

SYNTAX

2.5

NOV - DEC 1986

TABLE OF CONTENTS

Looking Ahead.....	2
Letters.....	3,4,5,6,7,8,9
Machine Language Primer.....	10,11
Program Reviews: Electronic Game Pack 2.....	12
Jumpman Junior.....	13
RECIPE FILER.....	14
FCAUG BASIC LIB. #2 & #3.....	15
Book Review:.....The Hacker's Guide To ADAM Vol. 1.....	16,17
Adam Graphics.....	17
Programs.....	18,19
FCAUG Product List.....	20



FIRST CANADIAN ADAM USERS' GROUP
P.O. Box 547 Victoria Station
Westmount, P. Q.
H3Z 2Y6

POSTAGE PAID	PORT PAYE
Permit No. 7152 Montreal, Que.	

Looking Ahead...

We are approaching the end of our second year of operation. The Adam is still doing well despite all the initial criticism. In fact, there are many people who ask us where they can still buy one. The fact remains that a lot of people realize that the Adam is a great first computer. For its price, it delivers a reliable word processor, a CP/M compatible system and a great programming machine. In addition, there is now plenty of software to choose from even specialized titles if you have special computing needs.

We at FCAUG have decided that as the Adam continues to evolve as a home computer, a stronger need for better and more accessible software and hardware exists. Therefore, in 1987 we plan to devote more time to software evaluation and development. We have accumulated many great programs (written both by us and others) which we are very anxious to make available to all active and interested users. We will continue with our publishing activities as well. The number of issues may be reduced if we find that we cannot allocate sufficient time to it. The SYNTAX newsletter will remain the central focus of our activities. Look for several new additions to our BASIC and CP/M Libraries. Ted Ewanchyna, a true die-hard Adam supporter, has joined our regular staff of writers.

We still do not have any 80-column cards in stock but this should change soon. We have acquired several game cartridges, modems and Coleco software which have been added to our product list (p.20). Another useful item is the "Flippy Tape". These are specially formatted tapes having two directories allowing you to use both sides. Each side has a maximum of 128 blocks. This means that you can use one tape to store two separate software titles.

SUBSCRIPTION RENEWAL FORM
Fill out and return to

NAME: _____
ADDRESS: _____
CITY: _____ PROVINCE: _____
POSTAL CODE: _____ [] New Member check here

- [] Please renew my subscription to SYNTAX for one more year. My payment of \$20.00 is enclosed.
- [] I will not be renewing my subscription. Keep me on your info mailing list. (\$2 fee is required)
- [] Please send these back issues:
(\$4.00 for single issues, \$3.50 for 6 or more)
- [] I have the following questions or suggestions.

W
E
S
T
M
O
U
N
T
X
Q
U
E
S
T
I
O
N
S
A
R
E
A
V
A
I
L
A
B
L
E

L E T T E R S

Dear FCAUG,

We purchased a program by Strategic Software last Christmas from the U.S. (paying duty, shipping and exchange - ouch!). It is called PaintMASTER. With PaintMASTER you can make complicated drawings easily and save them. However, you cannot use them in your programs, at least it is not shown in the manual. The manual is well written and easily understood. My son enjoys the program. It is written in Basic with some machine language routines so I hope to learn from it. I would rate the program 7 out of 10. It would rate a ten if one could easily access the pictures from his own Basic programs. We borrowed the Super Cobra game by Parker Brothers from a neighbour who has a Colecovision. The game does not work on the Adam. Does anybody know a way around this and are there any other games like this?

John Stensil
Ingersoll, Ont.

PaintMASTER is an interesting program. We hope to write a review on it soon. It is now being sold with PicturePRINT which will allow you to print your pictures on your Adam printer. There are for some unknown reasons a few game cartridges like Defender, The Heist and Super Cobra that do not work properly on the Adam. It might have something to do with the revised Colecovision game board in the Adam.

Dear Sirs,

Being a novice at computers, my biggest problem is with error messages. I have purchased two books on the Adam and along with the SmartBASIC Programming Manual, I cannot find out how to correct the more complex errors such as, "End Of Data, No Buffers Available and No More Room". I am also having difficulty running the following programs and was wondering if anyone else has the same problems: Recipefile from PROGRAMMING ADAM, and the Mailing Labels System from ADAM'S COMPANION.

Christi Norris
Fenelon Falls, Ont.

We have mentioned the most frequent reason for the "No More Room" error message on several occasions. Refer to SYNTAX 2.3 for more details. In most cases, it is because the 35 file limit of the standard one block directory has been reached. This error will also appear when you work with random access text files. If when

SYNTAX is published bimonthly by First Canadian Adam Users' Group (FCAUG), P.O. Box 547, Victoria Station, Westmount, Que., H3Z 2Y6
Subscription Rates: 6 issues for \$22.00 CDN. Second Class Mail
Registration No. 7152. POSTMASTER: Send all address changes and notice of undelivered copies to the above address. Return Postage Guaranteed. Copyright (c) 1986 by FCAUG. All rights reserved.

you open your file, you specify a length parameter which is too large, the file you create might not be able to fit on your media resulting in this error. A "No Buffers Available" error will result again when working with random access text files. If you open too many files you use up all the memory buffers that are available to handle the data. An "End of Data" error occurs when you try to read or write past the end of your file. You cannot read more records than there are. The error informs you that you tried to read or write to a record which does not exist. There are several programs in both books that do not execute properly. SYNTAX 1.3 gave the corrections for the MIXNMATCH program. In our BASIC LIBRARY #1, we have included a working copy of the happy birthday program. Unfortunately, we do not have the corrections for the two programs you mention. Members who have the required changes should let us know so we can pass it along to all other interested users.

Dear FCAUG,

I have been doing some experimenting with sequential and random access files. I wrote a program which uses a random access file. Each time I run the program, it adds one record to my file. After I have entered the 33rd record, the message, "No More Room" appears all the time. Why does this happen? I create and use my random access file on one disk. I would also like to inquire if you have in your CP/M Public Domain software a Basic language which is easy and fast to run under CP/M 2.2.

Real Courcy
Jonquiere, Que.

