINT format. If you don't already have SmartPAINT (from DIGITAL EXPRESS), a program is included that permits you to store your own or RLE pictures in the format. And, the program includes an 11K ramdisk which does not require the 64K memory expander.

Clipper comes complete with a detailed, easy - to - understand user's guide that tells you how to make the most of the program. Plus, it even includes instructions on how to use clip art in your own graphics screens. How much is this breakthrough program? The retail price is only $19.95; and, "NIBBLES & BITS" subscribers can get it at the discount price of just $14.95. Clipper provides access to an entirely new concept for ADAM!!

DIGITAL EXPRESS ... quality you CAN depend on!!!

ADAM USERS' GROUPS

ALABAMA

James E. Gilbert
4608 Lakeview Drive
Hunstville, AL 35810

Gulf Coast ADAM Users Club
6665 Timbers Drive
Mobile, AL 36609

Victor L. Watford
P.O. Box 777
Russellville, AL 35653

Montgomery ADAM Users Group
Mike Heuerman
654 Sarver Avenue
Montgomery, AL 36109

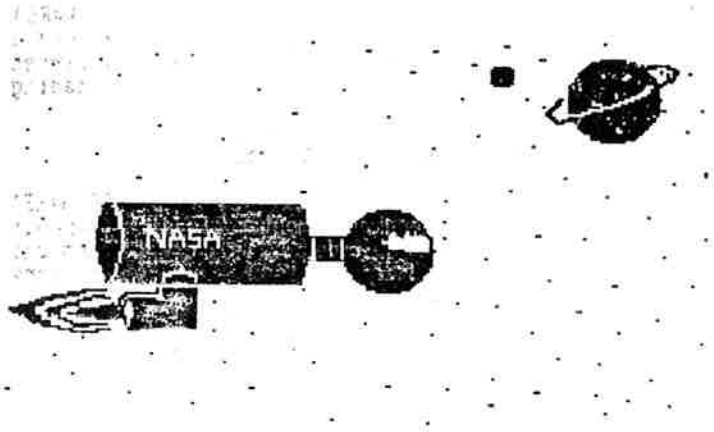
ALASKA

Richard Baines
7210 Bulen Drive
Anchorage, AK 99507

ARIZONA

Dan Levitt
4525 South White Pine
Tucson, AZ 85730

Robert R. Marentes
9425 North 38th Avenue
Phoenix, AZ 85021



BULLETIN BOARD

ORPHANWARE Service Center
Attn: Thom Andree
5641 Leibold
Huber Heights, OH 45424

ADAM data drives
Terry Fowler
Route 2, Box 2756
Pearland, TX 77581

New ADAM Software
Walters Software
Route 4, Box 289 - A
Titusville, PA 16354

New newsletter: ADAM ALIVE
E & T SOFTWARE
1010 Westminster
Garland, TX 75040

ADAM Software
GJMG Enterprises
5558 Hollywood Avenue
Maple Heights, OH 44137

Song Files for VideoTunes
Send SASE to:
VIDEOSONGS
Box 151
Harmony, IN 47853

game controller extension cables
D.L. DECKER ENTERPRISES
Route 2, Box 15
Spring Mills, PA 16875-9720

Surplus ADAM peripherals
H & R CORPORATION
401 East Erie Avenue
Philadelphia, PA 19134
(215) 426-1708

Looking for other ADAM users
Anthony Yulo
111-2 Spruce Street
Minot AFB, ND 58704

ADAM Software and Hardware
American Design Components
62 Joseph Street
Moonachie, NJ 07074
toll free: (800) 524-0809
in NJ: (201) 939-2710

PROGRAMMING UTILITY SOFTWARE

☐☐☐ Intel-BEST 3.3 (by DIGITAL EXPRESS)
* makes over three dozen changes to SmartBASIC V1.0; includes nine very user friendly MUSIC commands

>>> $24.95 (each) retail price
>>> $18.95 (each) for N&B subscribers

☐☐☐ Intel-LOAD V1.0 (by DIGITAL EXPRESS)
* converts BASIC 1.0 programs to LOAD up to 12 times faster; stays in RAM; onscreen help; two BSAVE options

>>> $15.95 (each) retail price
>>> $11.95 (each) for N&B subscribers

☐☐☐ Intel-LOAD V2.0 (by DIGITAL EXPRESS)
* converts BASIC 2.0 programs to LOAD up to 12 times faster; stays in RAM; onscreen help; two BSAVE options; works only in STDMEM

