P8

the AARDVARK JOURNAL

june 1980 vol.1, no. 2

)*** WHAT WE GONNA DO IN THIS ISSUE ***

SEVERAL ARTICLES PLANNED FOR THIS MONTH HAVE BEEN DELAYED - INCLUDING THE CHECKBOOK PROGRAM AND A TUTORIAL ON SPEED IN BASIC- SO THAT WE COULD DEVOTE AT LEAST PART OF A JOURNAL TO SMALL DISK USERS.

I DECIDED TO DO THAT BECAUSE I GOT A LOT OF QUESTIONS THIS MONTH ON HOW TO USE DISKS - AND GOT TIRED OF ANSWERING THEM ONE AT A TIME. THE ARTICLES ON DISK USAGE ARE NOT, REPEAT NOT, DESIGNED TO BE A COMPLETE TUTORIAL. WE HAVE ONE IN THE WWORKS, BUT IT IS MUCH TO LONG TO INCLUDE IN THE JOURNAL. WHAT THIS MONTHS ISSUE DOES IS TO ANSWER THE QUESTIONS THAT I GET THE MOST.

WE HAVEN'T FORGOTTEN THE REST OF YOU. THERE ARE ARTICLES ON HOW TO GET CLOSER TO TRUE RANDOM NUMBERS, A REPORT ON OSI AT THE FAIRRE, A BEGINNERS CORNER, AND SOME PROGRAMS FOR THE C1P.

AS A NICE BONUS, THERE IS A SIMPLE WORD PROCESSING PROGRAM THAT WILL WORK ON DISK OR WITH TAPE. (O.K. IT ISN'T REALLY A WORK PROCESSOR, JUST A LETTER WRITER WITH DELUSIONS OF GRANDEUR)

)*** I MET OSI ***

HOW'S THAT FOR AN ODDITY? IT COST ME A GRAND, BUT IT MAY HAVE BEEN WORTH IT. I SPENT A WEEK GOING TO THE NCC CONFERENCE IN L.A. OSI WAS THERE MAINLY TO SHOW OFF SOME NEW STUFF.

THERE WASN'T MUCH NEW FOR THE HOBBYIST, AS THEY WERE MAINLY INTERESTED IN SHOWING OFF THE NEW INTEGRATED BUSINESS PACKAGE RUNNING UNDER 65U, BUT IT WAS MUCH MORE IMPRESSIVE THAN THE PREVIOUS SOFTWARE.

THERE WERE NO NEW MACHINES OR HARDWARE, BUT THERE WAS ANOTHER PEICE OF VERY INTERESTING SOFTWARE. I SAW A C4PMF RUNNING PASCAL AND FORTRAN. THE PROGRAMMERS I TALKED TO SAID THAT IT REQUIRES 48K AND DUAL DISK DRIVES, BUT NO HARDWARE MODS TO C4 OR C8 MACHINES. THE PACKAGE OF BOTH PROGRAMS IS SUPPOSED TO SELL FOR $450

THE HAPPIEST THING I SAW WAS EVIDENCE OF A NEW AND BETTER ATTITUDE FROM OSI. THE PEOPLE THERE WERE VERY INTERESTED IN IMPROVING THE DOCUMENTATION AND FOR THE FIRST TIME WERE SOLICITIOUS OF OUTSIDE HELP FOR SOFTWARE AND INFORMATION. THEY EVEN PLAN TO PUT OUT A LIST OF ALL NON-OSI SUPPLIERS OF OSI SOFTWARE AND SEND IT TO ALL OF THEIR DEALERS. I HAVE SEEN TWO BITS OF EVIDENCE THAT THE NEW ATTITUDE IS AT LEAST TEMPORARILY REAL. I GOT A PHONE CALL FROM OSI (THAT SURPRISED ME - I USUALLY CAN'T GET THEM TO EVEN ANSWER A PHONE) AND THEY ASKED FOR A COPY OF THE AARDVARK CATALOG. I ALSO SAW AN OSI JOURNAL PUBLISHED IN MICRO THIS MONTH THAT SEEMS TO SHOW A WHOLE NEW ATTITUDE. THE INFORMATION IN THE JOURNAL WAS NOT THE USUAL-HYPE A HIGH PRICED SYSTEM- STUFF, BUT WAS ACTUAL INFORMATION ON THE OPERATING SYSTEM OF SMALL OSI COMPUTERS.

MAYBE A NEW DAY IS DAWNING.

ONE BIT OF BAD NEWS. THE STRING BUG IN THE ROM BASIC WILL BE HERE FOR A WHILE, ACCORDING TO SOME OF THE OSI PEOPLE AT THE SHOW. THEY SAID THAT PART OF THE REASON THAT OSI WAS ABLE TO GET THE PRICE OF THE COMPUTERS SO PHENOMENALLY LOW WAS THAT THEY CONTRCTED FOR A VERY LARGE NUMBER OF ROMS FROM THE SAME MASKS. THEY SAID THAT CHANGING THE MASKS AT THIS POINT WOULD INCREASE THE COST OF THE MACHINE OUT OF PROPORTION TO THE BENEFITS. THAT ALSO SEEMS TO RULE OUT HAVING BUILT IN EDIT FUNCTION ANY TIME IN THE NEAR FUTURE.

BEFORE CLOSING THIS ARTICLE, I HAVE TO POINT OUT THAT THE INFORMATION I GOT WAS FROM CASUAL CONVERSATION WITH OSI PERSONEL. IN A COMPANY OF THAT SIZE, I DOUBT THAT ANY ONE PERSON KNOWS EVERYTHING ABOUT OPERATIONS, SO SOME OF THE INFORMATION COULD BE IN ERROR.

)****ON VARIABLE NAMES****

I GET SEVERAL PROGRAMS PER WEEK IN FOR EVALUATION (SOMETIMES SEVERAL A DAY), AND A CONTINUING AND COMMON PRBLEM IS BEGINNING TO DRIVE ME UP A WALL. I AM SURE THAT THOSE OF YOU WHO USE AND EVALUATE SOFTWARE FROM OTHERS ARE HAVING THE SAME PROBLEM.

I HAVE NOW TAKEN A SOLUMN OATH THAT I WILL BURN UNUSED AND UNREAD ANY PROGRAM THAT COMES INTO THIS OFFICE WITH MORE THAN TWO (2) VARIABLES EACH STARTING WITH THE LETTERS X AND Y. FOR SOME DAMNED FOOL REASON, PROBABLY STARTING WITH A DEMENTED HIGH SCHOOL ALGEGRA TEACHER, EVERYONE THINKS THAT EVERY VARIABLE MUST START WITH ONE OF THOSE TWO LETTERS-OR IF THEY LEARNED TO PROGRAM IN COLLEGE, THEY USE 'I'.

IN ONE OF THE WORST CASES OF COMPUTER TONGUE TIE THAT I HAVE EVER PERSONALLY WITNESSED, WE RECENTLY RECIEVED A DOODLER PROGRAM IN THE MAIL THAT HAD NO LESS THAN 16 VARIABLES STARTING WITH X OR Y. HE USED X1,X2,X3,X4,X5,XB,----XADNASEUM!!-AND THEN STARTED WITH 'Y'.

I WOULD LIKE TO HUMBLY POINT OUT THAT WHILE BASIC ONLY RECOGNIZES TWO LETTERS OF A VARIABLE NAME, HUMANS CAN RECOGNIZE MORE. BASIC WILL ALLOW THE USE OF WONDERFULL VARIABLE NAMES LIKE 'CORNER','BALL', 'MONTHLY','YEARLY','MINE', AND 'YOURS'. HUMANS, INCLUDING YOU, WHO READ THE PROGRAM LATER WILL BE SO GRATEFUL FOR A VARIABLE NAME THEY CAN UNDERSTAND WITHOUT A LOT OF FIGURING.

IN THE INTEREST OF MEMORY ECONOMY, WE SUGGEST THAT PROGRAMERS USE THE FULL NAME THE FIRST TIME IT APPEARS IN THE TEXT AND ONLY THE FIRST TWO LETTERS IN THE FOLLOWING TEXT.

)***BRIGHT IDEA OF THE WEEK***

SOME OF THE HOBBYISTS AROUND YPSILANTI MICHIGAN HAVE BEEN USING A CHEAP AND NOT TO RELIABLE MEANS OF SENDING PROGRAMS TO OTHER HOBBYISTS. HOBBYIST #1 PUTS HIS TAPE RECORDER CLOSE TO THE PHONE WHILE HOBBYIST #2 RECORDS OFF THE PHONE WITH A 98 CENT RADIO SHACK PHONE MIC. SOMETIMES IT EVEN WORKS. I SENT BOB RETELLE A GARBAGE COLLECTOR ROUTINE THAT WAY A FEW WEEKS AGO.

)***ON DISK SYSTEMS-OS65D***

WHILE THIS ARTICLE WILL ASSUME THAT YOU HAVE READ THE OSI DISK BASIC MANUAL, IT WILL NOT ASSUME THAT YOU UNDERSTAND MUCH BEYOND THE LOAD, PUT, AND RUN COMMANDS. IT WILL REPEAT A LOT OF INFORMATION THAT IS ALREADY IN THE MANUAL AND WILL THEREFOR BE BORING TO THOSE WHO ARE ALREADY VERY FAMILIAR WITH THEIR DISKS. I AM SIMPLY GOING TO TRY TO MAKE THAT INFORMATION A LITTLE MORE PALATABLE AND FILL IN A FEW OF THE GAPS.

I AM ONLY GOING TO COVER OS65D IN THIS ARTICLE AS THAT IS THE MORE COMMON SYSTEM AMONG THE SUBSCRIBERS TO THIS JOURNAL. THOSE OF YOU WHO HAVE GONE WHOLE HOG AND BOUGHT THE MORE EXPENSIVE OS65-U ARE HEREBY REFERED TO OUR COMPETITOR 'PEEK(65)'. I AM ALSO GOING TO CONCENTRATE ON THE MORE COMMON 5 1/4' DISK. WHERE THE 8' VALUES DIFFER, I WILL INCLUDE THE DATA FOR 8' IN ( ).
GOOD LORD THAT IS A LOT OF EXPLANATION-AND WE AINT EVEN STARTED YET.

BY A VERY NARROW MARGIN, THE MOST COMMON QUESTION I GET ABOUT DISKS IS 'HOW DO I START A NEW DISK?'. THAT IS NOT A SIMPLE PROBLEM FOR SOMEONE WITH A SINGLE DISK DRIVE. THERE ARE ACTUALLY THREE VALID ANSWERS FOR THAT QUESTION AND ANSWERING IT WILL GIVE US AN EXCUSE TO LOOK AT THE DISK.

AT THE SIMPLEST LEVEL, ANY DISK THAT HAS BEEN ITITIALIZED CAN BE USED FOR PROGRAM AND DATA STORAGE. IN ESSENCE, WHAT INITIALIZATION DOES IS TO WRITE TRACK HEADERS ON THE DISK SO THAT THE DOS (Disk Operatins System) CAN FIND TRACK 12 AND TRACK 14 AND SO ON. ONCE THAT IS DONE, YOU CAN STORE DATA AND PROGRAMS BY TRACK NUMBER. HOWEVER, AT THAT LEVEL IT IS HARD TO TELL WHAT IS ON THE DISK (YOU CANT USE A DIRECTORY PROGRAM) AND YOU HAVE TO BOOT UP WITH ANOTHER PROGRAM BEFORE YOU CAN USE THE NEW DISK.

AT THE SECOND LEVEL, YOU HAVE TO COPY A DIRECTORY ONTO THE DISK AND IT HAS TO BE ON TRACK 12 (8 FOR 8'). NOTE THAT IT IS NOT NECESSARY TO PUT A DIRECTORY PROGRAM ON THE DISK. AS LONG AS TRACK 12 HOLDS DIRECTORY INFORMATION, YOU CAN SAVE AND RUN PROGRAMS BY NAME-AND IF YOU CALL IN A DIRECTORY PROGRAM FROM ANOTHER DISK, YOU CAN EVEN TELL WHAT'S ON THIS ONE. AT THIS LEVEL YOU HAVE A FAIRLY HANDY DISK, BUT YOU STILL HAVE TO BOOT UP WITH ANOTHER DISK BEFOR YOU CAN USE THIS ONE.

TO MAKE A BOOTABLE DISK, YOU HAVE TO HAVE THE I.O. ROUTINES, BASIC, AND, IF YOU WANT TO DO MACHINE CODE, AN ASSEMBLER. THE I.O. ROUTINES AND BASIC ARE ON TRACKS 0-6 (0-4 FOR 8') AND THE ASSEMBLER/MONITOR OCCUPIES TRACKS 7-12 (5-7). TO BOOT UP A DISK, WE HAVE TO HAVE AT LEAST THE FIRST SIX TRACKS. IF YOU HAVE A DUAL DISK SYSTEM, NO PROBLEM-JUST USE OSI'S COPY PROGRAM.

THE AARDVARK JOURNAL IS PUBLISHED SIX TIMES A YEAR BY RODGER OLSEN, AARDVARK TECHNICAL SERVICES, 1690 BOLTON, WALLED LAKE, MI 48088. SUBSCRIPTION RATE: $9.00 A YEAR. COPYRIGHT 1980 BY RODGER OLSEN. FIRST CLASS POSTAGE PAID AT WALLED LAKE, MI 48088

IF YOU HAVE A SINGLE DISK SYSTEM, FORGET THE COPY PROGRAM. IT WON'T WORK. USE THE ONE THAT WE HAVE INCLUDED AT THE END OF THIS ARTICLE. IT WILL COPY THE FIRST 14 TRACKS FOR YOU.

