ON/OFF SWITCH as a possible source of computer problems has been identified by a subscriber. The spring-loaded contacts may corrode and have momentary failures. The real solution is to solder across the contacts and disconnect the unit at the wall plug. You should be aware that there is always current in the unit as long as the wall transformer is connected. The switch only shuts off part of the continuity.

CONTROLLER PROBLEMS. A number of subscribers have indicated problems with the hand controllers, especially in slots 3 and 4. If it is a problem only with the ports, then probably the I/O chip has a failure. Controller problems can usually be helped by cleaning. We have not been able to locate any controller source other than the list price outlets such as M.Wards.

CLUB NEWS The Chicago Area Group (p.1 for address) is starting a software library for their members.

RF MODULATOR is a gadget needed to convert the output of the Bally into a form usable by the TV set. One is included with your machine, it sits in the lower left corner and the shielded line going to your TV set originates there. Those of you who have purchased spare boards need one to get a visual display of what's going on inside the machine. Those I had last year are all gone, but I see the latest Poly Paks catalog has them for 17.95, part no. 92CU6527. The illustration appears to be identical to the Bally unit.

BALCHEK There is a very small interest by subscribers for the Balcheck tool, which requires someone to burn the program into a 2716 chip. If anyone is thinking of doing this, let me know.

GAME ENHANCEMENTS Virtually every program listed can be improved or modified. One designer has indicated that, for him, once the game has been made to operate to a sufficient degree of difficulty, the interest in further programming is lost and he begins to look for a new programming challenge. Now is the time for the game players to take over and revise the game to suit themselves.

PRINTED MATERIAL - Over the past years, we have located various pieces of material that pertain to the Bally Arcade. We have also been fortunate enough to have some technical types who have done research into the machine and documented their findings. These items are available for the cost of printing and postage, and a list with some descriptive material follows:

o SERVICE MANUAL PA-1 This is a Bally-produced slick cover item subtitled "Fun and Brains". It describes the mechanical disassembly of the Arcade box, the parts list, the parts layout, and a schematic diagam. I get these in batches from Bally, and usually have a stock on hand. Cost is $1.

o SERVICE MANUAL PA-2 Also Bally-produced, it looks like PA-1 but the cover is a dull paper. It contains some not-useful data on the custom chips (clock rates and timing diagrams, but not enough/right kind to do the experimenter much good.) Also has a schematic. I do not stock this, I'm just listing it for the record.

o EXECUTIVE SOFTWARE In essence, a listing of software that does something in the Bally and how to execute it. Tom Wood disassembled the on-board ROM, and provides us with his findings. Part of the document lists the ROM subroutines that are executed by an RST 38H instruction, while the second part contains the listings from 0000-1FFF. 27 pages, $2.50

o HACKER'S GUIDE The Bally software design group made up a report of features not documented in the Bally BASIC instruction book. Virtually all of the data had been discussed in the first volume of the ARCADIAN before the release of this document as we discovered the features by experimentation. 21 pages, $2.
o MANUAL OF HARDWARE AND SOFTWARE The Bally software design group made up a large document explaining a number of subroutines, machine operations, and details of the interior workings. (it also contains the specifications of the custom chips, same as mentioned in PA-2 above). The second half provides the disassembly data for just about everything possible, including expanded on-board games data. The material is frustrating in that it occasionally discusses the "commercial" mode of operation - high resolution - that is only available with additional memory and a better operating system - capability beyond our current capacity. A shade over 300 pages, $30.
o BALCHEK A program was developed by the Bally software engineers which "looked at" the operation of the printed circuit board and determined if all was well, or if any problems existed. It would then identify the errant area. This program was placed into a 2716 chip and a couple of more parts added for a readout capability, and all boards would be inspected by this machine. Tom Wood ran this through his disassembler and has provided us with the listing, to which I have added the instructions. 60+pages, $6.50
BALLY
o BALLY BASIC Tom Wood again, this time with the Tiny BASIC cartridge taken apart - 63 pages, $6.50

PROGRAM ENHANCEMENTS occupy a goodly amount of space this issue. They have generally resulted from the material printed in Vol.2, pages 101,102, and page 4 of this volume.
o To start with, on p. 101 we described machine code techniques that cause a blob to move back and forth on the screen - rapidly - without jerking or spoiling the picture also there, as an illustration of Blue Ram usage. Well, that clever mid-Westerner, Brett Bilbray, utilized the ideas presented there and invented the enclosed program which essentially does the same thing but uses Bally BASIC only. This means that those of you who don't have the Blue Ram as yet can see this interesting operation.
o Next, Steve Walters used the identifier scheme mentioned on p. 102 and further discussed on p. 4 in conjunction with a no-memory storage scheme to get a nicer-appearing and more useful program.
o Dave Ibach explains how to do a few tricks with the tape loading schemes he outlines.
o John Perkins reveals a few more secrtes about the Blue Ram memory addition, and has some words about using the Blue Ram for BASIC storage.