The reason for the error is due to a bug in the way data files are stored. Each time you write to the file, you create a new file. After a while these files accumulate and clutter the directory on your disk until eventually you get the "No More Room" message. For some reason, SmartBASIC will not write a new text file over the deleted ones. These deleted versions of your file just sit there occupying valuable space on your media. To remedy this problem, you can create multiple block directories on your disks. This will not eliminate the problem but at least it will not occur as often. A better solution is to run your media through a "crunch" routine which will in effect recover all those blocks containing deleted data files. This type of program re-arranges the existing undeleted files to maximize available space by writing over deleted files. We will make one available in one of our next BASIC Libraries. We have many CP/M programs including Basic compilers such as EBASIC which we hope to place in our CP/M Libraries and make available shortly.

Dear Editor,

I feel that one does not need to transfer the tape from an Adam data pack to a suitable audio cassette in order to make copies. (SYNTAX 1.6) If one is careful, one hole can be drilled through the top and bottom case of the Adam data pack allowing it to fit into a regular dubbing machine. If the Adam tape is completely wound in one direction or the other in order that the leader is

exposed, the hole can be drilled without much danger. I carefully brought enough of the leader out of the case to go around an eraser or plastic bottle cap, or the like. The eraser or plastic cap protects the tape while drilling takes place. The size of the hole required is actually marked on the Adam case. Only one hole needs to be drilled as the cassette always goes into the drive with the side up showing the Adam logo. The drilled hole, however must go through both sides of the shell. I find it works fine as I have access to a high speed multiple cassette dubbing machine.

John D. Evans
Thunder Bay, Ont.

Dear Editor,

Some time ago, I received an invitation to join the FCAUG. At the time, I was only using my Adam as a word processing machine and therefore had little need of contact with others. However, with the demise of Coleco's association with Adam, the situation described in your letter of introduction has worsened to crisis level whether one wants to compute or not. During the coming of Halley's Comet last fall and this spring, I de-mothballed some old BASIC programs which allow one to calculate the locations of objects in the sky (Altitude and Azimuth) given their celestial co-ordinates (Right Ascension and Declination). This system was very useful in advising friends and correspondents, all over Canada and the World, of where to look for a glimpse at this elusive object. In spring, I went to Australia to view the comet close-up and was quite satisfied with the accuracy of my predicted sighting angles. While in Australia, I saw Adam computers on sale marketed through a subsidiary of CBS Electronics (USA). I took this as a good sign inasmuch as good software and hardware could become available if this change eventually was to happen in Canada as well. Thank you for the invitation and my apology for ignoring it until now. The comet trip and preparation for it was a year long job and had the priority. Unfortunately, during that time I could not afford to buy the memory expander and disk drive unit while available. That in retrospect seems a gross error.

Earl R. Milton
Lethbridge, Alta.

Thanks for sharing such a fascinating experience with us. As you discovered, Adam computers were indeed sold in Australia as well as Europe by CBS. This is encouraging news because it creates a large and strong network of international users. Through this association, we hope to benefit from the exchange of information and ideas that will develop. We are already in contact with several users in France and the Netherlands. Hopefully, we will have an Australian contact in the near future. Anyone having friends, relatives, correspondents or business contacts in Australia, should ask them to spread the word about Adam users and our Group. It probably would have been a wise decision to pick up a disk drive or two since they are quite scarce. As for memory expanders, you are able to buy them here at home for less.

Dear Editor,

After reading the article on multiple block directories in SYNTAX 2.3, I set out to find an easier way to create a directory of any size I wanted. This is what I discovered. There is a program in EOS that creates the directory called << INIT TAPE DIR >>. This program has a start address of F323H (62231 Dec.). The EOS program requires an input from SmartBASIC of the number of blocks (sectors) that will be directory blocks. SmartBASIC's << INIT >> program gets this value from a fixed memory location and places the value in the "C" register prior to calling the EOS program. To make directories of any size, all we have to do is poke the number of blocks we wish to have as directory blocks before we INIT the tape or disk. I have added the following note to the INIT page in my SmartBASIC manual. "Before using INIT remember to << POKE 25308,n >>, where n = the number of blocks wanted in the directory." Immediately after using the INIT command we can check to see how many blocks are in the created directory by typing: PRINT PEEK (64902). A big THANK YOU goes to Peter and Ben Hinkle for volume 2 of The Hacker's Guide to ADAM which contained the start address for INIT that enabled me to find the rest.

Neil May
Edmonton, Alta.

Dear FCAUG,

I have written a short turnkey system for Adam which requires a passcode in order to access SmartBASIC. The problem, however, is that anyone can get around my passcode by simply pressing Control-C. Can I write a line into the program that will prevent an unauthorized user from breaking into the program?

Michael Walsh
Mount Pearl, Nfld.

Yes, you can do it several ways. The simplest way that comes to mind is to insert an ONERR GOTO statement in the program to override the computer's normal error-handling procedures. When the unauthorized party hits Control-C, or any key other than the one(s) he is instructed to use, whatever instructions you have written into the (jump to) line will be executed. The ONERR GOTO should be placed at the beginning of the program. Here is an example on how to do this.

```
10 ONERR GOTO 9999
9999 TEXT: NEW
```

Line 9999 would normally be placed at the end of your program. Make sure you save the program before trying this out as the NEW command will erase it from memory when you go to press Control-C to break out of the program. Still, another method to disable the Control-C break procedure is to insert POKE 16134, 255 in the program. If you poke this same memory location with a value of 27, the Escape/WP Key will replace the break function normally performed by Control-C. The default value of this memory location is 3. What you must realize, however, is that there are no foolproof methods to keep a determined person from getting into any program.

Dear Sir,

I understand that an expansion board is now available which is CP/M compatible and which displays an 80-column screen. What significance does this have for Adam owners?

W. Wayne Hallett
Milltown, Nfld.

An 80-column card will make using CP/M software more practical because it will eliminate the need to scroll back and forth to see what you are working on. To use the card you will need a good high-resolution monitor as the output will be small on the screen. When using the word processor, you will be able to see complete lines instead of the wrap-around display you get with the standard format. The 80-column card will make the Adam an even better system to use than it already is.

Dear FCAUG,

I wrote to you recently asking your opinion about the editorials in the American Adam publications regarding the Adam Evaluation Club. You had not had a chance to read the articles, so I am reproducing them here.