>>> $15.95 (each) retail price
>>> $11.95 (each) for N&B subscribers

☐☐☐ SmartBEST V1.0 (by DATA DOCTOR)
* makes several changes to SmartBASIC V1.0; not compatible with Intel-BEST 3.3

>>> $16.95 (each) retail price
>>> $14.95 (each) for N&B subscribers

☐☐☐ SmartTRIX I (by DATA DOCTOR)
* a set of 10 user friendly programming aids; two very nice sprite programs; 60 page manual; disk and DDP versions not compatible

>>> $29.95 (each) retail price
>>> $24.95 (each) for N&B subscribers

☐☐☐ BASICaide (rev2) (Mr. T. Software)
* several SmartBASIC 1.0 enhancements including a new "CHAIN" command for merging programs and a new "BIN" command that executes the built-in function for converting SmartBASIC 1.0 programs to LOAD up to 12 times faster

>>> $11.95 (each) retail price
>>> $9.95 (each) for N&B subscribers

☐☐☐ TurboDISK 1.0 (by DIGITAL EXPRESS)
* creates a ramdisk ability from SmartBASIC V1.0; corrects INIT blocks and BSAVE short buffer; includes TurboCOPY -- a utility for controlling files and copying copy buffer

>>> $24.95 (each) retail price
>>> $19.95 (each) for N&B subscribers

☐☐☐ FontPOWER (by DIGITAL EXPRESS)
* utility using Coleco-like graphics for designing your own font sets; 8 font sets including "script", "roman", "cory", & "bold"; shows you how to use font sets in high or low resolution graphics; plus three font shape tables for use in HGR or HGR2 mode

>>> $16.95 (each) retail price
>>> $12.95 (each) for N&B subscribers

☐☐☐ SpritePOWER (by DIGITAL EXPRESS)
* utility using Coleco-like graphics for designing your own sprites; includes three sets of sprites; extensive instruction manual; shows you how to use the sprites in your own programs; does NOT require the 64K RAM card; totally machine code program (36K)

>>> $19.95 (each) retail price
>>> $14.95 (each) for N&B subscribers

☐☐☐ MegaUtil (by MARATHON COMPUTER PRESS)
* an excellent collection of varied programming aids; includes ByteWriter (block editor), CopyWriter (media back-up utility), PD modules, programming tips, more +++

>>> $32.95 (each) retail price
>>> $27.95 (each) for N&B subscribers

☐☐☐ TurboDISK 2.0 (by DIGITAL EXPRESS)
* creates a powerful ramdisk ability for SmartBASIC 2.0

>>> $15.95 (each) retail price
>>> $11.95 (each) for N&B subscribers

☐☐☐ Clipper (by DIGITAL EXPRESS)
* introduces the concept of "clip art" to ADAM; totally machine code program; clip a section from one hi-res screen and put it on another (in your own programs); evens includes an option to let you draw clip art

>>> $19.95 (each) retail price
>>> $14.95 (each) for N&B subscribers

RECREATION/GAMES SOFTWARE

☐☐☐ MageQuest (rev 2) (by REEDY SOFTWARE)
* superb graphic adventure; includes 9 levels of play in the main adventure plus 3 solo adventures; additional solo adventures are available from REEDY SOFTWARE

>>> $16.95 (each) retail price
>>> $14.95 (each) for N&B subscribers

☐☐☐ TRIVIAPAC I (by Mr. T. Software)
* 1200 questions; 6 categories; one to four players; graphics and sound; many hours of fun; DDP version only

>>> $17.95 (each) retail price
>>> $14.95 (each) for N&B subscribers

☐☐☐ KID'S TRIVIAPAC (by Mr. T. Software)
* 1080 questions; 6 categories; one to four players; graphics and sound; many hours of fun; DDP version only

>>> $17.95 (each) retail price
>>> $14.95 (each) for N&B subscribers

☐☐☐ Strategy Strain (by DATA DOCTOR)
* nine intellectually challenging computer classics; graphics and sound; superb Star Trek adventure

>>> $18.95 (each) retail price
>>> $14.95 (each) for N&B subscribers

☐☐☐ Lab Mouse (by REEDY SOFTWARE)
* exciting game that puts you in the role of a laboratory mouse stuck in a maze; all hi-res graphics; five skill levels; auto-loading