PROGRAM REVIEWS We have an on-going scheme to review programs submitted by persons wishing to sell them. The reviewers use a form (reduced copy is included in this issue), and then send me the summary that is included at the bottom of the form. I print the summary plus any other comments.

USER GROUP is being attempted in the New Jersey area. Contact Rob Rosenhouse at 201-755-2289 in the evenings for more data and location/time, etc. The group will probably be a part of the Amateur Computer Group of NJ, that holds meetings at the Union County Technical Institute.

BASIC-user interrupt service for Foreground/Background processing, by Brett Bilbray with an assist from Dave Ibach, based on the Blue Ram program of page 101.

Brett Bilbray
14430 Barclay
Dearborn MI 48126

FOREGROUND - BASIC operating
BACKGROUND - Machine vector movement and writing
PROGRAM:

```
10 CLEAR
20 A=19584;B=A;C=640
30 D= -9741; GOSUB C
40 D=19518;GOSUB C
50 D=18413;GOSUB C
60 D= -8130;GOSUB C
70 D=3539;GOSUB C
80D= -1063;GOSUB C
90 D=201;GOSUB C
100 A=19680
110 D=19683;GOSUB C
120 A= 19683
130 D= -20275;GOSUB C
140 D=-3296;GOSUB C
150 D=29677;GOSUB C
160 D=19568;GOSUB C
170 D=28721;GOSUB C
180 D=-2740;GOSUB C
190 D=-10811;GOSUB C
200 D=-8731;GOSUB C
210 D=-539;GOSUB C
220 D=-9243;GOSUB C
230 D=12828; GOSUB C
240 D=19770; GOSUB C
250 D=255; GOSUB C
260 D=6151; GOSUB C
270 D=16205; GOSUB C
280 D=19768; GOSUB C
290 D= 19744;GOSUB C
300 D=6151; GOSUB C
310 D=589; GOSUB C
320 D=-7683; GOSUB C
330 D=-7715; GOSUB C
340 D=-11807; GOSUB C
350D= -3647; GOSUB C
360 D=31725;GOSUB C
370 D=19568; GOSUB C
380 D=-13829; GOSUB C
390 A=19736
400 D=14367; GOSUB C
410 D=9293; GOSUB C
420 D=2125; GOSUB C
430 A=19744
440 D=-26624; GOSUB C
450 D=16384; GOSUB C
460 D=0; GOSUB C
470 D=2050; GOSUB C
480 D=-24566; GOSUB C
490 D=-30685; GOSUB C
500 D=-21846;GOSUB C
510 D=-22846; GOSUB C
520 D=8200; GOSUB C
530 D= 2080; GOSUB C
540 D=8200; GOSUB C
550 D=0; GOSUB C
560 D=-32735; GOSUB C
570 D=1280; GOSUB C
580 D=0; GOSUB C
590 D=768; GOSUB C
600 D=5; GOSUB C
610 D=0; GOSUB C
620 D=3; GOSUB C
630 CALL (B): STOP
640 %(A)=d; A=A+2; RETURN
```

By just inputting the above program, you will be able to use the BASIC while at the same time the object (generated by the program) will move about the screen at a speed controlled by KNob #1.

DATAMAX UV-1 Zgrass University Machine" is the current name for the new Bally product mentioned here on p.84 of Vol 2. I understand that Bally builds the pieces and Datamax (350 N. Eric Dr. Palatine IL 60067) then assembles and markets it. The latest specs include: a Z80 microprocessor; 16K Screen Buffer RAM; 16K ROM that includes the ZGrass language; 32 K of user development RAM. 2 DC motor controllers (tape machine, movie or tv camera,etc.) 2 RS232 output ports for peripherals; disc drive interface; audio tape interface; and a card rack, allowing addition of other items. All of that plus more at 2495. Add a 13" color monitor at 695. This machine has been in development for the past couple of years, with the close cooperation of Dr. DeFanti of the University of Illinois who utilizes the Bally Arcade in his classes.