From the NIAD newsletter:

"Software piracy is abounding in the Adam community!! I was appalled to receive a catalog from a company in Canada that is openly selling COPIES of most of the Coleco software and many titles from other vendors! The vast majority of these products are still available on a retail basis and selling copies is literally stealing the profits from the retail and wholesale outlets! Evidently the rationale for this illegal action is that the Adam is orphaned and the owners are left high and dry with no software available, so people are doing us a favour by illegally copying and distributing this software - BALONEY. The motives are unfortunately quite clear - the almighty dollar. These people are making large profits at the expense of others. I constantly hear of complaints like "why don't we have more new software coming out from other software companies". If this kind of piracy continues NO ONE will put forth the effort to develop software for the Adam because they know their profits will be significantly reduced by this attitude of "...it is okay to sell or give copies of programs away cause we own the orphaned Adam." I am warning each and every Adam owner out there. Do NOT support this attitude by buying this copied software or taking copies of software from friends - if you do it will kill the development of Adam software and hence the Adam. This is a very serious matter that should not be taken lightly by any Adam owner. It is up to you. You can save a few dollars by supporting this illegal attitude, but end up drying up the source of quality Adam software. The only Coleco developed software that is legally in the Public Domain are those titles that were never commercially released. The titles are on our PD catalog listing in this issue. NO other Coleco titles are in the PD and hence can not be copied for sale or given away. NO other Adam titles developed by third party vendors have fallen into the PD. Most of these programs are still avail-

able commercially via mail order. I feel so strongly about this that I will turn over any cases I see of this happening to the proper authorities. Software piracy is against the law, period. Additionally, we are seriously investigating the use of copy protection software for commercially sold Adam software. I ask you for your cooperation in stopping this piracy."

From Expandable Computer News:

"We also received a mailing from a Canadian company that is offering for "evaluation" copies of Coleco cartridges and software on data pack. This appears to be simply a disguised method of selling illegal copies of the software. Because of this I will not name the company. If you received one of these mailings, I urge you not to buy products that are likely to be illegal."

Well, what do you think? My feeling is that the whole situation is somewhat altered by the fact that we live in Canada. Software is hard to get and although we can mail order (as I do all the time), the exchange rates are killing us. AEC offers an alternative and I must admit that for the most part I do use the stuff as evaluation software (i.e. if I like a program, I will buy the original). Also, NIAD's view seems rather simplistic to me. Although I am sure that Pete Cooper from AEC is making a good profit, I really do think that he is committed to helping Adam owners. He helped me get my second disk drive. He has always answered my questions promptly and offered me help to use my Adam to the fullest extent.

Stan Wong
Edmonton, Alta.

We are touching on a delicate subject in the computing world with this letter. But since FCAUG has been responsible for making most of you aware of AEC by carrying their ads in the past, we wish to offer the following comments on the subject.

I agree with you that software is next to impossible to get here in Canada. Although that is a valid point, I think that what the Americans are trying to point out is that software piracy is illegal and therefore shouldn't be done at all. Furthermore and more specifically, piracy is killing third party support of the Adam computer and that by pirating software we are cutting our own throats. These two points are valid ones. However, I do agree with you that NIAD's analysis is a little too simplistic, so let us examine these points a little more closely.

The first point essentially boils down to asking whether the author of a work is entitled to remuneration for his/her work. Most people (and certainly programmers) would unequivocally answer, YES. This, then is what I call the programmer's argument, and it applies to any media - print, audio, or visual. The other argument is computer specific. Simply put, disks and tapes are very easy to duplicate. It is also very difficult to police this activity if done illegally. This point then is not really an argument or a defense, but a recognition of reality. The great problem since microcomputers have arrived then is how to recon-

cile these two points.

NIAD suggests a copy protection scheme for the future, while both ECN and NIAD condemn groups like AEC and then ask others to do the same in the interim. Both wish to stop piracy. Not only do I think these proposals naive but they also miss the point. Let's let experience be our guide here. IBM users have had to deal with software clubs that allow their members the opportunity to evaluate programs before they buy them for years now. And copy protection routines have been around for just as long. But for every copy protection routine developed there has been another routine developed to break it. Any knowledgeable programmer can eventually get into a program and if he's not capable, there are plenty of protection-breaking programs which he can buy legally.

Ward Christiansen, a long time public domain software developer, has this to say in his .doc file for his PD program RESOURCE. "Most good software has to be paid for. I strongly suggest the legitimate purchase of licensed software...(Yes, I do occasionally "try" one, but then buy it if I plan on using it)." Is Mr. Christiansen condemning piracy? Yes, but he's also being realistic, a smart shopper, and extremely responsible. And that is what it comes down to for us. No amount of finger pointing is going to save our computer. People are cutting their own throats when they get software from an evaluation club and use that copy as if they had bought the actual program. The individual who follows this practice is as much to blame as the supplier for in a free society, it is the individual who makes his own decisions. Clubs like AEC may or may not be tasteful to you, but they do provide a service if used responsibly. Since there are more users than evaluation clubs, it is the responsible actions of each individual that will determine the fate of our computer. It is up to you, the user, to decide and to act responsibly when making your next software purchase. [Ron Saunders]

Pirating software will not make it easier to find since it discourages new development. Our objective is to make existing and new software affordable so that acquiring illegal copies becomes a less attractive alternative. We can sympathize with the opinions expressed as we are often victims of both software and SYNTAX pirating ourselves! [J.D. Moore]

Useful SmartBASIC 2.0 Pokes

POKE 17184,0 to 15 (border color)
POKE 17240,0 to 255 (normal text color)
POKE 17251,0 to 255 (inverse background color)

POKE 19361,3: POKE 19365,14: POKE 19366,02: POKE 19367,00: POKE 19368,00 (allows you to shorten the CATALOG command to CAT)

The SmartBASIC 1.0 equivalent pokes for CAT are:

POKE 20183,3:POKE 20187,14:POKE 20188,2:POKE 20189,0:POKE 20190,0

Machine Language Primer

This time we deal with two EOS routines used to read and write blocks to disks or tapes. The directory of every EOS tape or disk is located in block 1. A block is 1K bytes of data. A tape is divided into 256 blocks (blocks 0 to 255) while a disk contains 160 blocks. A standard directory contains only 1 block (see SYNTAX 2.3 p.7) but we will be able to expand it to 2 with our program which uses the routines to modify the directory entries.

At memory location FCF3 (hex) EOS has a routine which will read a block from a media into a memory buffer. Before calling we must do the following: load reg. A with the device number, load reg. HL with the starting address of the 1K buffer, and place the block number we want to read in the registers BC and DE. After calling the block read routine, we will have the contents of that block in the buffer in memory if there was no error. If there was an error then reg. A will return a non-zero error code. Normally reg. A will return zero. The device numbers are 4 for disk 1, 5 for disk 2, 8 for tape 1, and 18 for tape 2. The block number as you can see takes 4 bytes. This allows for an incredible four billion blocks. For us, we always set the BC register pair to zero and the DE reg. pair to the block number.