>>> $13.95 (each) retail price
>>> $11.95 (each) for N&B subscribers

☐☐☐ Entertainment Pack (by REEDY SOFTWARE)
* three challenging computer classics (connect 4, blockade, and slide puzzle); great graphics; fast animated sprites; one or two players

>>> $16.95 (each) retail price
>>> $14.95 (each) for N&B subscribers

GUIDES/BOOKS/INSTRUCTIONS

☐☐☐ The Hacker's Guide to ADAM (vol one)
* Ben Hinkle's in-depth guide to the technical aspects of exploring ADAM; 60 pages; 18 programs

>>> $12.95 (each) retail price
>>> $10.95 (each) for N&B subscribers

☐☐☐ The Hacker's Guide to ADAM (vol two)
* Ben Hinkle's detailed guide to SmartBASIC V1.0; 110 pages; HELLO program includes several BASIC enhancements

>>> $12.95 (each) retail price
>>> $10.95 (each) for N&B subscribers

☐☐☐ Hacker's Guide software (by Ben Hinkle)
* all the programs from volumes one and two

>>> $5.95 (each) retail price
>>> $4.95 (each) for N&B subscribers

☐☐☐ EZ Ref 101 (by DIGITAL EXPRESS)
* approximately 700 Z80 instructions listed in NUMERICAL sequence; 9 pages; decimal, hex, op codes, operands

>>> $2.45 (each) retail price
>>> $1.95 (each) for N&B subscribers

☐☐☐ EZ Ref 102 (by DIGITAL EXPRESS)
* approximately 700 Z80 instructions listed in ALPHABETICAL sequence; 9 pages; decimal, hex, op codes, operands

>>> $2.45 (each) retail price
>>> $1.95 (each) for N&B subscribers

☐☐☐ Pinball Construction/HardHat Mac Guides
* 40 pages of instructions for the popular public domain package

>>> $2.45 (each) retail price
>>> $1.95 (each) for N&B subscribers

MISCELLANEOUS UTILITY SOFTWARE

☐☐☐ ShowOFF I (by DIGITAL EXPRESS)
* self-booting graphics design package (enter text, draw polygons, save pictures, etc.) with a variety of print options (preset for Epson FX / IBM 5152 printer codes); printing graphics requires a Centronics parallel interface for printer

>>> $29.95 (each) retail price
>>> $24.95 (each) for N&B subscribers

☐☐☐ ShowOFF II (by DIGITAL EXPRESS)
* machine code print enhancements for SmartWriter (adds CONTROL features to SmartWriter) and SmartBASIC; requires Centronics parallel interface, a Panasonic KX 1080 or 1080i printer, and a 64K expander

>>> $19.95 (each) retail price
>>> $14.95 (each) for N&B subscribers

☐☐☐ ShowOFF IIa (by DIGITAL EXPRESS)
* very similar to ShowOFF II except that it is compatible with any dot matrix printer that supports EPSON FX escape codes; works with the EPSON and STAR line of printers and the Okimate 20; does not include line justification commands or internal document margin control

>>> $19.95 (each) retail price
>>> $14.95 (each) for N&B subscribers

"NIBBLES & BITS" SOFTWARE

☐☐☐ N&B binder set 01 (by DIGITAL EXPRESS)
* all six issues from 07/86 thru 12/86 in a sturdy 3-ring binder; includes two DDP's or two disks containing all the programs

>>> $29.95 (each) retail price
>>> $24.95 (each) for N&B subscribers

☐☐☐ N&B binder set 02 (by DIGITAL EXPRESS)
* all six issues from 01/87 thru 06/87 in a sturdy 3-ring binder; includes two DDP's or two disks containing all the programs

>>> $29.95 (each) retail price
>>> $24.95 (each) for N&B subscribers

☐☐☐ N&B issue programs (by DIGITAL EXPRESS)
* set 01: all the programs from 07/86 thru 09/86
* set 02: all the programs from 10/86 thru 12/86
* set 03: all the programs from 01/87 thru 03/87
* set 04: all the programs from 04/87 thru 06/87

>>> $9.95 (each) retail price
>>> $4.95 (each) for N&B subscribers

COLECO COPYRIGHTED SOFTWARE

☐☐☐ SmartLOGO (data pack only)
* Coleco's version of the popular language; 350 ++ page manual