ASSUMING YOU ARE GOING TO RUN BASIC PROGRAMS, ALL YOU NEED ARE THOSE FIRST SIX TRACKS, A DIRECTORY, AND A PROGRAM SOMEWHERE ON THE DISK CALLED 'BEXEC*' WHEN THE DISK BOOTS UP, IT WILL PUT BASIC INTO THE BOTTOM OF MEMORY AND THEN RUN WHATEVER PROGRAM YOU CALLED BEXEC* SEE HOW SIMPLE IT IS-

(1)INITIALIZE A DISK
(2)COPY TRACKS 0-6
(3)COPY A DIRECTORY
(4)PUT ON A BEXEC*

THAT BRINGS US TO AN EXCUSE TO LOOK AT THE OSI 'UTILITIES' - PARTICLULARLY 'BEXEC*' NOW BEXEC* IS A SPECIAL PROGRAM-BUT NOT AS SPECIAL AS IT THINKS IT IS. ACCORDING TO RUMORS AROUND OSI, THAT IS PRONOUNCED AS IF IT WERE B-EXECUTIVE-AS OPPOSED TO A-EXECUTIVE-WHICH NO ONE REMEMBERS EITHER HAVING SEEN OR MISSED. AS WE JUST WENT OVER, DOS WILL LOOK FOR A PGRAM CALLED BEXEC* WHEN IT BOOTS UP AND WILL GET MIGHTY CONFUSED IF IT DONT FIND IT. THE NICE THING IS THAT IT DOENS'T CARE WHERE THAT PROGRAM IS OR WHAT'S IN IT. AS CURRENTLY SET UP, THE OSI BEXEC* DOESN'T DO MUCH. IT WILL CALL 'DIR' OR 'CHANGE' OR POKE INTO BASIC THE KEYWORDS FOR LIST NEW AND CONTROL-C. YOU CAN HOWEVER CHANGE ALL THAT. FOR INSTANCE, ON THE AARDVARK BEXEC*, WE PUT A DIRECTORY, CHANGE, CREATE, AND DELETE ALL ON ONE TRACK AND CALL IT BEXEC*. WE THEREFOR GET A DIRECTORY AS SOON AS WE BOOT UP AND HAVE ALL OF OUR MOST NEEDED ROUTINES ON HAND AS SOON AS WE PUT THE BEXEC* PON A DISK. IF YOU WANT TO DO A SIMILAR PROGRAM, ALL YOU HAVE TO DO IS TO PUT THE INFORMATION IN LINES 25, 10040, AND 10070 OF THE OSI BEXEC* INTO ANY NEW PROGRAM AND THEN ENTER ANY MATERIAL YOU WANT TO HAVE AS AN EXECUTIVE.

IT IS ALSO IMPORTANT TO REMEMBER THAT YOU CAN PUT THE BEXEC* ANYWHERE EXCEPT TRACKS 0-6 AND 12. ON SOME APPLICATIONS DISKS WHERE WE NEED A LOT OF FILE SPACE, WE PUT BEXEC* ON TRACK 7 AND THEN HAVE TRACKS 8-11 AND 13-39 FOR FILE SPACE. THE LETTER WRITER THAT I AM USING TO COMPOSE THIS OCCUPIES TRACKS 7-11 AND THE TEXT IS STORED ON TRACKS 13- 39.

)**CHANGE**

DEFINITELY AN EXAMPLE OF STRANGE PROGRAMMING. THE PROGRAM IS 65 LINES LONG, AND THE WORING PART IS ONLY 4 LINES LONG. WHAT IT DOES IS TO AS MANY AS THREE POINTERS. THE POINTER FOR THE BOTTOM OF BASIC WORK SPACE IS IN LOCATIONS 121 AND 120 (STORED LOW BYTE, HIGH BYTE IN THE BEST 6502 TRADITION). TO ADD 8 PAGE BUFFERS, YOU PEEK LOCATION 121, ADD 8 FOR EACH BUFFER AND POKE IT BACK INTO 121. THE POINTER FOR THE TOP OF WORK SPACE IS IN LOCATION 132 AND 133. IF YOU ASK FOR STORAGE SPACE AT THE TOP OF WORK SPACE, CHANGE PEEKS THOSE TWO LOCATION, SUBTRACTS THE DESIRED SPACE AND POKE THE VALUES BACK IN. THE ONLY OTHER THING IT DOES IS TO POKE THE TERMINAL WIDTH INTO LOCATION 23.

MOST OF WHAT IT DOES IS TO ASK IF YOU ARE SURE, VERY SURE, VERY SURE WITH SUGAR ON IT THAT YOU REALLY, REALLY, REALLY, WANT TO CHANGE THE WORK SPACE.

)***CREATE***

CONSISTS OF IS AN ENTRY IN THE DIRECTORY. TRACK 12 CONSISTS OF 8 BYTE ENTRIES. THE FIRST 6 BYTES ARE THE NAME OF THE FILE AND THE NEXT TWO ARE THE STARTING AND ENDING TRACKS-IN BCD. IF THE FILE IS EMPTY, THE FIRST 6 LOCATIONS ARE FILED WITH ***'S (ASCII 42'S). THAT IS WHY YOU HAVE TO COPY A DIRECTORY RATHER THAT JUST CREATE ONE WITH THE DIRECTORY PROGRAM. THERE IS NO EMPTY SPACE AND A BLANK TRACK WOULD CONFUSE THE DIRECTORY PROGRAM. NOTE THAT THE ENTRY IN THE DIRECTORY DOES NOTHING TO THE DISK BUT TELL DOS WHERE TO FIND THE PROGRAM. WIPING OUT OR CHANGING THE DIRECTORY ENTRY DOES NOTHING TO THE DATA OR PROGRAM. YOU CAN THEREFOR, FOR INSTANCE, EXPAND OR CONTRACT A FILE BY DELETING IT AND RECREATING IT IN THE SAME SPACE WITH A DIFFERENT NUMBER OF TRACKS.

)**DELETE**

MAINLY FINDS THE ENTRY FOR THE PROGRAM AND THEN WRITES *'S INTO THE NAME SPACES.

)** #9 ERRORS **

A 'PARITY' ERROR ACCORDING TO THE BOOKS. THAT MEANS THAT THE DOS SYSTEM THINKS WHAT IS ON THE TRACK IS NONSENSE. IT USUALLY COMES FROM ONE OF THREE PROBLEMS.

(1) DISK SPEED IS OFF
(2) YOU FORGOT TO INITIALIZE THE DISK
(3) OR, MOST COMMONLY, YOU FORGOT TO PUT ANYTHING IN THE FILE YOU JUST OPENED. ONE OF THE NASTY THINGS ABOUT OPENING A FILE (DISK OPEN,6,ETC) IS THAT THE DOS TRIES TO READ WHAT IS IN THE FILE JUST IN CASE YOU WANT TO USE IT LATER. IF THERE IS NOTHING IN THERE, THE DOS GOES BOING AND GIVES YOU A #9 ERROR. OSI'S SOLUTION IS TO PROVIDE YOU WITH 'ZERO' A MACHINE CODE UTILITIY THAT FILLS THE FILE WITH-GUESS WHAT- ZERO'S. THERE ISN'T ANYTHING MUCH MAGIC ABOUT ZEROS, BUT THE SOLUTION'S PRETTY. IT BEING THAT I DON'T LIKE TO TAKE UP A TRACK WITH A UTILITY LIKE ZERO, I DONT USE IT. SINCE IT DOESNT MATTER WHAT IS IN THE FILE BEFORE YOU USE IT, AND AS ALL FILES ARE AT LEAST ONE TRACK LONG, I USUALLY TYPE IN 'PU (FILE NAME)' AS SOON AS I CREATE THE FILE AND STORE BEXEC* OR THE CREATE UTILITY IN IT. IT WILL INITIALIZE THE FILE FINE AND WILL BE OVERWRITTEN AS SOON AS I PUT SOME REAL DATA IN. SAVES CALLING ANOTHER ROUTINE.

)**DATA FILES - WHAT THEY FORGOT TO MENTION**

I AM NOT EMBARRASSED TO ADMIT THAT I BLEW AWAY ABOUT 10 DAYS OF WORK WHEN I GOT MY FIRST DISK SYSTEM FROM OSI. EMBARRASSED I'M NOT - MAD I AM. I LOST THE TIME BECAUSE OF THE INADEQUACIES OF THE INSTRUCTIONS ON HOW TO USE DISK FILES. MAYBE I CAN SAVE SOMEONE ELSE THE SAME PROBLEMS.

THIS IS WHAT HAPPENED. I FOLLOWED OSI'S EXAMPLE OF HOW TO CREATE OPEN, AND CLOSE A DATA FILE AND IT WORKED. I THEREFORE WROTE A PROGRAM OF MY OWN TO GIVE AND SCORE TESTS. WHEN I HAD THE PROGRAM PARTIALLY TYPED IN AND DEBUGGED WITH DUMMY DATA, I OPENED A FILE TO DO A REAL TEST -- AND WIPED OUT MY PROGRAM -- SEVERAL TIMES.

BLAST! THAT NEVER HAPPENED ON DEC SYSTEMS!

I SPENT CONSIDERABLE TIME FINDING THE PROBLEM. WHAT OSI FORGETS TO MENTION IS THAT THE SYSTEM STORES THE FILE SPACE - OR LACK OF FILE SPACE - WHEN IT STORES THE PROGRAM. THE RATHER SIMPLE MINDED DOS STORES EVERYTHING FROM 327E (317E FOR 8") UP TO THE LIMIT OF THE WORKSPACE. WHEN IT COMES BACK, IT IS PUT INTO THE SAME SPACE - WHETHER YOU CHANGED THE WORKSPACE OR NOT. THAT MEANS TWO THINGS:

(1) IF YOU WROTE THE PROGRAM WITHOUT DISK BUFFERS, YOU WIPE IT OUT WHEN YOU OPEN THE FILES AS THEY GO IN THE SAME SPACE WHERE YOU DONE PUT THE PROGRAM.

(2) PROGRAMS WITH BUFFERS TAKE A LOT OF DISK SPACE. IF THE BUFFERS ARE THERE, THEY GO ON DISK TOO. THAT MEANS THAT A 6K PROGRAM WITH TWO BUFFERS (AT 2K EACH) TAKES FIVE TRACKS ON THE DISK (10K) TO STORE.

ONCE YOU UNDERSTAND THAT, IT IS ACTUALLY A HANDY SYSTEM AS PROGRAMS USING FILES ARE ALWAYS SET UP AS SOON AS THEY ARE CALLED. CHANGE NEED BE RUN ONLY BEFORE THE PROGRAM IS WRITTEN

WHAT IF YOU GOOF? WHAT IF YOU WRITE THE PROGRAM IN NORMAL WORKSPACE AND THEN NEED BUFFERS? THERE ARE SOME SOLUTIONS, BUT THEY REQUIRE A LOT OF MEMORY OR A TAPE INTERFACE. YOU WILL NEED ENOUGH MEMORY FOR THE BUFFERS, THE PROGRAM, AND AN INDIRECT FILE SPACE THE SIZE OF THE PROGRAM.

TO ADD A BUFFER, LOAD THE PROGRAM AND STORE IT IN AN INDIRECT FILE. TYPE IN 'NEW'. PEEK (133), IF YOU HAVE NO BUFFERS, THAT VALUE SHOULD BE A 50. ADD 8 FOR EACH BUFFER YOU WANT (12 FOR 8') AND POKE THE NUMBER BACK. NOW RETURN THE PROGRAM TO MAIN MEMORY WITH A CONTROL X. IT WILL BE TREATED AS NEW ASCII INPUT AND PUT IN THE CORRECT MEMORY LOCATION. NOW STORE IT ON DISK AND PRAY THAT YOU REMEMBERED TO MAKE THE DISK FILE BIG ENOUGH TO HANDLE THE BUFFERS.

YOU SHOULD BE ABLE TO DELETE BUFFERS THE SAME WAY, BUT I HAVEN'T TRIED IT. WE DID NEED TO ELIMINATE SOME BUFFERS RECENTLY. I HAD SET UP FILE BUFFERS ON THE ADVENTURE PROGRAMS AND THEN FOUND THAT IF I ELIMINATED THE BUFFERS, THE EXTRA MEMORY WOULD LET ME STORE STUFF WITHOUT FILES. I CHEATED. I READ THE PROGRAM OUT TO THE CASSETTE INTERFACE, RAN CHANGE, AND READ IT BACK IN. I SAVED 4K.

)**CHANGING THE RECORD SIZE**

RANDOM ACCESS FILES UNDER 65D ARE PRESET AT 128 BYTES IN LENGTH. THAT SIMPLY MEANS THAT WHEN YOU GET A RECORD FROM A RANDOM ACCESS FILE, DOS WILL COUNT OUT 128 BYTES FOR EACH RECORD AND STOP AT THE ONE YOU WANT. NOT BAD, BUT RATHER WASTEFULL OF DISK SPACE IF YOU ARE STORING 8 BYTE PART NUMBERS IN AN INVENTORY PROGRAM AND NOT ENOUGH IF YOU ARE STORING QUESTIONS FOR A QUIZ. OSI DOES MENTION THAT THE RECORD LENGTH CAN BE CHANGED BY SOFTWARE-BUT THEY FORGOT TO MENTION HOW TO DO IT. WE GOT THIS SET OF POKES FROM AN OSI DEALER AND HAVE USED IT ON 8' DISKS.