Memory location FCF6 (hex) contains the complementary write-block-to-media routine. The registers are set up in the same way as in the read routine except the 1K of data in the buffer is written to the appropriate device. The assembly code one can use to read and write a block looks like this:

```
LD BC,0      'zero top of BCDE block #
LD DE,1      'read block # 1
LD A,4       'read from device=disk 1
LD HL,6BD0   '1K buffer at 27600 (dec)
CALL FCF3    'read a block
RET          'return
```

The write routine is identical to the one given above except we CALL FCF6 instead. The machine code in our program makes use of the fact that the addresses only differ by the least significant byte and by poking the appropriate value we can use one routine to both read and write a block.

Getting back to the program provided, here is some background on how the directory is set up. Each directory block consists of 39 - 26 byte entries. The first entry is called the volume record. Its 26 bytes look as follows:

```
12 bytes - volume name (terminated by 03 if < 12)
1 byte   - directory size (in blocks) can have a max.
           of 127 blocks. The most sig. bit is for
           delete protection (1 = on, 0 = off)
4 bytes  - a code which tells EOS that this is a directory
4 bytes  - total # of blocks on volume
2 bytes  - reserved
3 bytes  - creation date (yy/mm/dd)
```

Following the volume record are the file entries which look as follows:

- 12 bytes - file name (terminated by 03)
- 1 byte - file attributes
- 4 bytes - starting block number of file
- 2 bytes - physical file size (in blocks)
- 2 bytes - file size used (in blocks). The file might be in a file space > actual file size.
- 3 bytes - creation date

The file attributes byte controls how the system looks at a file. Each bit in the byte corresponds as follows (from most significant to least significant bit): permanently protected, write protected, read protected, user file, system file, deleted file, execute protected, not a file. When the corresponding bit is one then the above condition will hold (ie. if bit 0 is 1 then the entry is not a file). The program provided will allow you to go through a directory and toggle these bits by pressing the number key to their label's left. By pressing return you move to the next file entry. Pressing the Escape key at any time will save your changes and end the program which is similar to reaching the end of the directory. Modifying the directory entries is only the 1st option of the program. The 2nd option allows you to create a 2 block directory. You should only choose this option when using a fresh tape/disk or one which has been INITIALIZED first.

```
100 & directory modifier
110 LOMEM :29000
120 TEXT
130 FOR i = 0 TO 14: READ d: POKE 27500+i, d: NEXT
140 DATA 1,0,0,17,0,0,62,0,33,208,107,205,243,252,201: & block read/write code
150 PRINT "4=disk1 5=disk2 8=DDP1 18=DDP2"
160 INPUT "enter a device number: "; dn$: IF dn$ <> 4 AND dn$ <> 5 AND dn$ <> 8
AND dn$ <> 18 THEN 160
170 PRINT "Choose:": PRINT " 1) modify directory entries": PRINT " 2) double
directory size"
180 GET a$: IF a$ = "2" THEN 420
190 IF a$ <> "1" THEN 180
200 & modify directory entries
210 bl$ = 1: GOSUB 460: b$(0) = "NO ": b$(1) = "YES"
220 HOME: VTAB 4: PRINT "0) not a file": PRINT "1) execute protected": PRINT "
2) deleted file": PRINT "3) system file"
230 PRINT "4) user file": PRINT "5) read protected": PRINT "6) write protected"
: PRINT "7) permanently protected"
240 entry = 27626: dsize = PEEK(27612): IF dsize >= 128 THEN dsize = dsize-128
250 i = 0: HTAB 1: VTAB 2: IF nm$ = "BLOCKS LEFT" THEN 400
260 nm$ = ""
270 x = PEEK(entry+i): IF x = 3 THEN 290
280 PRINT CHR$(x): : nm$ = nm$+CHR$(x): i = i+1: GOTO 270
290 PRINT: PRINT
300 x = PEEK(entry+12): FOR i = 7 TO 0 STEP -1: b(i) = 0: IF x >= (2^i) THEN
x = x-(2^i): b(i) = 1
310 HTAB 26: VTAB 4+i: PRINT b$(b(i)): NEXT
320 GET a$: IF a$ = CHR$(13) THEN 370
330 IF a$ = CHR$(27) THEN 400
340 IF a$ < "0" OR a$ > "7" THEN 320
350 HTAB 26: i = VAL(a$): VTAB 4+i: b(i) = 1-b(i): PRINT b$(b(i))
360 POKE entry+12, 128*b(7)+64*b(6)+32*b(5)+16*b(4)+8*b(3)+4*b(2)+2*b(1)+b(0):
GOTO 320
370 entry = entry+26: IF entry >= 28614 THEN entry = 27600: GOSUB 480
380 IF entry = 27600 THEN bl$ = bl$+1: GOSUB 460: IF bl$ > dsize THEN END
390 GOTO 250
400 GOSUB 480: END
410 & double directory
420 bl$ = 1: GOSUB 460: POKE 27612, 2: POKE 27617, 0: POKE 27618, 1: POKE
27669, 2: POKE 27671, 2
430 POKE 27691, 3: POKE 27695, 252: IF dn$ < 8 THEN POKE 27695, 157
440 GOSUB 480: PRINT "done": END
450 & read a block
460 POKE 27512, 243: POKE 27507, dn$: POKE 27504, bl$: CALL 27500: RETURN
470 & write a block
480 POKE 27512, 246: POKE 27507, dn$: POKE 27504, bl$: CALL 27500: RETURN
```

Program Review: Electronic Game Pack 2 - by APE Software

This package is APE's follow-up to their excellent EGP 1 released a year ago. Four action games are included in Electronic Game Pack 2. They are: Car Wars, Crater Tag, Sky King, and Snake Weeds. These are arcade-type games.

On initial contact with this new package, one might think that these are simple games to play. Well, I soon found out that the opposite is true in most cases. The first game, appropriately titled Car Wars, pits you against a computer car. You must race around a square track to collect all the dots in the different lanes. You can drive at 3 speeds. The computer controls the black car which will try to meet your car head on. To keep it out of your way, you must control your speed and switch lanes often. As the rounds progress both cars move faster with the computer car getting more aggressive. This program provides a good game of speed and chase.