>>> $27.95 (each) retail price
>>> $22.95 (each) for N&B subscribers

☐☐☐ SmartFiler (data pack only)
* Coleco's general purpose database program; 38 page manual

>>> $18.95 (each) retail price
>>> $15.95 (each) for N&B subscribers

MISCELLANEOUS SUPPLIES

□□□ Coleco/LORAN digital data packs
* designed and formatted by Loranger Manufacturing

>>> $4.95 (each) retail price
$39.95 (for 10) retail price
>>> $3.95 (each) for N&B subscribers
$33.95 (for 10) for N&B subscribers

□□□ Plain Label digital data packs
* Sony brand formatted by E & T SOFTWARE

>>> $3.95 (each) retail price
$33.95 (for 10) retail price
>>> $2.45 (each) for N&B subscribers
$18.95 (for 10) for N&B subscribers

□□□ Plain Label 5.25" disks for ADAM
* double sided, double density, with envelope

>>> $.79 (each) retail price
$6.95 (for 10) retail price
>>> $.49 (each) for N&B subscribers
$4.25 (for 10) for N&B subscribers

□□□ SmartWRITER printer ribbons
* black ink, just like the one that came with your ADAM

>>> $5.75 (each) retail price
$15.95 (for 3) retail price
>>> $5.25 (each) for N&B subscribers
$14.75 (for 3) for N&B subscribers

□□□ Panasonic printer ribbons
* black ink, nylon, approximately one million characters, fits these models: 1080, 1080i, 1090, 1091, 1091i, and 1092

>>> $6.95 (each) retail price
>>> $5.45 (each) for N&B subscribers

□□□ multipurpose adhesive labels
* white, tractor feed, 3 1/2 x 15/16, fan fold, single column

>>> $2.95 (for 500) retail price
$5.45 (for 1000) retail price
>>> $2.25 (for 500) for N&B subscribers
$3.95 (for 1000) for N&B subscribers

□□□ word processing computer paper
* white, tractor feed, 9 1/2 x 11, fan fold, 20 lb. wt., clean edge, one part, single column

>>> $4.25 (250 sheets) retail price
>>> $3.45 (250 sheets) for N&B subscribers

EDUCATIONAL SOFTWARE

□□□ The Spanish Vocabularian
(by MARATHON COMPUTER PRESS)
* a unique program for ADAM; includes electronic dictionary; includes 1600 words; expandable to 7400 words; quizzes; printed study sheets; report cards

>>> $18.50 (each) retail price
>>> $15.95 (each) for N&B subscribers

□□□ Quikfax Quest (by DIGITAL EXPRESS)
* three academic quizzes; includes study mode (on - screen and hardcopy); US capitals, world capitals, and Chemistry elements

>>> $18.95 (each) retail price
>>> $14.95 (each) for N&B subscribers

■■■■■■■■■■■

□□□□□ Unless otherwise noted, all software is available on disk or datapack.

□□□□□ All DIGITAL EXPRESS media is warrantied to be free from defects in materials and workmanship. If the storage medium proves defective at any time, return it to us for repair or replacement (at our descretion).

□□□□□ The product prices listed herein may be subject to change after September 30, 1987.

■■■■■■■■■■■

SPECIAL OFFER

Until September 30th, any product order with DIGITAL EXPRESS with a sublotal greater than $19.99 gets a **FREE**, unformatted, blank disk. Be sure to mention this offer with your order.

DEI Public Domain Facts

You may get any of the volumes described below on DATA PACK or DISK for ONLY $5.95. Subscribers also have an option to get a volume FREE (limit three per calendar month); this option does NOT apply to the volumes in the "Coleco Unreleased Titles Library".

Here's how to get one FREE. (1) Contribute an original program for any library. (2) Send a signed statement that the program is NOT copyrighted. (3) Send the program on DDP (digital data pack) or disk; one DDP or disk for each volume that you want to exchange. And, (5) include a return mailer with sufficient postage or send $2.50 for shipping costs.

Public domain software is offered as a quick, inexpensive means for you to expand your ADAM software library. Note, however, that public domain software is not necessarily of commercial quality. Although we do attempt to winnow out flawed programs, there is no guarantee of quality regarding these packages.

SmartBASIC V1.0 LIBRARY

You must boot your own SmartBASIC first in order to use the volumes in this library. All programs will speed load. Each volume (except the utility volumes) is controlled by a user friendly ramdisk (does NOT require the 64K expander) central menu.