BYTE Magazine, in the November issue, has an article written by Dr. DeFanti which I found very interesting as it has a bit about the early history of the Bally.

REVIEWS OF OTHER COMPUTERS are requested from time to time. If you have access to one or another of the popular home computers and feel that you could write an honest comparison with the Bally, please let me know. This would be something to have done by mid-Summer, and I've a few ideas on format so that one could make some reasonable judgements by looking at a chart, table, etc.

Blue Ram Programming Tutorial. Many questions have arisen concerning the procedure for actually running programs written into the Blue Ram in machine code. Once a program (such as the one on page 101 of the ARCADIAN) has been entered into the Blue Ram via the Utility, how is it actually made to run? The RUN statement is used to run BASIC programs while the CALL statement is used to run machine code programs or routines. If the program in the Blue Ram was entered beginning at 6000H then a CALL 24576 will start that program running. Similarly, a program entered at 6100H can be run by the statement: CALL 24832 GO. The program on page 101 was entered beginning at 6000H, so a CALL 24576 GO will start it running. Note that the Blue Ram Utility has two means within it for running machine code programs: "GOSUB " followed by a four digit hexidecimal address will run a machine code program beginning at that address. Therefore, after entering the program on page 101, enter: GOSUB 6000 and the program will run. A GOSUB 6C00 will also run a program beginning at 6000H. The primary difference between GOSUB 6000 and GOSUB 6C00 is that a more complex program can be written and run using the GOSUB 6C00 entry because there is a routine at 6C00 in the Blue Ram (with the Utility loaded) that performs some special housekeeping of the Z80 registers to prevent the machine code program from upsetting the BASIC environment and clobbering the BASIC program. The GOSUB xxxx entry is a special form accepted by the Utility and it will not work from within a BASIC program. To run a machine code program from a BASIC program, the CALL nnnnn form must be used. In this case, nnnnn is the decimal equivalent of the hexidecimal address. 24576 is the decimal equivalent of 6000 hexidecimal. There is a conversion feature in the Blue Ram Utility that was inadvertantly left out of the documentation: The left square-bracket [can be entered while the Utility is running, followed by a four digit hexidecimal number and the Utility will print the decimal number equivalent. If less than four digits are used, press GO following the last digit to evoke the printout. This is especially useful when writing a machine code program at address 6450, for example, and then wondering how to run it from BASIC. The Utility would show [6450] = 25680 meaning that the machine code program at 6450H can be run by the statement: CALL25680 GO.

Blue Ram BASIC? Most Blue Ram owners are not adept at machine code programming, the major forte of the Blue Ram. Therefore we are investigating ways in which BASIC programmers and hackers can use the Blue Ram for BASIC program enhancement without having to learn any machine code. The Blue Ram owner's manual describes the procedure for using the Blue Ram memory for extended strings i.e. the %(nnnnn) statement, however there are two other ways a non-machine code programmer can extend a BASIC program. The first way consists of loading "service routines" into the Blue Ram through the Utility and then CALLing them from within the BASIC program. It is not necessary to understand how the service routine works, only how to call it. The calling sequence may require certain variables to be set up prior to the call and/or certain variables may be returned. Another fascinating possibility uses routines in the Blue Ram to extend the repertoire of BASIC statements and functions. Examples of both kinds of routines are given at the end of this paragraph. The second method for using

the Blue Ram in conjunction with BASIC programs is to enter some of the BASIC program into the Blue Ram and execute it from there. Up to 4219 additional bytes of program can be written this way for a total program length of over 6000 bytes (characters). The first step is to divide the overall program into two parts, one part for the normal 1800 byte BASIC memory and the other part for the Blue Ram. There are some restrictions on the Blue Ram part so plan the division very carefully. The following restrictions apply to Blue Ram BASIC program parts.

1. Direct GOTO snd GOSUB cannot be made to Blue Ram program parts. Instead, an indirect linkage must be defined. In the normal BASIC part the linkage takes the form: CALLnnnnn where nnnnn is the decimal equivalent of the beginning address of the Blue Ram linkage part. The Blue Ram linkage part requires four bytes: 11 hh hh C9 where hh hh is the hexidecimal address of the beginning of the Blue Ram program part being linked to. Note that the hh hh is in reverse byte order. Example:

BASIC program part:

```
10 CLEAR ;CALL24576
20 PRINT "ALL DONE"
```

Blue Ram program part:

```
6000 11 04 60 C9 3B 74 22
6007 54 48 49 53 20 4D 45
600E 53 53 41 47 45 20 49
6015 53 20 46 52 4F 4D 20
601C 54 48 45 20 42 4C 55
6023 45 20 52 41 4D 22 3B
602A 6E 32 30 0D
```

The contents of the Blue Ram part, after the linkage, is the ASCII equivalent of the BASIC statement:

```
;PRINT "THIS MESSAGE IS FROM THE BLUE RAM";GOTO 20
```

An editor for entering characters directly into the Blue Ram is being developed by Perkins Engineering to avoid having to cope with linkage programs and ASCII translations. The Blue Ram part will be entered, via the editor, in much the same way as Bally BASIC now accepts program statements.