Crater Tag is a game where the player must be quick on the trigger. The object is to move the green astro-miner to gather all the blue gems while avoiding the fireball creatures that inhabit this cratered planet. You make your miner scurry about by pointing the joystick in the direction you want to go and tapping either fire button. By tapping on both fire buttons simultaneously, you can really make him move his tail. Once all the gems have been collected, a purple spacecraft appears which you are to embark to reach the next crater. This sounds like an innocent little challenge but it is not. The pace is hectic and your hand will get tired quickly.

The concept behind the next game, Sky King, is intriguing. You control a new military aircraft called a Helipod in a simulated test of combat conditions. Your mission is to fly over enemy anti-aircraft bases and destroy their firing positions located on the pyramid structures. They fire deadly electro-loops which will destroy your Helipod on contact. Using the joystick and both fire buttons, you must drop your bomb and detonate it when it is on target. When the electro-loops get out of hand, you can use your limited number of shields to avoid destruction. Destroying all the enemy firing positions moves you to new bases.

Snake Weeds is like the name implies about snake-like weeds that spring up all over the place. You must stop this plant life from spreading by shooting at the killer spores they release using your cannon gun. Snake weed pairs release spores. You must avoid shooting a spore by a single snake weed as it will create a new generating pair. If there is a snake weed pair beside the spore or it is between single snake weeds, then those snake weeds will be destroyed except a spore will be left behind. It takes careful planning and quick attacks to eradicate these deadly snake weeds.

APE Software has done a good job with all four games. They offer interesting scenarios, an excellent element of challenge and good visual presentation. They are all simple to understand and fun to

play. These are the type of games where you have to sit down and plan your attack carefully otherwise you will not experience the genuine challenge they offer. The only minor criticism I have is with Crater Tag. I found the astro-miner too small and a bit hard to maneuver.

Program Review: Jumpman Junior by EPYX

It has been some time since we reviewed a game cartridge so we thought we would take a look at a non-Coleco title for a change. Epyx is well-known for its excellent computer games for other systems like Apple, Atari, etc. Fortunately for us, they also released a few of their most popular titles including Pitstop, Gateway To Apshai and Jumpman Junior on ROM cartridge for the Colecovision back in the days when it reigned supreme.

Jumpman Junior has 12 game levels and from one to four players may play. Different difficulty levels are possible ranging from 1 (fastest) to 8 (slowest). The first 12 levels are played in sequence. If the player is successful in completing all 12 levels, the game will then select random levels. Only the number 1 hand controller is used regardless of how many people are playing.

The objective of the game is to defuse (by touching) all the bombs found in each level. Each screen represents a different level of a space Substation which is under invasion by the enemy ALIENATORS. Jumpman must climb up and down ladders, ropes and elevators to reach the bombs. There are many bombs to defuse and many dangers making it difficult for Jumpman to reach the bombs in time. To get out of tight situations, you can make Jumpman leap out by pressing the fire button and the joystick at the same time. This gives him added jumping power.

As you play the game and progressively move on to a faster speed level, you will start to appreciate the true complexity of the challenge. This is an action game at its best. You must maneuver your Jumpman over fires, avoid speeding bullets, jump gaps, overcome moving walls, etc., in a deadly race against time and destruction.

This game has a few interesting aspects not found with most other game cartridges. One is that you can enter your initials (maximum 3) in the high score screen. The variation in speed is the highlight of the game. At top speed, you move about the structures as fast as a speeding bullet yet you must be most cautious because there are more traps and perils awaiting your slightest error. At the other extreme (slowest speed), your Jumpman moves virtually in slow motion.

With its fast pace and 12 levels, Jumpman Junior is full of challenge and variety. It also has its humorous moments and some cynical screen titles making it a tempting and entertaining game to tackle.

Program Review: RECIPE FILER - Coleco Industries Inc. (1984)

Recipe Filer is one of three programs in the Coleco Home Helpers Software series. If you are familiar with SmartFILER or Address Book FILER, you will quickly see the similarity. While SmartFILER is a general purpose database program, Recipe Filer is intended for one specific purpose - to file recipes.

Recipe Filer has all the standard features of a filing program (i.e. add, delete, search, index, print) plus a couple of unique ones; "Shopping List" and "See Links". It does not allow you to create your own form. All the fields and their length are already pre-selected for you. Although this sacrifices some flexibility, it does make the program very user-friendly when entering your recipes or using any of the 30 already included.

When adding recipes, a blank recipe card appears on the screen containing the name of each field for which you can enter data. The fields are: Recipe Name, Recipe Type, Time, Main Ingrid, Link To, Quantity...Ingredients, Directions, and Comments. The first two and "Main Ingrid" are the searchable fields. The last three fields are able to hold a lot of information as they make up the major part of every recipe. It is not recommended that you store your recipes on the program media. Use a blank data pack or disk instead to hold your database.

Like in SmartFILER, you can retrieve records or recipes using the search or index method. Index will display a list of all recipes on your database in alphabetical order by one of the 3 searchable fields you specify. With the search method you can find recipes by name, by type, by main ingredient, or by any combination of these. Search with partial information is also possible. When you display a recipe, the two new features can be used. Pressing "See Links", gives you a list of all other recipes linked to the one you are viewing. The "Shopping List" feature allows you to enter items directly on a list or take ingredients and their quantities from a specific recipe and automatically add them to your list. You can print out this list and take it with you to the store.

The print option works just like with SmartFILER except that you are not able to create your own formats. You can print a list of recipes, selected recipe names on the list, individual recipes, or your workspace.

This program is highly recommended to those who enjoy cooking. Although you can use SmartFILER to do the same job, Recipe Filer has its advantages. It is already set up to accept data plus it has nice extras like colorful graphics and two unique features. The 30 recipes included are also an added bonus.

The program's major weakness is that it is slow. It is either reading or writing to the media most of the time. One final note, Recipe Filer shows revision 14 5/23/84 on data pack and revision 16 8/15/84 on disk. It remains to be seen, however, if the latter contains corrections or only cosmetic changes to the program.

Program Review: FCAUG BASIC LIB. #2 & FCAUG BASIC LIB. #3

This second package like the first one consists of 30 programs. We have completely re-done the format for all three program Libraries. (If you ordered Library #1 and would like the revised edition just send us your original tape plus \$2 for return postage and handling.) Now the media containing the programs is self-booting. You simply insert it into the drive and pull the reset switch. Once SmartBASIC is loaded, the title screen will appear followed by the main menu screen. The names of all the programs will be listed. To RUN any of the programs, you simply enter its corresponding number and hit the RETURN key.