BYTES PER RECORD	POKE 12042	12076	5' POKE12042	POKE12076
16	192	4	194	4
32	96	5	96	5
64	48	6	48	6
128	24	7	24	7
256	12	8	12	8
512	6	9	6	9
1024	3	10	3	10

)** I.O. FUNCTIONS DISK VERSION **

I CANT TAKE CREDIT FOR THE INFORMATION IN THIS SECTION. IT WAS WRITTEN BY AN AUTHOR WHO WOULD RATHER REMAIN ANONYMOUS

MEMORY LOCATIONS OF INPUT ROUTINES

24F6	ACIA (CPU BOARD)
252B	KEYBOARD
2518	UART (430/550 BOARDS)
2386	NULL
2389	MEMORY INPUT
23A1	MEMORY BUFFER DISK INPUT
23F0	MEMORY BUFFER DISK INPUT
24B0	ACIA ON 550 BOARD

MEMORY LOCATIONS FOR DISK OUTPUT ROUTINES

242D	ACIA ON CPU
2599	VIDEO OUTPUT
250D	UART ON 430/550
249F	LINE PRINTER INTERFACE
2390	MEMORY OUTPUT
2302	MEMORY BUFFER DISK
2403	MEMORY BUFFER DISK
242B	ACIA ON 550 BOARD

)USING THOSE DISK FUNCTIONS IN YOUR BASIC PROGRAMS

(THE FOLLOWING IS BY ONE OF OUR AUTHORS - WHO PREFERS ANONYMITY)

MOST OF US HAVE USED BASIC BEFORE WE GOT OUR DISK SYSTEMS, AND WE SOMETIMES OVERLOOK THE POWER OF THE OSI DISK FUNCTIONS. I HOPE IN THIS ARTICLE TO GIVE SOME VERY USEFUL APPLICATIONS FOR SOME OF THE DISK FUNCTIONS.

PROBABLY THE MOST CONVENIENT AND EASIEST TO USE IS THE DISK! GO NNNN FUNCTION. THIS IS VERY HANDY AS A USR TYPE FUNCTION, AS LONG AS NO DATA NEED BE TRANSFERED DIRECTLY. THE GO FUNCTION CAN BE USED WITH NO INITIAL SET UP. FOR EXAMPLE:

```
     10  DISK!"GO  252B"
     20  X=PEEK(9815):  REM  X=PEEK(9804)  ON  C1-PMF
```

WILL READ THE KEYBOARD IN A SIMILAR MANNER, ANY PROGRAM CAN BE CALLED , LIKE A CLEAR THE SCREEN ROUTINE.

ANOTHER CONVENIENT COMMAND IS THE DISK!"CALL NNNN=TT,S". THIS TELLS THE SYSTEM TO CALL THE DATA ON TRACK TT, STARTING AT SECTOR S AND PUT IT IN MEMORY STARTING AT NNNN. IN MOST CASES A USR ROUTINE IS STORED IN THE FORM OF DATA STATEMENTS AND POKED INTO PLACE. THIS TAKES MORE THAN FOUR TIMES AS MUCH SPACE AS THE ROUTINE ITSELF USES. HOWEVER, IF YOU USE A CALL ROUTINE, IT ONLY TAKES ENOUGH PROGRAM SPACE FOR THE ONE STATEMENT AND ENOUGH SPACE TO STORE THE ROUTINE ITSELF. FOR EXAMPLE, IF YOU HAVE A FUNCTION TO BE LOCATED AT 5800 (HEX), ONCE YOU HAVE THE PROGRAM ASSEMBLED AT 5800 USE A 'PUT' COMMAND TO SAVE IT, SAY, ON TRACK 39. THEN TO LOAD IT IN YOUR BASIC PROGRAM JUST ADD THE FOLLOWING STATEMENT:

```
     10  DISK!"CALL  5800=39,1"
```

THEN TO USE IT JUST USE THE USR OR DISK!"GO ----" FUNCTION.

THE USE OF THE PRINT AND INPUT TO DEVICE NO. FUNCTIONS ARE ALSO VERY HANDY, BUT IN COMES CASES REQUIRE SOME INITIAL SETUP. ON THE NON-DISK SYSTEMS, IT WAS VERY DIFFICULT TO PRINT TO THE SCREEN WITHOUT SCROLLING. THIS USUALLY REQUIRED A LOOP AND SOME FORM OF TH MID$ FUNCTION. NOW WITH THE SETTING OF A MEMORY POINTER, YOU CAN USE PRINT COMMANDS JUST AS BEFORE AND BE ABLE TO PRINT ANYWHERE ON THE SCREEN WITHOUT SCROLLING. FIRST YOU MUST USE THE DISK!"MEM NNNN,MMMM" COMMAND WHERE NNN REFERS TO THE INPUT LOCATION, AND MMM REFERS TO THE OUTPUT LOCATION. SUPPOSE IN A GAME YOU WANTED TO PRINT OUT THE PLAYER'S SCORE (SC) AT THE TOP OF THE SCREEN AT LOCATION D104 (HEX). TO SET UP THE MEMORY, YOU WOULD USE THE STATEMENT:

```
     10  DISK!"MEM  D104,D104"
```

AS YOU MAY HAVE NOTICED, I PUT THE SAME ADDRESS IN BOTH INPUT AND OUTPUT FIELDS. THIS DISK FUNCTION, UNLIKE THE IO FUNCTION, REQUIRES THAT BOTH FIELDS ARE FILLED. SINCE IN THIS CASE WE AREN'T USING MEMORY INPUT IT DOESN'T MATTER WHAT IS PUT IN THE INPUT FIELD. NOW THAT THE FIELDS ARE SET UP, ALL WE NEED TO DO IS:

```
     20  PRINT#5,"YOUR  PRESENT  SCORE  IS";SC;
     30  PRINT#9
```

THE REASON FOR THE ; AT THE END OF LINE 20 IS TO PREVENT THE PRINTING OF A CARRIAGE RETURN AND LINE FEED, WHICH ARE GRAPHIC CHARACTERS. THEN YOU MUST GENERATE THIS CR LF. LINE 30 ACCOMPLISHES THIS WITH A PRINT TO A NULL DEVICE. IF THE SCORE CHANGES AND YOU WANT TO PRINT THE NEW SCORE, YOU MUST DO THE DISK!"MEM FUNCTION AGAIN SINCE IT IS SELF INCREMENTING AND IS POINTING AT THE LAST LINE PRINTED.

NOW TO GET A LITTLE TRICKIER. IF YOU WANT TO INPUT DATA AND PRINT TO THE SCREEN AS IT IS INPUT WITHOUT SCROLLING, IT REQUIRES THE USE OF THE DISK!"MEM FUNCTION AND THE DISK!"IO, NN" FUNCTION. FIRST, SET UP THE MEMORY AS BEFORE:

```
     10  DISK!"MEM  D104,D104
```

NOW YOU MUST TRANSFER OUTPUT CONTROL TO MEMORY WITH THE LINE:

```
     20  DISK!"IO  ,10"
```

NOW YOU ARE READY TO DO YOUR INPUT STATEMENT:

```
     30  INPUT  "WHAT  IS  YOUR  MOVE";A
```

IN THIS METHOD YOU CAN'T PREVENT THE PRINTING OF THE CR AND LF CHARACTERS. TO GET RID OF THESE CHARACTERS IS A PROBLEM. THE EASIEST WAY I HAVE FOUND IS TO FIGURE THE INPUT DATA STARTS AND REPRINT IT:

```
     40  DISK!"MEM  D114,D114
     50  PRINT#5,A;"          ";:PRINT#9
```

THEN YOU MAY WANT TO RESET THE IO DEPENDING ON THE PROGRAM WITH:

```
     60  DISK!"IO  ,02"
```

ANOTHER USEFUL FUNCTION IS THE RUN "FILE NAME". IF YOU HAVE A LONG PROGRAM AND ARE TIGHT ON SPACE, YOU CAN PUT THE INSTRUCTION IN A SEPARATE PROGRAM. SIMPLY WRITE A PROGRAM WITH JUST THE INSTRUCTIONS, THEN MAKE THE LAST STATEMENT OF THE PROGRAM:

```
     1000  RUN"GAME"
```

IT IS ALSO HANDY TO USE THE RUN COMMAND IN PLACE OF AN END STATEMENT TO LOAD BEXEC*. THIS WAY, IF YOU HAVE A MENU PROGRAM IN BEXEC* IT WILL BE RESTARTED.

```
1 REM *BY FRANK KNEPP*
2 REM *   04/16/80   *
10 FORCL=1TO26:PRINT:NEXT:PRINT"   SINGLE DRIVE COPY":PRINT:PRINT
20 PRINT"COPIES TRACKS 1 - 14":PRINT:PRINT
30 PRINT"TRACK 0 MUST BE COPED":PRINT:PRINT"WITH TRACK 0 READ/WRITE
40 PRINT:PRINT:POKE132,250:POKE133,55
90 GOSUB900
100 DISK!"CA 3800=01,1":DISK!"CA 4000=02,1":DISK!"CA 4800=03,1"
110 GOSUB800
120 DISK!"SA 01,1=3800/8":DISK!"SA 02,1=4000/8":DISK!"SA 03,1=4800/8"
130 GOSUB900
140 DISK!"CA 3800=04,1":DISK!"CA 4000=05,1":DISK!"CA 4800=07,1"
150 GOSUB800
160 DISK!"SA 04,1=3800/8":DISK!"SA 05,1=4000/8":DISK!"SA 07,1=4800/8"
170 GOSUB900
180 DISK!"CA 3800=06,1":DISK!"CA 3900=09,1":DISK!"CA 4000=08,1"
190 DISK!"CA 4800=10,1":GOSUB800
200 DISK!"SA 06,1=3800/1":DISK!"SA 09,1=3900/5":DISK!"SA 08,1=4000/8"
210 DISK!"SA 10,1=4800/8":GOSUB900
220 DISK!"CA 3800=11,1":DISK!"CA 3C00=12,1":DISK!"CA 3D00=12,2"
230 DISK!"CA 3E00=12,3":DISK!"CA 3F00=12,4":DISK!"CA 4000=13,1"
240 DISK!"CA 4800=14,1":GOSUB800
250 DISK!"SA 11,1=3800/4":DISK!"SA 12,1=3C00/1":DISK!"SA 12,2=3D00/1"
260 DISK!"SA 12,3=3E00/1":DISK!"SA 12,4=3F00/1":DISK!"SA 13,1=4000/5"
270 DISK!"SA 14,1=4800/8":PRINT:PRINT"Copy complete!":END
800 PRINT:PRINT"Insert destination disk":PRINT"Press ESC key
810 IFPEEK(57100)<>222THEN810
820 RETURN
900 PRINT:PRINT"Insert source disk":PRINT"Press ESC key
910 IFPEEK(57100)<>222THEN910
920 RETURN
```

)*** TRACK 0 COPIER ***

AFTER BOOTING UP, TYPE 'EXIT' (NO QUOTES), A* WILL APPEAR ON SCREEN.

```
     FOR  8"  DISKS:  CA  0200=01,2
     FOR  5"  DISKS:  CA  0200=13,1
```

THEN TYPE GO 0200, YOU WILL GET OSI'S DISKETTE UTILITIES. 2) IS TRACK 0 READ/WRITE. SELECT 2 **AND THE INSTRUCTIONS FOR READING AND WRITING TRACK 0 WILL BE DISPLAYED.** FOR COMMAND?, TYPE IN **R4000, THE COMPUTER WILL COME BACK WITH THE INSTRUCTIONS AGAIN TO SIGNAL THAT THE OPERATION IS COMPLETE.** NOW REMOVE THE MASTER DISK FROM THE DRIVE AND PUT IN YOUR INITIALIZED NEW DISK. **TYPE:**

```
     W4000/2200,12  (8")
     W4000/2200,8  (5")
```

GO BACK TO THE MASTER DISK, USE E TO EXIT FROM THE TRACK 0 PROGRAM, GO TO 0200 AGAIN, AND USE **1) COPIER** TO COPY TRACKS 1-76 (8") OR 1-39 (5").

)**'TRUE' RANDOM NUMBERS**

IT IS AN OPEN SECRET THAT THE RANDOM NUMBER FUNCTION IS NOT PARTICULARLY RANDOM. UNFORTUNATELY, IT BRINGS UP THE SAME SEQUENCE OF 'RANDOM' NUMBERS AFTER EACH COLD START. A FURTHER PROBLEM THAT MOST SIMULATORS TEND TO IGNORE IS THAT THE SEQUENCE REPEATS. IN THE SYSTEM USED BY OSI, THE SEQUENCE IS 1861 NUMBERS LONG. 1862 STARTS ALL OVER AGAIN IT IS USUALLY THOUGHT OF AS A PROBLEM FOR GAMES PROGRAMMERS, BUT IT ALSO EFFECTS OTHER KINDS OF SIMULATION. THE PATTERNS SET UP BY THE GENERATOR ARE SUBTLE BUT DEFINITE. IF, FOR INSTANCE, YOU SET UP A CARD GAME WHICH BASES THE NEXT CARD EACH TIME ON THE NEXT RANDOM NUMBER, THE PLAYERS, AFTER MANY GAMES, CAN SOMETIMES GET A "FEEL FOR" THE BEST TIMES TO BET. ON A MORE SERIOUS LEVEL, IT AFFECTS LARGE SCALE SIMULATIONS. FOR INSTANCE, I WAS RECENTLY ASKED FOR ADVICE ON A PROGRAM THAT WAS SUPPOSED TO TELL WHETHER COINS FOUND IN FRANCE COULD HAVE BEEN MINTED AT A SPECIFIC MINT IN ROME. THE RESEARCHER HAD INFORMATION ON THE PROBABLE ERROR IN SIZE THAT COULD BE CREATED BY THE MINTING METHODS OF THE TIME. WHAT HE WAS ATTEMPTING TO DO WAS TO MINT A MYTHICAL SET OF COINS FROM WITHIN THE LIMITS DECIDED BY PROBABLE ERROR EXPERIENCE WITHIN THE MINT. HE WOULD THEN MATCH UP THE FICTITIOUSLY MINTED SET TO THE SET THAT WAS FOUND AND CALCULATE THE PROBABILITY OF THEM BEING FROM THE SAME SET. HE WAS DOING IT THE HARD WAY AS THERE WERE BETTER STATISTICAL METHODS, HOWEVER, HIS IMMEDIATE PROBLEM WAS THAT HE WAS RUNNING SEVERAL THOUSAND TRIALS IN EACH RUN AND THE REPEATING PATTERN IN THE RANDOM NUMBER GENERATOR WAS MAKING THE OUTCOMES REPETITIOUS. IT WAS AN INTERESTING PROBLEM, AND ONE FACED BY GAMES PLAYERS WHO DO NOT WANT A DIFINITE, EVEN IF SUBTLE, REPETIOUS PATTERN IN THEIR GAMES.