2. Blue Ram statements are considered extensions of the statement which CALLs them. Therefore each such Blue Ram part statement CALLed from the normal BASIC part must begin with a semi-colon ; as a separator just as with any compound BASIC statement.

3. All transfers to Blue Ram program parts are via the CALLnnnnn statement. Instead of GOTO 2000 therefore, a CALL to 24576 is used. Where it is desired to reference a Blue Ram part as a subroutine, a dummy subroutine must be created in the normal BASIC part. The dummy subroutine will then transfer to the Blue Ram part. The Blue Ram part will contain the body of the subroutine as well as the RETURN statement. For example:

```
 10 GOSUB 200;STOP
200 CALL24576

;PRINT "THIS IS A BLUE RAM SUBROUTINE";RETURN
```

The last "PRINT" statement is the example subroutine in the Blue Ram.

4. Line numbers have no meaning and cannot be referenced to access the Blue Ram program parts. However, since BASIC expects a line number after each "line" of program, pseudo line numbers must be used for multiple line Blue Ram program parts. For example:

```
;PRINT "THIS IS THE FIRST LINE"
xxPRINT "THIS IS THE SECOND LINE"
xxFOR N=1TO 20;PRINT N,NxN;NEXT N
```

The xx at the beginning of each additional line is a two character filler used to take the place (and space) of the normal line number. This must always be two characters and must always be present although the choice of characters is arbitrary. If ASCII numbers are used, they will be of more meaning in the listing. The editor being developed will include facilities for applying these pseudo line numbers and for dumping and loading the Blue Ram program parts as well as listing those parts. Bally BASIC will not directly acknowledge the Blue Ram program segments when performing normal :PRINT ;LIST or LIST functions. However, for experimenters, the following statement will render a listing of Blue Ram program parts entered in ASCII via the Blue Ram Utility:

```
FOR N=(first address) TO (last address);TV=%(N);NEXT N
```

The Blue Ram Utility can be used to dump these program parts to tape just as if they were machine code segments.

A Service Routine Library is underway and routines are solicited from anyone who can provide machine code and documentation for such routines. The routines should be of a general usage nature and fully documented in terms of what they do and how to CALL them. Initial allocations for the service routines will begin at address 6D00 (27904 decimal) and progress upward until that area is full. We will then start on another area. Of course not all routines need be loaded at once. The following are three candidate service routines to begin the library:

```
6D00 CD 0F 6D 4E 06 00 D3
6D07 C0 ED 43 70 4E D3 40
6D0E C9 21 FF 6B ED 4B 6E
6D15 4E AF ED 42 C9 CD 0F
6D1C 6D 3A 70 4E 77 C9
```

These two routines allow access to up to 3072 half-size strings, where each string entry can contain a positive number from 0 to 255 or a character. The string entry number is set into A and the contents of the string is set into or retrieved from B. Example: A=14;B=QxN;CALL27930 stores the product of Q and N in half-string entry #14. FOR A=0TO 20;CALL27904;PRINT B;NEXT A will print the contents of the first 21 half-string entries. CALL27930 stores the contents of B into half-string entry A. CALL27904 fetches the contents of half-string entry A into B.

```
6D22 21 70 4E 06 FF 13 CD
6D29 CF 2F FE 3B 28 04 FE
6D30 0D 20 07 26 00 68 22
6D37 6E 4E C9 CB 74 28 E9
6D3E 04 BE 20 E5 26 00 18
6D45 E1
```