"N&Bgames01": An assortment of text adventures, board games, and animation games -- 130K of files.

"N&Bgames02": An assortment of text adventures, board games, and animation games -- 154K of files.

"N&Bgraph01": A variety of graphics displays and music programs -- 88K of files.

"N&Bmath01": Several scientific and financial math programs -- 114K of files.

"N&Butil01": Intended for more advanced programmers this volume includes programming utilities -- 108K of files.

SmartPAINT Files LIBRARY

In order to view/use the volumes in this library you should have SmartPAINT (from ShowOFF I) or the HGR Picture Manager program in the February 1987 issue of "NIBBLES & BITS" (page 16).

"N&Bpix001": 13 different HGR picture files.
"N&Bpix002": 13 different HGR picture files.
"N&Bpix003": 13 different HGR picture files.
"N&Bpix004": 13 different HGR picture files.
"N&Bpix005": 13 different HGR picture files.
"N&Bpix006": 13 different HGR picture files.
"N&Bpix007": 13 different HGR picture files.
"Art Gallery 2": 13 picture files of Smurf - like characters. Compiled by REEDY SOFTWARE.

Coleco Unreleased Titles LIBRARY

"SmartBASIC 2.0": Improved interpreter; 49K program; works with or without the 64K expander; includes new commands STDMEM, EXTMEM, MERGE; plus more ...

"Pinball Construction/Hardhat Mac": Best of Electronic Arts; latest version with two demo pinball games; 1 to 4 players with Pinball Construction; one or two players with Hardhat Mac.

"ADAMLink II": Supports uploading and down loading of SmartWriter compatible files; includes U/D instructions; requires the ADAMLink modem.

"Jeopardy": The extremely popular ADAM game; just like the game show; great graphics; hall of fame; one to three players.

"Super SubRoc:" 90K arcade-type game; super graphics; hall of fame.

"Troll's Tale": Easy to play graphic/text adventure; supports one player; disk and DDP versions NOT compatible.

"Video Hustler": Graphic billiards game; one or two players; from an unreleased cartridge.

CP/M 2.2 LIBRARY

The volumes in this library require that you boot your own CP/M 2.2 package first.

"CP/Mgames01": 30 games.

"CP/Mgames02": 25 games.

"demo carts": requires 64K expander; music samples, system tester, Coleco software demo cart, Coleco back - up utility, plus more.

Pinball Games LIBRARY

Each volume in this library is self-booting or may be used with the Pinball Construction Set.

"N&B-PBgames01": 10 pinball games.

"N&B-PBgames02": 10 pinball games.

Miscellaneous Collections LIBRARY

"MWplus01": A collection of improvements to MultiWrite by Strategic Software. Requires MultiWrite software. Written by Jim Guenzel.

"N&Bacalc01": several paradigm and other files stored in ADAMcalc format; contributed by Terry Fowler; 148K of files.

"EZpak": a self - booting disk / DDP makes a great medium to store your own files on; contains EZmenu and EZcopy.

"ezFILER": a self - booting disk / DDP containing an address filer utility with advanced print options; graphics screen after reset.

"SHAPEMAKER": a large collection of font shape tables; a very nice hi - res shape design utility; several demonstration programs and instruction files. Written by Guy Cousineau.

PRODUCT ORDER FORM

YOUR NAME _____

ADDRESS _____

CITY _____ STATE _____

ZIP _____ PHONE _____

ID NUMBER _____

PRODUCT	QNTY	MEDIA	PRICE

Subtotal→	.
Shipping→	.
Tax→	.
Other→	.
Subscription→	.
Grand Total→	.

TAX:
- applies only to residents of West Virginia
- compute as 5% of order subtotal

SHIPPING:
- inside 48 contiguous states: $3.00
- elsewhere: $4.50

TO ORDER:
- complete this form; send check or money order (US FUNDS) to:

DIGITAL EXPRESS
P.O. Box 37
Oak Hill, WV 25901

- THANK YOU for shopping with DIGITAL EXPRESS -

August 1987 issue of N&B

P.O. Box 37
Oak Hill, WV 25901

DIGITAL EXPRESS

NIBBLES & BITS

The monthly newsletter for
users and programmers of

ADAM™

THE COLECOVISION
FAMILY COMPUTER SYSTEM