FORTUNATELY, THERE ARE SIMPLE AND ELEGANT SOLUTIONS. MOST COMPUTERISTS SOLVE THE "SAME START" PROBLEM BY HAVING THE COMPUTER DRAW SEVERAL RANDOM NUMBERS BEFORE THE GAME STARTS. THAT WAY THE GAME WILL START AT A DIFFERENT POINT IN THE RANDOM SEQUENCE EACH TIME. I SEE THREE COMMON METHODS IN MOST OF THE PROGRAMS SENT IN FOR EVALUATION.

THE FIRST IS TO HAVE THE PLAYER PUT IN A NUMBER TO START THE GAME. I.E.
100 INPUT "INPUT A NUMBER FROM 1 TO 1000";NUMBER
110 FOR X=1TONUMBER:Y=RND(8):NEXT THAT IS NOT BAD, (I USED TO USE IT), BUT IT MEANS THAT PLAYERS CAN PICK THEIR GAMES AS EACH GAME THAT STARTS WITH 500 WILL BE THE SAME AS EVERY OTHER GAME THAT STARTS WITH 500. TO GET AROUND THAT, THE COMPUERISTS OFTEN TRY TO DISGUISE THE RANDOM DRAW. FOR INSTANCE BY ASKING FOR A NAME OR DATE.
100 INPUT"WHAT IS YOUR NAME";N$
110 FOR X=1TOASC(N$):Y=RND(8):NEXT NOT BAD, BUT IT SUFFERS FROM THE SAME PROBLEMS AS METHOD ONE. EVEN WORSE, EVERY GAME FOR "BRADLEY" WILL START THE SAME WAY.

WE HAVE A SOLUTION WHICH I RECOMMEND FOR POLLED KEYBOARDS. IT IS SIMPLE, REQUIRES A MINIMUM OF INPUT BY THE USER, AND CAN'T BE "FIXED". AS WE HAVE MENTIONED IN PREVIOUS PUBLICATIONS, THE SHIFTS CAN BE POLLED AT LOCATION 57100 (BASE 10) WITHOUT DOING ANY POKING OR TURNING OF THE KEYBOARD STROBE. IF THERE IS NO KEY PRESS, A 540 BOARD (C2/4/8) WILL RETURN A 1 IF YOU LOOK AT 57100. A 600 BOARD (C1/SUPERBOARD) WILL RETURN A 254. IF ANY OTHER NUMBER IS SEEN, ONE OF THE KEYS IS PRESSED. WE USE IT THIS WAY:

```
100  ?"PRESS  SHIFT  KEY  TO  START":?  (REM  PRINT  -  NOT  INPUT)
110  IF  PEEK(57100)<>1  AND  PEEK(57100)<>254  THEN  X=RND(8):GOTO110
```

WHEN YOU PUT IN THAT PROGRAM, THE SYSTEM WILL IDLE THERE DRAWING RANDOM NUMBERS UNTIL THE SHIFT KEY IS TOUCHED. IT WILL THEN FALL THROUGH AND START THE PROGRAM. IT IS EASY TO USE AS A SINGLE TOUCH BY THE USER IS ALL THAT IS REQUIRED. IT CAN'T BE FOOLED AS THAT SEQUENCE ON A ONE MEG C1-P CYCLES AT ABOUT 55 NUMBERS PER SECOND. YOU CANNOT PICK WHERE YOU ARE GOING TO START.

THE PROBLEM WITH MAKING THE SEQUENCE NON-DETERMINISTIC HAS LITTLE TO DO WITH THE FACT THAT THERE ARE ONLY 1861 NUMBERS IN A SEQUENCE. AFTER ALL, DIE ONLY HAVE 6 NUMBERS AND AS MANY EXPENSIVE LESSONS HAVE TAUGHT, THE BLASTED THINGS ARE VERY RANDOM. THE PROBLEM COMES IN THE FACT THAT WE TEND TO CHOOSE THEM AND USE THEM IN EXACTLY THE ORDER THE COMPUTER GIVES THEM TO US. WE NEED TO MIX UP THE WAY WE ACCEPT THEM. THE SAME PROBLEM WAS FACED BY THE MAKERS OF THE FIRST CONSUMER ORIENTED DIGITAL COMPUTERS - PINBALL MACHINES. THEIR SOLUTION CAN ALSO BE USED BY US.

THEIR PROBLEM WAS TO RANDOMLY PICK A MATCH NUMBER. I CAN'T SPEAK WITH EXPERTISE ON ALL

OLD PINBALLS, BUT THE ONE I DISASSEMBLED HAD A CLEVER GIMMICK. IT COUNTED WITH A SET OF COG WHEELS, SIMILAR IN PRINCIPAL, I THINK, TO AN OLD ADDING MACHINE. WHAT THEY DID WAS TO SET ONE WHEEL TO COUNT BACKWARDS RATHER THAN THE USUAL COUNT FROM 0 TO 9. THAT ALONE WOULD NOT HAVE DONE IT AS THE SCORE WOULD STILL HAVE MATCHED UP ONLY ON ONE SPECIFIC SET OF NUMBERS. WHAT THEY DID WAS TO USE THE PLAYERS' INPUT TO RANDOMIZE THE RANDOM NUMBER GENERATOR. WHILE EVERY BUMPER INCREMENTED THE SCORE COUNTERS, THE RANDOMIZER WAS SET UP ONLY TO RECOGNIZE THREE OR FOUR OF THE INPUTS. HOW MANY TIMES THOSE WERE HIT DECIDED HOW MANY TIMES IT DECREMENTED. IT WAS THEREFORE OUT OF SYNC WITH THE SCORE BY A RANDOM NUMBER CAUSED BY THE PLAYERS' ACTIONS.

WE CAN USE THE SAME SYSTEM WITH A COMPUTER. FOR INSTANCE, WE DID A CONCENTRATION GAME SEVERAL MONTHS AGO. THERE ARE 8 PLAYER INPUTS PER ROUND. EACH PLAYER GIVES THE ROW AND COLUMN OF THE TWO BOXES HE WANTS TO EXPOSE. TO MAKE EACH GAME DIFFERENT, WE COULD TWEEK THE RANDOM NUMBER GENERATOR EACH ROUND BASED ON ONE OR MORE OF THE INPUTS. FOR INSTANCE:

```
100 INPUT"ROW AND COLUMN";X,Y:FORI=1TOY:R=RND(8):NEXT
```

THE RANDOM NUMBER GENERATOR WOULD THEN BE AT A DIFFERENT POINT EACH TIME THE PLAYER DID AN INPUT. THEORETICALLY, HE COULD REPEAT THE EXACT INPUTS EACH TIME AND THEREFORE FOOL THE SYSTEM, BUT IT AIN'T BLOODY LIKELY.

WE CAN ALSO INCREMENT THROUGH THE 1861 NUMBERS BY AN INCREMENT OTHER THAN 1. IF WE STEP THROUGH BY 2, WE GET A DIFFERENT SEQUENCE AS WE DO IF WE INCREMENT THROUGH BY 3 AND SO ON. WE USED THAT ONE ON THE COIN PROBLEM. WE INPUTED TWO USER DECIDED NUMBERS EVERY 2500 TRIALS. (HUMAN FRAILITY WAS THE ONLY TRULY RANDOM FACTOR WE HAD AVAILABLE ON THE SYSTEM.) WE USED THE FIRST METHOD GIVEN IN THIS ARTICLE TO PICK A STARTING POINT IN THE SEQUENCE. WE THEN POLLED THE SECONDS FUNTION ON THE CLOCK AND USED THE LAST DIGIT TO DECIDE ON AN INCREMENT TO STEP THROUGH ON THIS TRIAL. IT STILL GAVE US A DETERMINED SET OF NUMBERS ONCE THE TWO INPUTS WERE DECIDED, BUT THE LARGE NUMBER OF POSSIBLE SEQUENCES GAVE A FAIRLY TRUE TO RANDOM OUTPUT.

ONE NOTE: IT DOES NO GOOD TO CHANGE THE RANDOM "SEED". IN OSI BASIC, THE NUMBER IN ( ) IS NOT A TRUE SEED. RND(8) GIVES THE SAME NUMBERS AS RND(2)

)*** GRAPHICS WITHOUT THE BIG NUMBERS***

WE HAVE RECIEVED SEVERAL LETTERS THIS MONTH DETAILING TO US THAT THERE WAS AN ALTERNATIVE METHOD OF PLOTTING GRAPHICS WITHOUT FIGURING EACH POINT.

THE METHOD GIVEN WAS IN EACH CASE THE METHOD THAT WE USE IN THE AARDVARK VIDEO GAMES AS PART OF OUR TECHNIQUE TO MAKE PROGRAMS RUN ON BOTH C1 AND C2 SYSTEMS. IT WAS PRINTED UP IN AN OLD CATALOG OF OURS, BUT I AM GOING TO REPEAT IT HERE IN SELF DEFENSE-JUST SO THAT I WONT HAVE TO READ ANY MORE LETTERS ABOUT IT.

ONE OF THE SIMPLER WAYS TO DO GRAPHICS IS TO PICK A POINT ON THE SCREEN, I USUALLY USE THE UPPER LEFT CORNER, AND RATHER THAN TRY TO GET ABSOLUTE VALUES FOR POSITIONS, FIGURE THEM ALL AS BEING SO MANY LINES DOWN AND SO MANY LINES OVER FROM THAT POINT. A TYPICAL PROGRAM SEGMENT TO DRAW A SIMPLE CROSS WOULD LOOK LIKE THIS

```
100LINE=32:CORNER=53248:REM ON C2/4 SYSTEM LINE=64
110X=15:FORY=1TO20:POKECO+X*LI+Y,161:NEXT
120Y=15:FORX=1TO20:POKECO+X*LI+Y,161:NEXT
```

THERE ARE SEVERAL NICE THINGS ABOUT THE SYSTEM. FIRST IT IS EASIER TO VISUALIZE THE VALUES YOU NEED. IT MIGHT BE UNHANDY TO FIGURE IN YOUR HEAD A VALUE LIKE 54113-65, BUT IT IS EASY TO PICTURE THAT 1 LINE LESS IS A MOVE UP AND ONE SPACE LESS IS A MOVE LEFT. YOU CAN THEREFORE DO GRAPHICS BY INCREMENTING AND DECREMENTING A LINE AND A SPACE COUNTER. IT ALSO HAS THE ADVANTAGE OF MAKING YOUR PROGRAMS MORE PORTABLE AND CHANGEABLE. BY CHANGING THE CORNER POINT, YOU CAN REPOSITION THE ENTIRE DISPLAY ON THE SCREEN. IF YOU MOVE UP FROM A C1 TO A C4, CHANGING THE LINE VALUE AND THE CORNER POINT WILL ADAPT THE PROGRAM TO THE NEW SYSTEM.

AS SEVERAL READERS POINTED OUT, IT ALSO MAKES IT FAIRLY EASY TO GRAPH FAIRLY COMPLEX FUNCTIONS.

)**ON THE CONT FUNCTION**

IF I REMEMBER RIGHT, THE MANUAL SAYS THAT THIS FUNCTION IS USED TO CONTINUE A PROGRAM AFTER A STOP COMMAND. I DONT REMEBER EVER USING FOR THAT, BUT IT DOES HAVE TWO IMPORTANT USES. (1)IF YOU ACCIDENTLY HIT THE BREAK KEY (AND WHO DOESN'T WITH AN OSI KEYBOARD), YOU CAN CONTINUE THE PROGRAM (ASSUMING YOU DONT CHANGE ANY PROGRAM LINE, BY TYPING 'CONT'. (2) MOST IMPORTANT, IT IS A GOOD DEBUGGING TOOL. IF YOUR PROGRAM IS GOING HAYWIRE, YOU CAN HIT BREAK INTENTIONALLY. YOU CAN THEN PRINT OUT THE VALUE OF ANY VARIABLE WITHOUT DISTURBING THE PROGRAM. IF YOU WANT TO KNOW WHAT HAPPENED TO X, HIT BREAK, TYPE 'PRINTX'(RETURN) AND THEN ENTER 'CONT'. YOU CAN EVEN CHANGE A VARIABLE- TYPE IN 'A$(L)='HERE'' FOR INSTANCE- AND THEN CONTINUE THE PROGRAM.

IT CAN BE A POWERFULL DEBUGGING TOOL- AND IT ONLY APPEARS IN BASIC, THE BIGGER LANQUAGES CANT DO THAT.

)***WE GET LETTERS***

PLACING A 50 TO 100 UF ELECTROLYTIC CAPACITOR BETWEEN PIN 40 OF THE 6502 AND +5 VDC WILL PROVIDE AN AUTO START WHEN THE SYSTEM IS TURNED ON. (ED. NOTE: HAVEN'T TRIED IT - THANKS TO THE OLD TV REPAIR DAYS, I CAN'T IMAGINE USING AN ELECTROLYTIC CAP NEAR MY 6502 - BUT IT SOUNDS GOOD.)