The programs in BASIC Program LIB. #2 fall into four categories.

1. SOUND: MUSICNETH - plays different sounds similar to those from arcade games. MUSICSUBR - a music subroutine which you can use in other programs. ZOISEAU - draws a chirping bird on the screen. This programs works with OISEAU.
2. GAMES: SINGLEDIE - roll the die and call the number. SLOTS - simulates a slot machine in a casino. STARPATROL - test your skill at shooting down the starship also computes your hit and miss ratio. MULTIMATH - a very complete educational math quiz. MOONBASE - (from SYNTAX 1.5 & 1.6) pilot your spacecraft to successful landings on different moon bases before running out of fuel or crashing.
3. GRAPHICS: ENTERPRISE - draws the spaceship from the Star Trek series. FORTINFUNC, SURFPLANE, MOUNTEXP, COSURF, GLOBE, 3D RIPPLE, 3D DETAIL - some intriguing graphics using functions. BLOCKNUMBR, BELL, BALL, GR FANTASY, FLAGS - examples of interesting applications in the GR mode. MULTIDRAW, LOVE, BEAU-NOME, JET, STARWARS, PYR - superb high resolution graphics meticulously done.
4. PADDLE INPUT: SYNTAX1.5 and HIRES PDL - two different programs to draw your own graphics in the HGR mode using the joystick.

FCAUG BASIC LIB. #3 is primarily an entertainment package. In this collection, you will find a variety of games and graphics. The games included are: HANGMAN, MISSILE, MATH, BARRIER, STREET, QUATRAIN, CAMEL, CRAZY MAZE, TOWERS, SNAKERIDER, TICTACTOE, and POKER. The POKER program written by FCAUG member Andre Auger is excellent and worth the price of this Library alone.

The graphics programs are: PORTRAIT - draw faces on the screen. VALENTINE - address a personalized valentine to your true love. MERRYX - use your printer to draw holiday wrapping paper. KALEID, POLYGON, DRAW, SPIRAL, and DIGITAL - unique graphics to entertain you. With LIB. 3, you can return to the main menu by running the "hello" file. With LIB. 1 & 2, you can only go back by rebooting. Use the catalog command to see the contents of each Library.

Although we have tested every program in each Library, sometimes in the duplication process there are blocks which are not properly written especially when using tapes. This will result in some programs not working. If you come across this problem, please let us know which program is involved and feel free to return the tape to us for correction or replacement.

Book Review: The Hacker's Guide To ADAM Volume 1

Books like this do not come along very often especially for the Adam computer. Fortunately for us, "The Hacker's Guide to ADAM" did. Written by a father and son team supposedly for personal pleasure, this over 60 pages manual is the most understandable and comprehensive report on Adam to date. Since it was released over a year and a half ago in the U.S., we feel it is certainly about time that we told you more about it.

First, this manual is not for everyone. If you are a user and are happy with running finished programs with nary a care of how they work, or are part of the exclusive word processor user set, you will get very little out of this book. On the other hand, if you want to learn how to program and are frustrated by the lack of information available to do so, this text is for you. (Sounds like the typical Adam user!) The Hinkles seem to have started their project by just wanting to learn more about their Adam. Compiling their findings, coupled with the occasional technical paper that they could find (Coleco didn't help, we should add), eventually led to what is the subject of this review.

I should add that I have most of the technical papers including both versions of the Coleco technical manual, and a host of other independent findings on the Adam. Do I use the Hinkles' Hacker's Guide? You bet I do. Even though I have this other information, the Hackers' Guide puts a lot of the things that I need to know right at my fingertips. In fact, it's the perfect mix. There is enough well-written text for the beginner to gain insight on the inner workings of the Adam. But for people like me that know this stuff already, the tables of information that I need are easily found. So, as you can see, individuals who want to learn more about their Adam can get help no matter what level they are at.

The Guide starts off with the basics. First, a block diagram of the system shows all the components in the Adam system. Some of you will be surprised to learn just how complicated a computer Adam actually is. Jumping to the back of the Guide, we find the pinouts of the various chips used in the Adam including identification of the different lines of all 4 expansion connectors and the cartridge connector. Returning to the beginning of the Guide, we find assembly language basics - number representation and identification of Z80 architecture and opcodes. I like the chart on p. 3 for converting between the hexadecimal and decimal number systems and the listing of the disassembler program later on.

Quite a bit of space is allocated to the explanation of the video and audio chips. As with every section, the Guide is liberally sprinkled with programs that illuminate the ideas discussed throughout the text. The rest of the Guide consists of about one page explanations of the other Adam components: the game controllers, the Adamnet serial bus, the keyboard, the printer, the tape, cartridges, disks, power supply, memory bank switching, the operating system (EOS), and the SmartBASIC interpreter.

The Guide covers a lot of ground. Because of this, one may be lead to believe that the authors are covering too much in too little space. To a certain extent that is true. However, the emphasis in the Guide is on programming. The material discussed is geared towards the programmer or "hacker". This means that the programmer will have enough technical information to program the Adam like never before. The examples are all highlighted in simple Basic and some of the programs are great. The Hinkles provide listings of a tape copy program, a tape editing program, and a cartridge backup (to tape or disk) program to name a few.

The Hackers' Guide to Adam is a detailed look at the Adam written with the programmer specifically in mind. Although, we have to some extent touched upon much of the same material in SYNTAX before, it is the convenience of finding it all in one reference manual that is so attractive about this publication. As a result, we feel that even most novices can benefit from it simply by reading it. The programs included are very useful and because they are well documented, they are excellent learning tools.

FCAUG is making the Hacker's Guide to Adam available in Canada. We have made arrangements with the Hinkles to sell their Guide plus a tape or disk that includes all the programs ready-to-run from the manual. We also have the Hacker's Guide volume 2 which looks at the SmartBASIC interpreter exclusively, and will be reviewed next issue. See the back page for ordering details.

A D A M G R A P H I C S

The two programs that follow were taken from Home Computer Magazine Vol. 5 No. 6. They both use the Hi Res screens and the H PLOT (line drawing) command.

The first program creates a ripple effect as if a rock was thrown into the water. It is created by plotting a series of sinusoidal waves with changing amplitudes and periods. The waves' amplitudes decrease as they move away from the center of the screen. The program also makes sure not to plot over any foreground waves by checking the array MAP (ie. screen map).