This routine compares the value in B with a list of possible matching characters and returns the matching list entry number in A. If no match is found, a number one larger than the list size is returned. The list is

placed in the statement immediately after the CALL address. For example: CALL27938,AXJKPTZ$ causes the value in B to be compared against the possible choices: "A", "X", "J", "K", "P", "T", and "Z" (the ASCII values of these letters not the contents of the variables) and sets A to the number of the match if any. Suppose that B contained the character "K". The value returned in A would be a 3 because the letter "K" was the number 3 entry in the list beginning with 0 for A, 1 for X, etc. If B had contained a value not in the list, say "Q" for example, the number returned in A would be 7, one more than the last entry in the list. The comma after the address 27938 is a required separator and is not considered part of the list. The dollar sign at the end of the list is the list terminator and is also not considered part of the list. Any character may be used as the terminator but the list must contain the terminator for proper operation of the service. The list may be of any length within the normal limits of statement length. Here is a more complex example:

```
10 FOR C=0TO 7
20 B=KP;TV=B
30 CALL27938,MARY DOE$
40 IF A=C NEXT C;STOP
50 PRINT "ERROR";GOTO 10
```

This example tests for the exact entry at the keyboard of the name MARY DOE. If any error is made along the way, the word ERROR is printed and the sequence is tried again.

By omitting the TV=B statement in line 20, this could be used as a secret password identifier. Only an exact spelling will pass. If you try to write a conventional program to do the same thing you will quickly see what a savings this service can be. And remember, routines in the Blue Ram are not lost during a RESET operation so these services can be kept on-line just like Bally BASIC. The same is true for Blue Ram BASIC program parts.

BLUE RAM NEWS. The next issue of the ARCADIAN will herald the arrival of the Blue Ram Modem Interface which is used to connect the Blue Ram to a Livermore Data Systems STAR modem. The modem and interface combination allows telephone communications between Bally owners, with local Computer Bulletin Boards, and with the mind-boggling world of the Source. A prototype of the modem interface has been working for over a month now and all that remains is pinning down the options and parts suppliers. A special communications program is in development which tailors the interface protocol to the Source formats and includes a 3 x 5 character set for a screen format of 40 characters by 14 lines, similar to APPLE's format. Also, a parallel interface to the Base 2 Model 800B printer is being considered as part of the modem interface to allow hardcopy printouts of text communications. The STAR modem is available from several dealers at $139.

Please feel free to contact John Perkins with any technical or engineering or other questions concerning the Blue Ram or its support devices and programs. (804) 428-9092 Source ID TCV403.

CORRECTION to ADDITIONAL BASIC FEATURES on page 102. The two bytes (m) and (n) are backwards in the calculation which should read (n)x256+(m). This will change the numbers on the last line to 770 vice 515 resulting in: :INPUT 770 or :LIST 770.

BLUE RAM PROGRAM. New capabilities of the Bally Arcade with the Blue Ram add-on was the program topic at a recent meeting of the Tidewater Computer Club at Norfolk, Virginia. John Perkins of Perkins Engineering presented practical home uses of the Bally and the following accessories developed by Perkins Engineering: Blue Ram, Keyboard, Modem interface, BSR controller interface, and Music compiler. Inputs were made easily with the keyboard, complete with the fancy features of N-KEY ROLLOVER, REPEAT, CONTROL CHARACTERS, etc.

The BSR, a popular home device controller, was operated via the controller interface by the Bally executing a demonstration program. A lamp was controlled by the BSR, and, while the club watched, the lamp flashed on and off and slowly dimmed and brightened as the Bally processed a time table set of instructions. We discovered how convienent and perhaps economical it would be to control automatically the hours certain home devices are turned on and off.

Next, we listened with pleasure to several Bally compiled musical selections from Bach and Handel in three part harmony. It was explained that the compiler is used to translate a person's own composition or regular sheet music into a form that is played by the Bally.

Then, a quick phone call connected the Bally (with modem) to the Source, using the modem interface. The Source is a nationwide communication/information sharing network. Previous club programs demo'ed the Source, but never was such realtime excitement generated. First of all, everyone was able to read inputs/responses in large print on the T.V. Secondly, as we sampled the smorgasbord of Source information, a New Hampshire computer group broke in, wanting to "chat" with us! Well, when they discovered we were communicating via a Bally Arcade, we read across the screen, "YOU'VE GOT TO BE KIDDING."

Yes, Bally Arcade and Perkins Engineering brought us (and New Hampshire) some surprises that night!

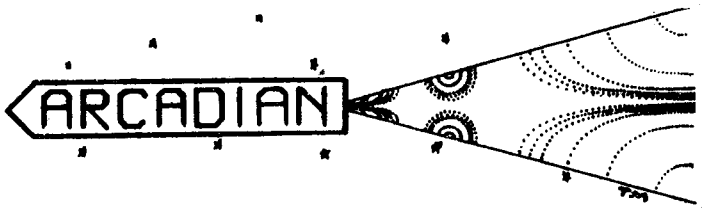
P.S. Bally users are in a majority at club meetings, and new Bally users are invited to attend bimonthly at ECPI, Stanwick Building.