MILES HUFFORD WRONG
SCOTTSDALE, AZ

I WOULD LIKE TO KNOW ABOUT DIFFERENCES IN THE MONITOR PROGRAM FOR THE C1P AND THE C2P. I AM ESPECIALLY INTERESTED IN MACHINE LANGUAGE PROGRAMMING.

I'M ALSO INTERESTED IN SLOW SCAN PROGRAMS.

ROBERT BEAM
EVANSTON, IL

(I'M GLAD TO HEAR THAT YOU HAVE AN INTEREST IN MACHINE CODE. HOWEVER, AS A PSYCHOLOGIST, I FEEL THAT I SHOULD TELL YOU THAT THERE ARE RELIABLE CURES FOR MOST FORMS OF MASOCHISM.

THERE IS DARNED LITTLE DIFFERENCE BETWEEN THE C1 AND C2/4 MONITORS. AS OSI IS RATIONAL, THEY AVOID CODE WRITING AS MUCH AS POSSIBLE. THE MAJOR DIFFERENCES ARE IN THE LOCATION OF THE MAJOR IO PORT (ACIA) - ITS AT FC00 IN THE C1 AND F000 IN THE C2 (OR VICE VERSA). ALSO THE C1P MONITOR BRANCHES OUT TO A ROUTINE THAT INVERTS THE KEYBOARD VALUES. R.O.)

CHANGING R15 AND R18 ON THE 540 REV. B-1 VIDEO BOARD TO POTS WILL ALLOW CORRECTION FOR CENTERING - ESPECIALLY OVERSCAN - ON "JUNK" TV SET MONITORS.

R. ESCHAUZIER
ORLEANS, VT

I AM LOOKING FOR HAM RADIO NETS (INFORMATIONAL) FOR OSI OWNERS. (20 METER NET). ANYONE INTERESTED PLEASE WRITE.

RON HALTERMON
WA4HWN
P.O. BOX 8942
NAPLES, FL 33941

IN OSI ROM BASIC RND(X) WON'T WORK. THE "X" MUST BE A POSITIVE NUMBER OR VARIABLE. RND(0) WILL REPEAT THE LAST RND GENERATED. UNDEFINED VARIABLES GIVE THE SAME RESULT AS ZERO. NEGATIVE NUMBERS GIVE A SMALL (E-05 OR -06) NUMBER UNIQUE FOR EACH NEGATIVE NUMBER. ALSO, EACH TIME A COLD START IS DONE, THE SAME SEQUENCE OF RANDOM NUMBERS OCCURS, EVEN IF DIFFERENT "SEEDS" ARE USED. TO GET AROUND THIS AND INSURE THAT YOUR NUMBERS ARE TRULY RANDOM:

```
10 INPUT "ENTER TIME OF DAY";T
20 FOR I=1 TO T: X=RND(T):NEXT
```

KERRY LOURASH
DECATUR, IL

(SEE THE RANDOM NUMBER ARTICLE FOR SEVERAL OTHER METHODS. R.O.)

I PURCHASED A C1 JUST BEFORE CHRISTMAS AND HAVE BEEN ENJOYING THE LITTLE MONSTER EVER SINCE. BUT I AM A BEGINNER AND HAVE PROBLEMS UNDERSTANDING MY MONSTER.

FOR INSTANCE: WHAT IS A DISASSEMBLER? WHAT IS IT GOOD FOR? THESE HEX LOCATIONS IN ROM BASIC; WHAT ARE THEY GOOD FOR? I THOUGHT BASIC WAS A LOT MORE POWERFUL THAN THIS HEX STUFF. WHY GO INTO THIS HEXADECIMAL STUFF WHEN BASIC COMMANDS WERE ORIGINATED, I THOUGHT, PRECISELY TO SIMPLIFY PROGRAMMING. AND HOW DOES ONE GET INTO THESE HEXADECIMIL ROUTINES IN THE FIRST PLACE. IF THEY ARE IN MACHINE CODE, THEN WHY CALL THEM "LOCATIONS IN ROM BASIC"?

YOUR "PEEK A PORT UTILITY" WORKED FINE, BUT I COULDN'T FIGURE HOW TO GET YOUR "INPUT WITHOUT SCROLLS" UP AND RUNNING (ED. NOTE: BOTH PROGRAMMING HINTS ARE IN THE CATALOG), MAINLY BECAUSE I DON'T KNOW WHAT TO DO WITH THAT X=USR(X), I SUPPOSE. AND ABOUT THE ONLY WAY I CAN FIGURE TO GET OUT OF THOSE ROUTINES WITHOUT LOSING MY PROGRAM IS BY HITTING BREAK AND W. IS THAT THE WAY THESE THINGS ARE DONE? CAN YOU EXPLAIN WAT THE ESC, RUB OUT, LINE FEED AND REPEAT KEYS ON MY C1P ARE FOR?

JOHN LEAHY
GONZALES, CA

(WOW! WELL, HERE GOES-- IT IS ADVISABLE, BUT CERTAINLY NOT REQUIRED , TO LEARN THE BASICS OF MACHINE CODE WHEN USING A MICRO-COMPUTER. BASIC IS HANDIER THAN CODE, AND MORE POWERFUL IN THE SENSE THAT YOU CAN DO MORE WITH IT SOONER - I DO 95% OF MY PROGRAMMING IN BASIC BECAUSE I CAN WRITE A BASIC PROGRAM IN ONE TENTH THE TIME THAT IT TAKES TO WRITE MACHINE CODE. BUT BASIC CAN DO LESS - NOT MORE - THAN CODE CAN. A DISASSEMBLER TAKES MACHINE CODE AND BREAKS IT DOWN INTO MNEMONICS (WORDS OR ABBREVIATIONS WHICH MAKE IT EASIER TO READ). FOR INSTANCE, A920 IS A COMMAND TO LOAD THE ACCUMULATOR WITH DECIMAL 32. THE DISASSEMBLER WILL GIVE YOU BACK LDA 32, WHICH IS THE MNEMONIC FOR LOAD A WITH 32. WHEN YOU GET SERIOUS ABOUT PROGRAMMING - AND HAVE A FIT OF MASOCHISM - YOU GET AN ASSEMBLER WHICH TAKES THE MNEMONICS AND GIVES YOU BACK MACHINE CODE. IT DOES PART OF THE WORK FOR YOU LIKE CALCULATING ADDRESSES FOR JUMPS, REMEMBERING WHERE YOU PUT THAT SUBROUTINE, AND REMEMBERING WHAT THE NUMBER CODE IS THE THE COMMAND YOU JUST ENTERED.

AS TO THE INPUTS WITHOUT SCROLLS ROUTINE - YOU DON'T HAVE TO WORRY ABOUT COMING BACK TOTHE PROGRAM. THE LAST BYTE IN A USR ROUTINE IS ALWAYS A RETURN FROM SUBROUTINE AND THE BASIC TAKES UP RIGHT WHREE IT LEFT OFF - EVEN IF IT IS IN THE MIDDLE OF A LINE. TO USE THE INPUT ROUTINE, YOU POKE THE VALUES GIVEN INOT LOCATIONS 11 AND 12. THAT TELLS BASIC WHERE THE MACHINE CODE ROUTINE IS. GO X=USR(X) OR ANY OTHER VARIABLE=USR(OF ANYTHING) AND THE BASIC WILL GO TO THAT LOCATION, EXECUTE THE ROUTINE - IN THIS CASE WAIT FOR A KEYPRESS - AND COME BACK TO BASIC. IN THIS ROUTINE IT WILL LEAVE THE VALUE OF THE KEY PRESSED IN LOCATION 531 WHERE YOU CAN PEEK IT OUT WITHOUT SCROLLING THE SCREEN.

ON THE TOPIC OF HEX LOCATIONS. THEY ARE VALUABLE BECAUSE THEY ALLOW YOU TO MAKE CHANGES IN THE WAY THE SYSTEM OPERATES THAT THE DESIGNERS DID NOT FORSEE OR PLAN FOR. FOR EXAMPLE, THERE IS NO WAY IN BASIC TO CHANGE A LINE LENGTH AFTER THE SYSTEM IS INITIALIZED. IF YOU HAVE A C1-P THAT CN BE A PROBLEM AS YOU WOULD LIKE TO HAVE 23 CHARACTERS ACROSS TO SEE IT WELL ON YOUR TV AND MUST HAVE THE WIDTH SET AT 72 CHARACTERS IF YOU ARE GOING TO MAKE A TAPE. KNOWING WHAT LOCATION BASIC STORES THE WIDTH IN (15) ALLOWS YOU TO CHANGE THE LENGTH WHEN YOU WANT TO BY SIMPLY STORING A NEW NUMBER THERE WITH A POKE STATEMENT.

ESC, RUBOUT, AND LINE FEED ARE THERE BECAUSE BUYING A STANDARD KEYBOARD IS CHEAPER FOR OSI THAN ORDERING ONES MADE SPECIAL. R.O.)

NOTHING IN MY MANUAL IS VERY CLEAR. FOR INSTANCE, HOW DO YOU SAVE A PROGRAM? (SAVING BY TRACK SEEMS TO WORK O.K.). THE BOOK SAYS: RUN "CREATE". NOW WHAT?? DO I PUT IN MY BLANK DISK NEXT? WHEN I DO THIS AND USE "PASS" FOR PASSWORD AND "PROGR1" FOR FILE NAME, IT GIVES ME AN "ERROR #A ERROR IN 510. (510 DISK!"CA 2E79=12,2":GOSUB10000). WHAT I NEED TO KNOW IS A STEP BY STEP PROCEDURE FOR USING THE UTILITIES. I AM FAMILIAR WITH BASIC AND I CAN WRITE PROGRAMS O.K., BUT LINE 510 DOES NOT MAKE MUCH SENSE TO ME, OTHER THAN THE GOSUB. THE INSTRUCTIONS FOR READING AND WRITING CASSETTES ARE EQUALLY CONFUSING. IT SAYS TO CREATE A TAPE WITH DISK! "IO 02", BUT NOTHING HAPPENS - NO LISTING TO THE SCREEN.

ED BRUNBERG
ELK GROVE, CA

(I HOPE THE ARTICLE ELSEWHERE IN THE JOURNAL ABOUT DISKS WILL BE OF SOME HELP. IT ISN'T A TUTORIAL, BUT IT SHOULD CLEAR UP SOME OF YOUR QUESTIONS. HERE ARE SOME SPECIFICS - TO CREATE A FILE FOR A PROGRAM, YOU MUST HAVE A DISK INITIALIZED AND A DIRECTORY ON TRACK 12 (8 FOR 8") YOU RUN CREATE (OR THE AARDVARK BEXEC* WHICH MAKES IT MUCH EASIER), ANSWER PASS FOR PASSWORD IF YOU ARE USING OSI'S CREATE, AND THEN INSERT YOUR DISK. THE REST OF THE INSTRUCTIONS ARE IN THE PROGRAM. IT SOUNDS LIKE YOU ARE GETTING ERRORS BECAUSE YOU ARE USING A DISK WITHOUT A DIRECTORY TRACK. ALWAYS CREATE THE FILE BEFORE WRITING THE PROGRAM. IF, HOWEVER, YOU FORGET TO DO SO, YOU CAN STORE THE PROGRAM TEMPORARILY BY TRACK NUMBER AND THEN ASSIGN A NAME TO IT. I TRY TO REMEMBER AN EMPTY TRACK TO PLACE IT ON SO I CAN CREATE THE FILE IN THE SAME PLACE AND NOT HAVE TO MOVE THE FILE AFTER I GIVE IT A NAME.

THE CA AND SA COMANDS ARE TWO THAT YOU NEED TO KNOW. THEY ARE WAYS OF MOVING PROGRAMS ON AND OFF THE DISK WITHOUT USING THE LOAD AND RUN COMMANDS. THEY ARE HANDY AS YOU CAN MOVE SMALL BITS OF CODE AROUND -ALLOWING YOU TO STORE USER COMMANDS ON DISK AND CALL THEM WHEN YOU WANT- AND THEY LET YOU DIRECTLY MODIFY WHAT IS ON THE DISK. THE LINE YOU QUOTE IN YOUR LETTER IS FROM THE CREATE PROGRAM, WHICH DOES MODIFY THE DISK DIRECTLY. THE LINE SAYS CALL TRACK 12, SECTOR 2 AND PUT IT IN MEMORY STARTING AT LOCATION 2E79. THE PROGRAM WILL THEN GO ON TO MODIFY THE INFORMATION WITH POKE STATEMENTS AND THEN WRITE IT BACK ON DISK WITH A 'SA' COMMAND SUCH AS 700DISK!"SA 12,2=2E79/1 WHICH MEANS SAVE MEMORY STARTING AT 2E79 (HEX) AND GOING UP FOR ONE PAGE (256 BYTES) BY PUTTING IT ON TRACK 12, SECTOR 2. THE PROGRAM WE PUT IN THIS ISSUE TO CHANGE THE NAME OF THE EXECUTIVE, USES A SIMILAR TECHNIQUE.

I DON'T KNOW WHICH SYSTEM YOU ARE USING TO MAKE TAPE, THE PORT ASSIGNMENTS VARY FROM SYSTEM TO SYSTEM, BUT IF YOU USED THE COMMAND EXACTLY AS IN THE LETTER, THERE WERE NO RESULTS BECAUSE YOU JUST ASSIGNED THE INPUT AS TAPE RATHER THAN THE OUTPUT. TO ASSIGN OUTPUT YOU HAVE TO EXECUTE DISK!"IO ,01" (C1PMF PORT YOURS MAY BE DIFFERENT) WITH A COMMA BEFORE THE NUMBER. IO 02 WITHOUT A COMMA SETS ONLY THE INPUT PORT. YOUR SYSTEM DIDN'T DO ANYTHING BECAUSE IT WAS WAITING FOR A TAPED INPUT. R.O.)