The second program is a bit longer than the first program but its results are more spectacular. It creates a very life-like tree. It works by using a technique called recursion. Recursion is a method for solving a problem using a routine which calls itself until some condition is met. In this case, the routine draws branches and continues to draw new branches until the branch length becomes 1. Every branch plotted represents a level of recursion. At each level the branch size is decreased and a new branch end point is created. By stacking this information, the next level of recursion can use this information and decide to continue drawing or to end. If it continues it draws a branch and places the updated branch information on the stack. If it ends then it drops the last information from the stack and draws a branch 90 degrees from the old one. It will then continue from there. The effect will become evident when you run this program. Try changing part of line 110 to RE(1,3) = 13 or 14 to get a bushier tree or to 10 or 11 to get a sparser tree.

```

10 & ripples in water
20 DIM map(255)
30 HGR2: HCOLOR = 6
40 FOR x = 0 TO 255: map(x) = 199: NEXT
50 FOR y = 179 TO 11 STEP -6: FOR x = 0 TO 255
55 d = .2*SQR((127-x)^2+(95-y)^2): f = COS(d)/(.08*(d+(d = 0)))
60 p = y-12*f: IF p > 191 THEN p = 191
70 IF p < 0 THEN p = 0
80 IF p < map(x) THEN map(x) = p: GOTO 100
90 p = map(x)
100 IF x = 0 THEN H PLOT x, p
110 H PLOT TO x, p
120 NEXT: NEXT: END

```

```

10 & tree drawing program
20 DIM re(100, 4), dir(8, 2)
30 TEXT: INPUT "Enter a tree number (1,2,3 ...): "; n: IF n < 1 THEN 275
40 n = RND(-n)
50 FOR z = 1 TO 8: READ dir(z, 1), dir(z, 2): NEXT
60 DATA 0, -1, 1, -1, 1, 0, 1, 1, 0, 1, -1, 1, -1, 0, -1, -1
70 & plot grass, and tree trunk
80 HGR2: HCOLOR = 1
90 FOR i = 170 TO 191: H PLOT 0, i TO 255, i: NEXT
100 HCOLOR = 8: FOR i = 105 TO 109 STEP 2: H PLOT i, 110 TO i, 175: NEXT
110 r = 2: d = 1: re(1, 1) = 107: re(1, 2) = 110: re(1, 3) = 12: re(1, 4) = 1
120 & branch drawing loop
130 IF r = 0 THEN END
140 IF re(r, 4) <> 0 THEN 170
150 d = d+1: GOSUB 220: GOSUB 270: GOSUB 330
155 IF re(r, 3) > .1 THEN r = r+1: GOTO 130
160 d = d-1: GOSUB 220: GOTO 130
170 IF re(r, 4) <> 1 THEN 200
180 d = d-1: GOSUB 220: GOSUB 270: GOSUB 330
185 IF re(r, 3) > .1 THEN r = r+1: GOTO 130
190 re(r, 4) = 0: r = r-1: GOSUB 220: GOTO 130
200 re(r, 4) = 0: r = r-1: d = d+1: GOSUB 220: GOTO 130
210 & adjust direction
220 IF d = 0 THEN d = 8: RETURN
230 IF d = -1 THEN d = 7: RETURN
240 IF d = 9 THEN d = 1: RETURN
250 RETURN
260 & push current branch info on stack
270 re(r, 1) = re(r-1, 1)+(re(r-1, 3)*dir(d, 1))
275 re(r, 2) = re(r-1, 2)+(re(r-1, 3)*dir(d, 2)): rn = INT(RND(1)*15)
280 IF rn < 1 THEN re(r, 3) = re(r-1, 3)*.5: RETURN
290 IF rn < 2 THEN re(r, 3) = re(r-1, 3)*.7: RETURN
300 IF rn < 4 THEN re(r, 3) = re(r-1, 3)*.8: RETURN
310 re(r, 3) = re(r-1, 3)-1: RETURN
320 & draw coloured branches
330 re(r, 4) = re(r, 4)+1: IF re(r, 3) > 4 THEN GOSUB 370
340 IF re(r, 3) <= 4 THEN GOSUB 460
350 RETURN
360 & draw yellow branches
370 GOSUB 500: ON d GOTO 380, 390, 400, 410, 380, 390, 400, 410
380 lx = -1: ly = 0: rx = 1: ry = 0: GOTO 420
390 lx = -1: ly = -1: rx = 1: ry = 1: GOTO 420
400 lx = 0: ly = -1: rx = 0: ry = 1: GOTO 420
410 lx = 1: ly = -1: rx = -1: ry = 1
420 IF re(r, 3) > 9 THEN HCOLOR = 8: H PLOT x1+lx, y1+ly TO x2+lx, y2+ly
430 IF re(r, 3) > 7 THEN HCOLOR = 8: H PLOT x1+rx, y1+ry TO x2+rx, y2+ry
440 HCOLOR = 8: H PLOT x1, y1 TO x2, y2: RETURN
450 & draw green or red branches
460 IF re(r, 3) < 1 THEN HCOLOR = 2
470 IF re(r, 3) >= 1 THEN HCOLOR = 1
480 GOSUB 500: H PLOT x1, y1 TO x2, y2: RETURN
490 & get branch coordinates from stack
500 x1 = re(r-1, 1): y1 = re(r-1, 2): x2 = re(r, 1): y2 = re(r, 2): RETURN

```

```

10 & 3-D Boxes - J.M. Roy
12 ONERR GOTO 52
14 TEXT
16 t = 0
18 sx = RND(1)*256: sy = RND(1)*192
20 dx = (2-4*(sx > 128))*(RND(1)*2+1)
22 dy = (2-4*(sy > 86))*(RND(1)*2+1)
24 lx = (RND(0)*10+5)*dx
26 ly = (RND(0)*10+5)*dy
28 HGR2: t = t+1
30 IF t = 4 THEN t = 5
32 IF t > 15 THEN t = 1
34 HCOLOR = t
36 HPLOT sx, sy TO sx+lx, sy
38 HPLOT TO sx+lx, sy+ly
40 HPLOT TO sx, sy+ly
42 HPLOT TO sx, sy
44 lx = lx+dx: ly = ly+dy
46 sx = sx+dx: sy = sy+dy
48 IF sx+lx < 0 OR sx+lx > 255 OR
   sy+ly < 0 OR sy+ly > 191 THEN
   FOR y = 1 TO 500: NEXT: GOTO 16
50 GOTO 36
52 TEXT: END