MANY TIMES WHILE WRITING PROGRAMS THAT UTILIZE POKE OR USR ROUTINES, THE COMPUTER LOCKS UP DUE TO AN ERROR IN THE PROGRAM (ACCIDENTLY POKES A BASIC LOCATION AND CAN'T WARM START). IS THERE ANY WAY TO RECOVER WITHOUT LOSING THE PROGRAM, PERHAPS BY GOING TO A LOCATION WITHIN THE MONITOR THAT WILL RESTORE BASIC? (OTHER THAN A274)

SCOTT BARKER
ONEIDA, NY

(AS YOU HAVE PROBABLY FIGURED OUT, THE USUAL PROBLEM WITH THE MACHINE WHEN YOU CAN'T WARM START IS THAT YOU HAVE GARBAGED OUT PAGE 0. YOU MAY NOT HAVE FIGURED OUT THAT PAGE 0 IS NOT - REPEAT NOT - AVAILABLE FOR USR FUNCTIONS. TO USE LOCATIONS THERE, YOU HAVE TO PUSH VALUES ON THE STACK AND RECALL THEM AFTER YOU LEAVE. THE USUAL PROBLEM IN A BASIC PROGRAM IS POKING A VARIABLE NAME THAT HAS NOT BEEN GIVEN A VALUE - I USED TO DO THAT FREQUENTLY WHEN WRITING VIDEO DISPLAYS. IN THAT CASE, YOU BREAK, GO INTO MONITOR AND PUT 4C,74, AND A2 IN LOCATIONS 0,1, AND 2. IF YOU HAVE GARBAGED UP ANY MORE OF PAGE 0 THAN THAT, YOU HAVE JUST LOST A PROGRAM. R.O.)

)*** GETTING RID OF A MONSTER ***

)** OR HOW TO DISABLE THE BREAK KEY **

IF YOU HAVE EVER TYPED IN SEVERAL PAGES OF COPY ON DISK - OR BEEN IN THE MIDDLE OF A REALLY GOOD GAME OF ALIEN INVADERS WITH THE HIGHEST SCORE YOU'VE EVER HAD -- AND THEN HIT THE BREAK KEY - THIS IS FOR YOU! DISABLING THE BREAK KEY AND MAKING IT A SWITCH OPERATED OPTION WILL GIVE YOU THE SAME PROTECTION AGAINST UNINTENTIONALLY WIPING YOUR PROGRAM THAT THE BIG MACHINES - THE C8'S AND UP - HAVE. (ON THE C8, THE BREAK KEY IS SAFELY SITUATED ON THE CPU UNIT - 18 INCHES OR SO AWAY FROM FLYING FINGERS)

THE PARTS LIST IS SHORT - A SINGLE POLE SINGLE THROW SWITCH AND ENOUGH WIRE TO REACH FROM THE BREAK KEY TO WHERE THE SWITCH WILL BE INSTALLED. RADIO SHACK SELLS A LARGE BASE SWITCH WHICH WILL FIT INTO THE HOLES IN THE BACK OF YOUR CASE WITHOUT DRILLING. ON THE C1, THE FOIL RUN FROM THE BREAK KEY TO THE CPU RUNS KITTYCORNER UNDER THE RUB OUT KEY TO A FEED THROUGH HOLE JUST ABOVE THE (=) KEY. CUT THE FOIL ANYWHERE HANDY ALONG THIS LINE. SOLDER ONE PIECE OF WIRE TO THE POST OF THE BREAK KEY TOWARD THE TOP OF THE KEYBOARD. THE OTHER END OF THIS WIRE GOES TO ONE SIDE OF THE SWITCH. SOLDER THE OTHER WIRE INTO THE FEED THROUGH HOLE AT ONE END AND THE OTHER SIDE OF THE SWITCH AT THE OTHER END. INSTALL THE SWITCH IN ONE OF THE HOLES IN THE BACK OF THE CASE AND YOU'RE DONE. TURNING THE SWITCH TO OFF DISCONNECTS THE BREAK KEY FROM THE CPU. IF YOU NEED TO BREAK, FLIP THE SWITCH AND HIT THE KEY.

ON THE C2/4, THE BREAK KEY LINES COME DIRECTLY OFF THE KEYBOARD JUST TO THE LEFT OF THE (1) KEY. CUT THE RED WIRE AND SPLICE IN JUMPERS TO THE SWITCH. INSTALL THE SWITCH AND THE MODIFICATION IS COMPLETE.

ALTERNATELY, YOU CAN USE A PUSH BUTTON SWITCH ACROSS THE TWO WIRES ON THE C2/4 OR FROM THE FOIL CUT TO GROUND ON THE C1 AND REPLACE THE BREAK KEY ALTOGETHER.

(FOR THOSE WHO GET WOOSY AT THE THOUGHT OF TACKLING THEIR COMPUTER WITH A SOLDERING IRON, THERE ARE OTHER WAYS AROUND THE PROBLEM. GERRY MELE OF BRUNSWICK, OHIO, CAME UP WITH A "Z" SHAPED PIECE OF FLATTENED BRASS TUBING WHICH HE EPOXIED ONTO THE CAS SO THE KEY WAS SHEILDED. HE SAYS TO USE A THICK, NON-RUNNY EPOXY AND BE CAREFUL NOT LET ANY DRIP INTO THE KEYS. (IT WOULD PROBABLY BE BETTER TO FIT THE SHIELD, THEN REMOVE THE CASE COMPLETELY FROM THE PC BOARD TO DO THE GLUING.) EVEN SIMPLIER, THO' NOT AS CHIC, IS A SIMPLE SHEILD MADE FROM HALF A FILE CARD AND TAPED INTO PLACE. (SEE ACCOMPANYING SKETCH) REMEMBER TO LEAVE ENOUGH ROOM BETWEEN ANY SHIELD AND THE KEY TO GET YOUR FINGER IN WHEN YOU WANT TO USE THE KEY. THE IDEA IS TO KEEP FROM HITTING IT ACCIDENTALLY, NOT TO RENDER IT INOPERABLE. J.O.)

)*** BEGINNERS CORNER ***

LETS DISCUSS INPUT STATEMENTS THIS MONTH. OSI'S MANUAL DOES A DECENT JOB OF SHOWING THE SYNTAX FOR A SIMPLE INPUT. IT IS SIMPLY:

```
100 INPUTA:    REM FOR VARIABLE VALUES
OR
100INPUTA$:    REM FOR STRING VARIABLES
```

THERE ARE TWO THINGS YOU NEED TO REMEMBER, HOWVER, FOR BETTER PROGRAMMING. THE FIRST AND MOST IMPORTANT IS THAT OSI'S BASIC ALLOWS YOU TO PRINT A PROMPT AS PART OF THE INPUT STATEMENT. I.E.

```
100INPUT"YOUR NAME";NAME$
OR
100INPUT"HOW OLD ARE YOU";YEARS
```

THE SECOND THING TO REMEMBER IS THAT YOU ARE ALLOWED TO DO SEVERAL INPUTS WITH ONE STATEMENT. I.E.

```
100INPUT"GIVE ME A ROW AND COLUMN";ROW,COLUMN:  REM WILL DEMAND TWO INPUTS DIVIDED BY COMMAS.
```

UNFORTUNATELY, YOU CANNOT COMBINE THE TWO REATURES AND REPEAT PROMPTS IN THE SAME STATEMENT

```
100INPUT"YEAR";Y"MONTH";M:     REM DOES NOT REPEAT *NOT* WORK!!
```

)** A SIMPLE PROCESSOR **

ACTUALLY THE POINT OF THE ARTICLE IS NOT TO GIVE YOU A WORD PROCESSOR, BUT TO ENCOURAGE YOU TO WRITE ONE WITH WHAT YOU LEARNED IN THE LAST ISSUE. I ALSO WANT TO POINT OUT THAT USEABLE PROGRAMS DO NOT NECESSARILY HAVE TO BE BIG AND FANCY PROGRAMS. THIS LETTER WRITER WAS UP AND RUNNING A FEW HOURS AFTER I GOT THE PAPER TIGER. WHILE IT IS NOT FANCY, IT DID ALLOW ME TO WRITE LETTERS WITH A HEADING PRINTED BY THE PRINTER, ALLOWED ME TO DO LIMITED EDITING BY REPLACING, DELETING, AND INSERTING LINES, AND ALLOWED ME TO STORE LETTERS ON DISK FOR FUTURE EDITING.

FOR DISK USERS, TYPE THIS IN AND CALL IT BEXEC*. IT WILL BOOT UP THE DISK, AND SET UP THE BASIC FOR WORD PROCESSING. PLACE ON THE DISK AS MANY FILES AS YOU CAN FIT THAT ARE CALLED 'FILE1','FILE2', AND SO ON. THE PROGRAM LOOKS FOR A FILE CALLED 'FILE' PLUS ANY NUMBER. THE NUMBER OF FILES YOU SET UP IS THE NUMBER OF MEMORIES YOU HAVE. I SUGGEST ABOUT 2 TRACKS PER FILE.

FOR TAPE USERS-

YOU WILL NEED AT LEAST 8K OF RAM AND YOU WILL NEED TO CORRECT THE GARBAGE COLLECTION PROBLEM. I'LL INCLUDE A SHORT PROGRAM THAT WILL CORRECT THE ROUTINE ON 8K MACHINES.

YOU WILL ALSO HAVE TO MODIFY THE SAVE AND GET SECTIONS FOR TAPE RATHER THAN DISK STORAGE. BEGIN BY DELETING LINES 500, 530, 550, 670 AND 680. MAKE LINE 510 INTO LINE 500. ADD--

```
530INPUT"IS  TAPE  READY";X$:SAVE:FORX=1TO1500:NEXT
550LOAD:GOTO90:REM  HIT  SPACE  TO  GET  KEYBOARD  BACK.
532FORX=1TOL:IFA$(X)="  "THEN?"#":GOTO537
535?A$(X)
537 NEXT:REM  WE  HAD  TO  REPLACE  BLANK  LINES  TO  AVOID  INPUT  ERRORS
670INPUT"IS  TAPE  RUNNING";X$:LOAD
695IF  A$="#"THENA$="  ":REM RESTORING BLANK LINES REM: HIT SPACE TO GET KEYBOARD CONTROL BACK.
```

GET RID OFF ALL REMS - YOU ARE SHORT ON SPACE.

THERE IS A LOT OF ROOM FOR IMPROVEMENT- WHICH YOU SHOULD ENJOY. ON THE LETTER WRITER THAT WE SELL (THE ONE I WROTE THIS ARTICLE ON) I ADDED SELECTABLE MARGINS AND WIDTH, AUTOMATIC CENTERING OF HEADINGS, AUTOMATIC RIGHT MARGIN JUSTIFICATION, SELECTION OF MANUSCRIPT OR LETTER FORMAT, BLOCK DELETIONS, AND FILE MERGING CAPABILITIES. I ADDED THEM ALL ONE AT A TIME AND IT WAS RATHER PAINLESS. BY NOW YOU SHOULD BE ABLE TO DO THE SAME

GARBAGE COLLECTOR CORRECTION

RUN THIS BEFORE YOU LOAD THE LETTER WRITER. IT WILL ERASE ITSELF
AND LEAVE A MACHINE CODE ROUTINE BEHIND

```
 5 REM COPYRIGHT 1980-AARDVARK ILLEGAL TO COPY
 10 POKE133,250:POKE134,30:POKE11,250:POKE12,30:L=7930:K=L
 20 FORI=45383TO45644:X=PEEK(I):POKEL,X:L=L+1:NEXT
 30 FORT1=1TO13:READX,Y:POKEX,Y:NEXT
 40 DATA7964,31,7963,140,7989,31,7988,134,8014,31,8013,203,8067,31,8066
 50 DATA140,8191,30,8190,254,7997,4,8146,2,8147,24
 60 NEW
```

```
 10 P=PEEK(10950):POKE8994,P:POKE8993,P:FORX=1TO32:PRINT:NEXT
 20 FORX=1TO64:PRINT"=";:NEXT:PRINT:PRINT
 30 PRINT:PRINT:PRINTTAB(15)"AARVARK LETTER WRITER":PRINT:PRINT
 40 FORX=1TO64:PRINT"=";:NEXT:PRINT:POKE2976,13:POKE2972,13
 45 REM TAPE USERS DELETE POKES IN LINE40
 50 DIMA$(150):PRINT"COPYRIGHT 1980":PRINT:PRINT
 55 REM DIM TO FIT YOU MEMORY SIZE
 60 PRINT:PRINT:L=1
 70 INPUT"WHO ARE YOU";N$:IFN$="BOSS"THEN850
 80 INPUT"DATE";D$
 90 INPUTA$:IFLEN(A$)<1THENA$(L)=A$:L=L+1:GOTO90
 100 IFASC(A$)=94THENGOSUB120:GOTO90
 110 A$(L)=A$:L=L+1:GOTO90
 120 A=ASC(MID$(A$,2,1)):IFA=42THENA=ASC(MID$(A$,3,1))
 140 IFA=76THEN280:LIST
 150 IFA=71THEN670:GET MEMORY
```

```
160 IFA=80THEN320:PRINT
170 IFA=83THEN500:SAVE
180 IFA=82THEN560:REPLACE
190 IFA=72THEN770:HELP
200 IFA=68THEN600:DELETE
210 IFA=78THEN750:NEW LETTER
220 IFA=73THEN630:INSERT
260 GOTO90
280 INPUT"FROM";X$:IFASC(X$)=65THENX=1:Y=L:GOTO300
290 X=VAL(X$):INPUT"TO";Y
300 INPUT"HARD COPY";X$:IFASC(X$)=89THENDISK!"IO ,03"
310 FORI=XTOY:PRINTI;A$(I):NEXT:PRINTCHR$(12):DISK!"IO ,02":GOTO90
315 REM CHR$(12)= PAPER TIGER PAGING COMMAND
320 INPUT"HARD COPY";A$:IFASC(A$)<>89THENPRINT:GOTO380
330 DISK!"IO ,03":PRINTCHR$(01);
350 REM -PUT YOU HEADING AND NAME IN LINES 380 AND 400
360 REM (01) AND (02) ARE DOUBLE WIDTH CHARACTER CODES ON PAPER TIGER.
380 PRINTTAB(14)"AARDVARK TECHNICAL SERVICES":PRINTCHR$(02)
400 PRINTTAB(25)"1690 BOLTON WALLED LAKE MICH. 48088 (313)624-6316
410 PRINT:FORX=1TO98:PRINT"=";:NEXT
420 REM NEXT LINE CENTERS TEXT -L= #LINES OF TEXT.
430 IFL<50THENFORX=1TO(40-L)/2:PRINT:NEXT
440 PRINTD$:PRINT:PRINT
470 FORX=1TOL:PRINTA$(X):NEXT:PRINT:PRINT
475 PRINTN$:PRINT
480 PRINTCHR$(12):DISK!"IO ,02":GOTO90
490 REM TAPE USERS SEE NOTES. LINES 500-590 ARE NOT FOR YOU.
500 INPUT"WHICH MEMORY (1-6)";X$:X$="FILE"+X$
510 INPUT"FROM LINE";A:INPUT"TO";A$:I=VAL(A$):IFI>LTHENPRINT"TO BIG":
520 IFASC(A$)=65THENI=L
530 DISK OPEN,6,X$:FORX=ATOI:PRINT#6,A$(X):NEXT
540 PRINT#6,"****":PRINT#6,"****"
550 DISK PUT:DISK CLOSE,6:GOTO90
560 INPUT"NUMBER";:IFI>LTHEN90
570 PRINT"REPLACE-"
580 PRINT" "A$(I):INPUTA$:IFA$<>"NO"THENA$(I)=A$
590 GOTO90
600 INPUT"DELETE";X:IFX>LTHEN90
610 PRINTA$(X):INPUT"O.K.";A$:IFASC(A$)<>89THEN90
620 L=L-1:FORY=XTOL:A$(Y)=A$(Y+1):NEXT:GOTO90
630 INPUT"AFTER LINE";X:IFX>LTHENGOTO90
640 INPUTA$:IFA$="NO"THEN90
645 REM "NO" MEANS YOU CHANGED YOUR MIND.

650 L=L+1:FORY=LTOX+2STEP-1:A$(Y)=A$(Y-1):NEXT
660 A$(X+1)=A$:RETURN
670 INPUT"WHICH MEM";X$:A$="FILE"+X$
680 DISK OPEN,6,A$
690 INPUT#6,A$:IFA$="****"THEN90
700 A$(L)=A$:L=L+1:PRINTA$:GOTO690
750 FORX=0TOL+1:A$(X)="":NEXT:PRINT:PRINT:L=1:PRINT"NEW LETTER":PRINT
760 GOTO90
770 PRINT"↑(SHIFT N) FOLLOWED BY A LETTER ENTER THE EDIT MODE
780 PRINT"L=LIST    P=PRINT    I=INSERT   R=REPLACE
790 PRINT"S=SAVE (10 MEMORIES)   G=GET FROM MEMORY
810 PRINT"D=DELETE   D,L,I, AND P ASK FOR LINE NUMBERS
820 PRINT"A OR ALL WILL GET ALL LINES LISTED ON L COMMAND
840 GOTO90
845 REM TAPE USERS DELETE AFTER 840
850 POKE741,76:POKE750,78:POKE2073,173:POKE2893,55:POKE2894,8
860 PRINT:PRINT:PRINT"SYSTEM OPEN, BOSS":PRINT:PRINT
```

```
 10 FORK=53248TO54272:POKEK,32
:NEXT
 11 PRINT"       SKEET!!!":PRIN
T"BY MIKE BASSMAN":FORK=1TO10:
PRINT:NEXT
 12 INPUT"DIFFICULTY";Z:FORK=1
TO30:PRINT:NEXT:IFZ<1ORZ>5THEN
11
 20 FORK=53412TO53435:POKEK,18
7:POKEK+23*32,187:NEXT
 30 FORK=53412TO54148STEP32:PO
KEK,187:POKEK+23,187:NEXT:DIML
T(Z)
 50 FORK=1TOZ:LT(K)=INT(RND(1)
*20+53605):NEXTK:POKE54117,32:
BL=Z*4
 80 LG=54095:POKELG,29:POKE530
,1:P=57088:POKEP,127
 100 IFPEEK(P)<>127THEN140
 110 IFPEEK(LG-1)<>32THEN500
 115 POKELG,32
 120 LG=LG-1:POKELG,29:GOTO500
 140 IFPEEK(P)<>191THEN160
 150 IFPEEK(LG+1)<>32THEN500
 155 POKELG,32:LG=LG+2:GOTO120
 160 IFPEEK(P)<>223THEN500
 165 IFBL=0THEN1100
 167 BL=BL-1:GOSUB300:LB=LG
 175 LB=LB-32:IFPEEK(LB)<>32TH
EN210
 190 POKELB,149:POKELB+32,32:P
OKELG,29:GOTO175
 210 FC=PEEK(LB):POKELB+32,32:
FORK=1TO10
 230 POKELB,232:FORG=1TO10:NEX
TG:POKELB,233:FORG=1TO10:NEXTG
:NEXTK:POKELB,32
 250 IFFC=187THENPOKELB,187
 260 IFFC=4THENS=S+1
 270 IFS=Z*2THEN1000
 280 POKE53382,ASC(MID$(STR$(S
),2,1)):GOTO500
 300 SL=53399
 310 FORK=1TOLEN(STR$(BL)):POK
ESL,ASC(MID$(STR$(BL),K,1)):SL
=SL+1:NEXTK
 330 POKESL,32:RETURN
 500 FORK=1TOZ
 510 D=INT(RND(1)*9+1)
 520 ONDGOTO530,540,550,560,57
0,580,590,600,610
 530 D=1:GOTO620
 540 D=-1:GOTO620
 550 D=33:GOTO620
 560 D=-33:GOTO620
 570 D=32:GOTO620
 580 D=-32:GOTO620
 590 D=31:GOTO620
 600 D=-31:GOTO620
 610 D=0:POKELT(K),32
 620 IFPEEK(LT(K)+D)<>32THEN51
0
 625 POKELT(K),32:LT(K)=LT(K)+
D:POKELT(K),4:NEXTK:GOTO100
 1000 FORK=53248TO54272:POKEK,
32:NEXT:PRINT"YOU WIN!!!":GOTO
1110
 1100 FORK=53248TO54271:POKEK,
32:NEXT:PRINT"YOU LOSE !!"
 1110 FORK=1TO10:PRINT:NEXT:IN
PUT"TRY AGAIN";Y$:IFLEFT$(Y$,1
)="Y"THENRUN
 1140 POKE530,0
```

```
 2 REM POTATO CHIP INVASION
 3 REM CONTROLS ARE KEYS 1,2 AND7
 5 POKE605,0:FORK=611TO625:POKEK,32
:NEXTK:POKEK,255
 10 DATA546,604
 20 DATA174,59,211,24,173,35,2,105,
32,141,55,2,173,36,2,105,0
 30 DATA141,56,2,142,195,208,24,173
,35,2,201,195,208,8,173,36,2
 40 DATA201,208,208,1,96,56,173,35,
2,233,1,141,35,2,173,36,2
 50 DATA233,0,141,36,2,76,34,2
 60 FORX=546TO571:I=PEEK(X+64490):P
OKEX,I:NEXT:POKE572,96
 70 POKE11,34:POKE12,2:X=USR(X):PRI
NT"     Potato Chips"
 90 PRINT"     -------------":FORK=1
TO10:PRINT:NEXT
 110 PRINT"Hit return to start game
";:POKE54117,32
 120 POKE11,0:POKE12,253:X=USR(X):P
OKE11,34:POKE12,2:X=USR(X):K=611
 123 IFPEEK(K)=255THEN130
 124 X=PEEK(K):POKE53413+K-611,X:K=
K+1:GOTO123
 130 READA,B:FORK=ATOB:READX:POKEK,
X:NEXT
 135 READA,B:FORK=ATOB:READX:POKEK,
X:NEXT
 140 POKE530,1:POKE57088,127:GL=541
27:F=2
 150 G1(1)=146:G1(2)=149:G1(3)=147:
POKEGL,G1(F):P=57088:T=9
 190 FORZ=1TO10:IFPEEK(P)<>127THEN2
50
 210 IFGL=54117ANDF=1THEN1000
 220 POKEGL,32:F=F-1:IFF=0THENF=3:G
L=GL-1
 230 POKEGL,G1(F):GOTO1000
 250 IFPEEK(P)<>191THEN300
 260 IFGL=54139ANDF=3THEN1000
 270 POKEGL,32:F=F+1:IFF=4THENF=1:G
L=GL+1
 280 POKEGL,G1(F):GOTO1000
 300 IFPEEK(P)<>253THEN1000
 305 BL=GL-64:Z=Z+3
 310 IFPEEK(BL)=32THEN400
 315 POKEBL+32,32:IFPEEK(BL)=235THE
NBL=BL-1
 330 FORK=99TO32STEP-1:POKEBL,K:POK
EBL+1,K:NEXTK:S=S+1:NG=NG-1:GOTO100
0
 400 POKEBL,G1(F):POKEBL+32,32:BL=B
L-32:IFBL>53443THEN310
 410 POKEBL+32,32
 1000 NEXTZ:T=T+1:X=USR(X):TG=INT(T
/10):IFNG>TGTHEN1050
 1010 X=INT(RND(1)*21+1)+53477:POKE
X,234:POKEX+1,235:NG=NG+1
 1050 POKE11,0:POKE12,28:X=USR(X):I
FPEEK(610)=0THEN190
 1070 FORK=255TO32STEP-1:POKEGL,K:N
EXTK:FORK=1TO35:PRINT:NEXT
 1090 PRINT"Your final score was";S
 1095 PRINT:PRINT:PRINT"The high sc
ore is";PEEK(605):PRINT:PRINT
 1097 IFS>PEEK(605)THEN1150
 1100 PRINT:PRINT:INPUT"try again";
Y$
 1110 IFLEFT$(Y$,1)="Y"THENRESTORE:
CLEAR:GOTO10
 1120 END
 1150 PRINT"You have beaten it.":PR
INT:PRINT:POKE605,S:INPUT"Your name
";S$
 1170 FORK=611TO611+LEN(S$)-1:POKEK
,ASC(MID$(S$,K-610,1)):NEXTK
 1171 POKEK,32:POKEK+1,32
 1172 S$=STR$(S):FORG=2TOLEN(S$):PO
KEK+G,ASC(MID$(S$,G,1)):NEXTG:POKEK
+G,255
 1180 PRINT:PRINT:GOTO1100
 1500 DATA7168,7223
 1510 DATA160,35,162,211,140,12,28,
142,13,28,24,173,35,211,201,234,240
 1520 DATA9,24,192,59,240,9,200,76,
4,28,169,255,76,34,28,169,0
 1530 DATA141,98,2,169,2,133,12,169
,34,133,11,169,59,141,35,2
 1540 DATA169,211,141,36,2,96
OK
```

```
 5 GOSUB500:NW=50:L1=53524:FORK=532
48TO54272:POKEK,32:NEXT:DIMA(30)
 20 FORK=53928TO53932:POKEK,161:NEX
T
 30 FORK=53514TO53898STEP32:POKEK,1
61:NEXT
 40 FORK=53515TO53520:POKEK,161:NEX
T:X=RND(X):X=INT(RND(X)*NW+1)
 70 FORK=1TOX:READW$:NEXTK:FORK=1TO
LEN(W$):POKE54055+K,95:NEXTK
 90 POKE11,0:POKE12,253
 100 X=USR(X):C=PEEK(531):IFC<65THE
N100
 120 FORK=1TOLEN(W$):IFASC(MID$(W$,
K,1))=CTHEN400
 140 NEXTK:GOTO600
 142 NR=NR+1:L=L+2:IFL=12THENL=0:L1
=L1+34
 147 POKEL1+L-2,C
 150 ONNRGOSUB160,200,240,280,310,3
30,350,370,380,390
 155 IFNR=10THEN5000
 157 GOTO100
 160 POKE53550,176:POKE53551,161:PO
KE53552,178
 170 POKE53582,177:POKE53583,161:PO
KE53584,175:RETURN
 200 POKE53614,210:POKE53616,207:PO
KE53710,209:POKE53712,208
 210 POKE53615,135:POKE53711,128
 220 FORK=53646TO53678STEP32:POKEK,
136:POKEK+2,143:NEXTK:RETURN
 240 POKE53742,187:POKE53774,187:RE
TURN
 280 POKE53744,187:POKE53776,187:RE
TURN
 310 POKE53644,150:POKE53645,150:RE
TURN
 330 POKE53649,151:POKE53650,151:RE
TURN
 350 POKE53773,150:RETURN
 370 POKE53777,150:RETURN
 380 POKE53676,168:RETURN
 390 POKE53618,165:RETURN
 400 FORK=1TOLEN(W$)
 410 IFASC(MID$(W$,K,1))=CTHENGOSUB
450:POKE54023+K,C
 420 NEXTK:IFNC=LEN(W$)THEN5200
 440 GOTO100
 450 IFPEEK(54023+K)=32THENNC=NC+1
 460 RETURN
 500 FORK=53248TO54272:POKEK,32:NEX
T:PRINT"        HANGMAN"
```

```
 520 PRINT"        -------":FORK=1TO
10:PRINT:NEXT:FORK=1TO2000:NEXT:RET
URN
 600 FORG=53524TO53524+34STEP34:FOR
R=2TO10STEP2:IFPEEK(G+R-2)=CTHENF1=
1
 630 NEXTR:NEXTG:IFF1=1THENF1=0:GOT
O100
 650 GOTO142
 1000 DATA"HELLO":DATAGYPSY,HIEROGL
YPHIC,CRUCIBLE,OFTEN,COMPUTER
 1020 DATAINCREDIBLE,CONVERTER,PROG
RAM,BICYCLE,LIGHTENING
 1030 DATARADIO,DIFFICULT,CASSETTE,
CREATIVE,BIAS,OPTIMIST
 1040 DATAPESSIMIST,INSTRUMENT,WALL
ET,SHUTTLE,MONEY,FRAME
 1050 DATAGARBAGE,PAPER,BAGGAGE,AIR
PLANE,ENGINE,RECORD
 1060 DATACUBICLE,PSYCHOPATH,PORTRA
IT,DECIMAL,DICTIONARY
 1070 DATAPHOTOGRAPH,FLUORESCENT,MA
GAZINE,PORTABLE,CARTON
 1080 DATADELIGHT,AUTOMOBILE,CLOSET
,VOLCANO,INEDIBLE
 1090 DATASLEEVE,CABINET,DUST,XYLOP
HONE,CELLO,PURPLE
 5000 D$="YOU LOSE THIS TIME!":GOSU
B5010:GOTO5020
 5010 FORK=1TOLEN(D$):POKE53444+K,A
SC(MID$(D$,K,1)):NEXTK:RETURN
 5020 FORK=1TO3000:NEXT:D$="THE WOR
D WAS                ":GOSUB5010
 5040 FORK=1TOLEN(W$):POKE54023+K,A
SC(MID$(W$,K,1)):NEXTK
 5060 FORK=1TO3000:NEXT:GOTO5210
 5200 D$="YOU WIN THIS TIME!":GOSUB
5010:FORK=1TO3000:NEXTK
 5210 INPUT"TRY AGAIN";Y$:IFLEFT$(Y
$,1)="Y"THENCLEAR:GOTO5
 5220 FORK=53248TO54272:POKEK,32:NE
XT:PRINT"BYE"
```