```

```

10 & Circle Drawing Demo
12 & for new Demo make
   line 44, HPLOT (x+cx),
   (y+cy) to x, y
14 HOME
16 HGR
18 x = 140
20 y = 90
22 FOR j = 1 TO 5
24 READ r, hc
26 HCOLOR = hc
28 GOSUB 36
30 NEXT j
32 & radii and colors
34 DATA 80,1,70,2,60,3,
   50,5,40,6
36 HPLOT x+r, y
38 FOR ct= 0 TO 6.28318531
   STEP .0174532925
40 cx = r*COS(ct)
42 cy = 6/7*r*SIN(ct)
44 HPLOT TO (x+cx), (y-cy)
46 NEXT ct
48 RETURN

```

```

-----
10 & Textcolor5
12 & Mark Vyse (1986)
14 POKE 17202, 1: POKE 17199, 30: POKE 1146, 93: TEXT
16 DATA black,1,green,2,lt.green,3,blue,4,lt.blue,5
18 DATA red,6,aqua,7,lt.red,8,pink,9,orange,10
20 DATA yellow,11,purple,13,gray,14,white,15,done,0
22 GOSUB 70
24 VTAB (2): INPUT "Print BACKGROUND COLOR: "; d$
26 READ f$, b: IF f$ = "done" THEN e = 1: GOTO 22
28 IF f$ <> d$ THEN 26
30 IF f$ = d$ THEN GOSUB 78
32 GOSUB 70
34 VTAB (6): INPUT "Print TEXT COLOR: "; g$
36 READ h$, a: IF h$ = "done" THEN e = 2: GOTO 32
38 IF h$ <> g$ THEN 36
40 IF h$ = g$ THEN GOSUB 78
42 IF g$ = d$ THEN RESTORE: GOTO 10
44 VTAB (10): INPUT "BORDER COLOR (y/n)?" : t$
46 IF t$ = "y" THEN GOTO 52
48 IF t$ <> "y" THEN q = b: GOTO 58
50 GOSUB 70
52 VTAB (15): INPUT "Print BORDER COLOR: "; s$
54 READ w$, q: IF w$ = "done" THEN e = 3: GOTO 50
56 IF w$ <> s$ THEN 54
58 POKE 17115, (a*16^1)+(b*16^0): POKE 17059, q: TEXT
60 VTAB (2): INPUT "OTHER COLORS (y/n)?" : z$
62 IF z$ = "y" THEN TEXT: RESTORE: GOTO 24
64 IF z$ <> "y" THEN TEXT: END
70 & error routine
72 IF e = 1 THEN VTAB (4): PRINT "reenter"
74 IF e = 2 THEN VTAB (8): PRINT "reenter"
76 IF e = 3 THEN VTAB (17): PRINT "reenter"
78 RESTORE
80 RETURN

```

----- F C A U G P R O D U C T L I S T -----

E-Z EDIT 1.0.....	\$15.00	[]
E-Z COPY 1.0.....	\$15.00	[]
FCAUG BASIC LIBRARY #1.....	\$10.00	[]
FCAUG BASIC LIBRARY #2.....	\$10.00	[]
FCAUG BASIC LIBRARY #3.....	\$10.00	[]
FCAUG BASIC LIBRARY #4 (avail. April).....	\$10.00	[]
CP/M 2.2 COMMUNICATIONS (MEX, MODEM 7, ADAMBOOT).....	\$10.00	[]
JEOPARDY.....	\$10.00	[]
SUPER SUBROC.....	\$10.00	[]
TROLL'S TALE.....	\$10.00	[]
HARD HAT MACK/PINBALL CONSTRUCTION.....	\$10.00	[]
SMARTBASIC 2.0.....	\$10.00	[]
ADAMLINK 2	\$10.00	[]
DATA DRIVE SPEED TESTER.....	\$10.00	[]

THIRD PARTY SOFTWARE

ELECTRONIC GAME PACK 1 - by APE Software.....	\$20.00	[]
ELECTRONIC GAME PACK 2 - by APE Software.....	\$20.00	[]
HACKER'S GUIDE TO ADAM VOL. 1 w/tape or disk.....	\$20.00	[]
HACKER'S GUIDE TO ADAM VOL. 2 w/tape or disk.....	\$20.00	[]
TIMEX - Time Display Program by C. Hills.....	\$10.00	[]
BSR - Block Transmitting Modem Program.....	\$20.00	[]
TRS-2-ADAM & TRS-DIR (convert TRS CP/M files).....	\$20.00	[]

ADAM GAME CARTRIDGES

LADYBUG, VENTURE, PEPPER II (Limited Quantity) each...	\$25.00	[]
B.C'S QUEST FOR TIRES.....	\$20.00	[]
JUMPMAN JUNIOR.....	\$15.00	[]
OMEGA RACE.....	\$15.00	[]
CABBAGE PATCH KIDS Adventures in The Park.....	\$15.00	[]
FRONT LINE (requires Super Action Controller).....	\$15.00	[]
DESTRUCTOR (requires Driving Module).....	\$15.00	[]

ADAM SOFTWARE (original manuals & media)

SMARTLOGO.....	\$30.00	[]
SMARTFILER (Rev. 27D).....	\$30.00	[]
EXPERTYPE.....	\$35.00	[]
ADAMCALC.....	\$55.00	[]
32 BASIC PROGRAMS (Book & tape).....	\$30.00	[]
32 BASIC PROGRAMS (Book only).....	\$20.00	[]
CP/M 2.2 & ASSEMBLER (on disk).....	\$60.00	[]

ADAM HARDWARE AND ACCESSORIES

64K MEMORY EXPANDER.....	\$75.00	[]
CENTRONICS (PARALLEL) PRINTER INTERFACE.....	\$85.00	[]
SMARTBASIC 1.0 on ROM CARTRIDGE.....	\$70.00	[]
ADAMLINK MODEM w/ADAMLINK 2 program.....	\$125.00	[]
SMARTBASIC 1.0 REPLACEMENT DATA PACKS.....	\$12.00	[]
FLIPPY TAPES (both sides are used).....	\$10.00	[]
PICA 10 DAISY WHEEL.....	\$12.00	[]
For Hardware items, please add \$3 for postage and handling.		[]

FCAUG Box 547, Victoria Station, Westmount, Que. H3Z 2Y6