)**SUBMITTING PROFESSIONAL PROGRAMS**

THESE COMMENTS DO NOT APPLY TO JOURNAL SUBMISSIONS.

I HAVE SOME SUGGESTIONS TO MAKE TO EVERYONE WHO PLANS TO SUBMIT PROGRAMS EITHER TO AARDVARK OR TO ANYONE ELSE FOR PROFESSIONAL PUBLICATION.

FIRST, MAKE SURE THAT WE DONT PUBLISH THE PROGRAM ALREADY. I GET ABOUT ONE DOODLER PROGRAM EVERY THREE DAYS. WE ALREADY PUBLISH ONE AND WOULD NOT CHANGE UNLESS THE NEW ONE WERE VERY, VERY GOOD. THEREFORE, BEFORE YOU SUBMIT A PROGRAM TO ANY SOFTWARE HOUSE, GET THEIR CATALOG AND MAKE CERTAIN THAT YOU AREN'T DUPLICATING EXISTING PROGRAMS. IF YOU ARE FIND THAT YOU ARE DUPLICATING A PROGRAM BUT STILL FEEL THAT YOURS MIGHT BE A SIGNIFIGANT IMPROVEMENT, INVEST IN A COPY OF THE ALREADY PUBLISHED PROGRAM AND HAVE SOMEONE YOU HATE DO AN HONEST EVALUTATION FOR YOU. REMEMBER THAT YOUR PROGRAM WILL HAVE TO BE A LOT BETTER THAN THE EXISTING ONE TO GET PUBLISHED. ANY GOOD SOFTWARE HOUSE HAS CONSIDERABLE TIME, EFFORT, AND MONEY TIED UP IN DEBUGGING, DOCUMENTING AND PUBLISHING THE EXISTING PROGRAM AND THEY ARE NOT GOING TO DUMP THAT WITHOUT A VERY GOOD REASON.

I ALSO SUGGEST THAT YOU DO NOT SEND TO ANY SOFTWARE HOUSE, PARTICULARLY US, ANY COPIES OF HANGMAN, 3 D TICK TACK TOE, FUNCTION PLOTTERS, DOODLERS, OR TWO MAN AIRPLANE COMBAT. I GOT BOXES OF THEM AND I AM SURE EVERYONE ELSE DOES ALSO.

ALRIGHT, WHAT IS A SOFTWARE HOUSE LOOKING FOR. WELL, THE VIDEO GAME IS NOT DEAD, BUT THE REQUIREMENTS ARE WAY UP. BOB RETELLE AND I AND SCHNIEDER HAVE DONE THE SIMPLER ONES VERY WELL-WE'VE HAD A LOT OF TIME AND EXPERIENCE. WE HAVE HAD TO TAKE OFF THE MARKET SEVERAL GAMES THAT WE SOLD A YEAR AGO BECAUSE THE STANDARDS OF TODAYS PROGRAMS ARE MUCH HIGHER THAN THEY USED TO BE. TO BE PUBLISHED, A PROGRAM NEEDS A NEW IDEA AND EXECELLENT GRAPHICS. TO BE REALLY SUCCESSFULL IN TODAYS MARKET, IT SHOULD BE SOMETHING ON THE ORDER OF BALLY'S NEW GALACTIC RAIDERS PROGRAM. SIMPLER,BUT SIMILAR. SIMPLER PROGRAMS CAN ALSO BE SUCCESSFULL, BUT REQUIRE AT LEAST A NEW IDEA FOR A GAME. NON-VIDEO GAMES CAN ALSO BE SUCCESSFULL AND ARE DESIRED BY SOFTWARE HOUSES. AARDVARKS NEWEST PROGRAMS ARE ADVENTURE PROGRAMS. THEY REQUIRE SEVERAL HOURS TO PLAY AND ARE OF RATHER HIGH SOPHISTICATION. THE MAIN THING THAT MOST AUTHORS MISS IN WRITING GAMES OF THE SORT IS THAT GUESS A RANDOM NUMBER GAMES ARE NOT PUBLISHABLE NO MATTER WHAT YOU CALL THEM. WE PUBLISH OVER 50 GAMES, AND ALL OF THEM REQUIRE SOME SKILL OR THOUGHT, AND HOPEFULLY BOTH. WE STILL GET SEVERAL PROGRAMS A WEEK THAT ARE CALLED BASEBALL OR FOOTBALL OR HANARABI THAT ARE REALLY JUST A MATTER OF GUESSING RANDOM NUMBERS.

ALONG THE SAME LINE, THE GAME MUST GIVE THE COMPUTER A UNIQUE FUNCTION. A BACKGAMMON OR CHECKER GAME THAT JUST USES THE COMPUTER TERMINAL FOR A BOARD FOR TWO PLAYERS IS A LOT LESS FUN THAN A REAL BOARD AND REAL DICE. IF YOU ARE GOING TO DO A BOARD GAME, MAKE SURE THAT IT CANNOT BE PLAYED ON A REAL BOARD EASIER.

THE BIGGEST DEMAND NOW IS FOR UTILITIES AND FOR SIMPLE BUSINESS PROGRAMS. AS SHNIEDER SAYS, 'PROGRAMS SHOULD WRITE PROGRAMS- WE SHOULD HAVE FUN'. PROGRAMS LIKE THE PACKER, RENUMBERER, AND EDITORS HAVE A WIDE POPULARITY AMONG BOTH BUSINESS AND PERSONAL USERS AND ANY NEW ONES WILL BE LOOKED AT CAREFULLY BY A SOFTWARE HOUSE.

BUSINESS SOFTWARE IS A WIDE OPEN FIELD THAT MOST SOFTWARE HOUSES ARE LOUSING UP. MOST OF THEM WANT TO PUBLISH DOUBLE ENTRY, GENERATE 15 REPORTS, AND CONFOUND THE BOOKKEEPER SOFTWARE THAT WILL SELL FOR $250- $500. WITH DISK MACHINES DOWN TO $1000 AND TAPE MACHINES DOWN TO $300, THESE MACHINES ARE INCREASINGLY BEING PURCHASED BY SMALL BUSINESSMEN WHO DON'T NEED SUCH FANCY OR EXPENSIVE STUFF. BOOKKEEPING RATHER THAN ACCOUNTING PROGRAMS THAT SELL FOR $50 RATHER THAN $500 ARE GOING TO BE GOOD SELLERS IN THE FUTURE AND WILL BE LOOKED AT BY AARDVARK AND ANY OTHER SOFTWARE HOUSE WITH SENSE.

I FREQUENTLY TALK TO CUSTOMERS WHO HAVE DEVELOPED SOFTWARE TO USE IN THEIR OWN SMALL BUSINESSES. SUCH PROGRAMS OFTEN CANNOT BE PROFITABLY MARKETTED ALONE BUT CAN BE ADDED TO AN EXISTING LINE PROFITABLY. I ALWAYS ENCOURAGE THOSE WHO HAVE DEVELOPED SOFTWARE FOR THEIR OWN USE TO TURN IT OVER TO AN EXPERT AND ATTEMPT TO RECOVER THE COST OF DEVELOPEMENT.

WE ARE LOOKING FOR, AND WILL PROBABLY DEVELOPE OURSELVES, SMALL PAYROLL AND INVENTORY PROGRAMS.

ONE LAST NOTE ONE GAMES. ONE FIELD HAS NOT BEEN TAPPED. GOOD SPORTS GAMES ARE NOT AVAILABLE FOR OSI. A FOOTBALL GAME THAT WAS NOT A SERIES OF RANDOM NUMBERS WOULD PROBABLY GET GOOD REVIEWS FROM A SOFTWARE HOUSE, AS WOULD GOOD GOLF AND BASEBALL GAMES. OSI PUT OUT THEIR ONLY GOOD DISK SOFTWARE ON GAMES DISK 5 WHICH IS SPORTS GAMES, BUT HAVE NOT MADE ANYTHING AVAILABLE FOR THE MORE POPULAR C1 MACHINES.

)**** NEW FROM AARDVARK ****

WE DO STAY BUSY. BIG NEWS THIS MONTH IS ADVENTURE FOR THE OSI. IF YOU HAVE NEVER PLAYED ADVENTURE, YOU'VE MISSED SOMETHING. IT IS BEST DESCRIBED AS 'INTERACTIVE FANTASY'- LIKE READING A BOOK, BUT YOU ARE ONE OF THE CHARACTERS. YOU TYPE IN ENGLISH COMMANS TO THE COMPUTER THAT ALLOW YOU TO MOVE AROUND AND EXPLORE THE ENVIRONMENT. YOU CAN ALSO PICK UP, DROP, SHOOT AND DO OTHER THINGS TO THE STUFF YOU FIND. THE WHOLE THING USUALLY TAKES A COUPLE OF DAYS TO PLAY.

I WROTE 'DEATH SHIP' AND 'ESCAPE FROM MARS' AND THEN BOB RETELLE WROTE 'TREK ADVENTURE'. THEY FIT IN 8K AND SELL FOR $14.95 ON TAPE OR 5' DISK AND $15.95 ON 8' DISK. .

WE HAVEN'T FORGOTTEN TO DO PRACTICAL STUFF TOO. WE HAVE ADDED 'LETTER WRITER' TO OUR DISK LINE. AS YOU CAN TELL FROM THE JOURNAL, I AM RATHER ENTHUSIASTIC ABOUT IT. IT SELLS ON DISK FOR $12.95 (I'M GOING TO RAISE THAT PRICE AS SOON AS WE GET THE CATALOGS REPRINTED).

WE ALSO HAVE A WORD PROCESSOR - MINIPROS - IN BASIC - NOT DESIGNED FOR BIG OFFICES- WHICH SELLS FOR $26.95 ON DISK. IT HAS FULL EDIT FEATURES PLUS ALL THE STULL IN THE LETTER WRITER.

OUR NEW INTELLIGENT TERMINAL PROGRAM IS A DOOSEY. IT ALLOWS YOU TO DOWNLOAD PROGRAMS FROM A MAINFRAME, EDIT WITH ABOUT 22 EDIT COMMANDS, RUN THE PROGRAM LOCALLY, OR SEND IT BACK TO THE MAINFRAME FOR EXECUTION. IT WILL BE AVAILABLE AS SOON AS THE LOCAL DEALER RETURNS MY C8. IT WILL SELL FOR $59.95 MAKING IT OUR MOST EXPENSIVE PROGRAMS AND ONE OF OUR BEST BARGAINS - IT IS REAL BIG SYSTEM TYPE SOFTWARE.

WE ALSO HAVE NEW DATA SHEETS-
IMPLEMENTING THE SECRET SOUND PORT ON THE C1P ($4.00)
BUILDING A LIGHT PEN FOR YOUR OSI ($4.00)
C1 SOUND BOARD PLANS - USING THE TI CHIP FOR REAL ARCADE STYLE SOUND ($5.00)

AARDVARK
TECHNICAL SERVICES
1690 Bolton, Walled Lake, MI 48088

FIRST CLASS