Karen P. Cravedi, VEEP

```
  1 .
  2 .
  3 .
  4 .
  5 .MASTERMIND
  6 .BY BOB WISEMAN
 10 :RETURN ;NT=1;CLEAR
 20 BC=6;FC=0
100 PRINT ;PRINT "THE GAME OF MASTERMIND";PRINT
110 PRINT "COLOR CODES:"
120 R=1200;GOSUB R;PRINT "R=RED      G=GREEN"
140 GOSUB R;PRINT "O=ORANGE   B=BLUE"
150 GOSUB R;PRINT "Y=YELLOW   P=PURPLE"
180 M=0
190 FOR N=1TO 4;@(N)=RND (6);NEXT N
192 PRINT "YOUR MOVE"
195 M=M+1
196 IF M=10GOTO 1000
220 FOR N=4bM+1TO 4bM+4
240 G=KP;TV=G
245 IF G=81GOTO 1000
250 IF G=89BC=126;@(N)=1;GOTO 400
260 IF G=82BC=98;@(N)=2;GOTO 400
270 IF G=71BC=156;@(N)=3;GOTO 400
280 IF G=80BC=43;@(N)=4;GOTO 400
290 IF G=79BC=110;@(N)=5;GOTO 400
300 IF G=66BC=249;@(N)=6;GOTO 400
310 TV=31;TV=63;TV=31;GOTO 240
400 NEXT N
410 B=0;W=0
420 FOR Z=1TO 4
421 P=Mb4+Z
423 IF @(P)=@(Z)B=B+1;@(P)=@(P)+14;@(Z)=@(Z)+7
425 NEXT Z
426 FOR Z=1TO 4;P=Mb4+Z
430 FOR N=1TO 4
435 IF Z=NGOTO 470
440 IF @(N)#@(P)GOTO 470
450 @(P)=@(P)+14
455 @(N)=@(N)+7
460 W=W+1;GOTO 500
470 NEXT N
500 NEXT Z
502 FOR N=1TO 4;IF @(N)>6@(N)=@(N)-7
504 NEXT N
510 PRINT #4,B," BLACK",#2,W," WHITE
522 Z=10bB+W;NT=0
524 FOR N=-ZTO 1STEP -1
525 &(22)=-32700
526 &(16)=Nb2;NEXT N
528 &(16)=0;NT=1
530 IF B<4GOTO 195
900 PRINT "YOU WIN"
910 NT=0;FOR Z=400TO 1STEP -4
915 &(22)=-32760
920 &(16)=Z;&(17)=Zc2
930 NEXT Z;NT=1
935 &(22)=0;&(16)=0;&(17)=0
940 A=KP;GOTO 10
1000 PRINT "TOO MANY TRIES
1002 FOR Z=1TO 400STEP 4
1004 &(22)=-32760
1006 &(16)=Z;&(17)=Zc2;&(18)=Zc10
1008 NEXT Z
1009 &(16)=0;@(22)=0;&(17)=0;&(18)=0;NT=1
1010 PRINT "THE CODE WAS:"
1020 FOR N=1TO 4
1030 Z=@(N)
1040 IF Z=1PRINT "YELLOW ",
1050 IF Z=2PRINT "RED ",
1060 IF Z=3PRINT "GREEN ",
1070 IF Z=4PRINT "PURPLE ",
1080 IF Z=5PRINT "ORANGE ",
1090 IF Z=6PRINT "BLUE ",
1095 NEXT N;A=KP;GOTO 10
1200 CX=-50;RETURN
```

Bob Wiseman
118 St. Andrews Dr.
Cincinnati, OH 45245

In this version of MASTERMIND, the computer holds a four-color code that you must guess. Enter the first letter of each color guessed via the keypad. Each BLACK response means a correct color in the correct spot, while a WHITE response means a correct color in the wrong spot.



Bally Basic

By: Steve Walters
556 Langfield
Northville, MI 48167

PROGRAM TITLE & INSTRUCTIONS
WITHOUT USING MEMORY

Bally Basic programs, especially games, are nicer for the user if a title and operating instructions appear on the screen to start the operation. This lets the tape be loaded and used by a new player without having to refer to a paper instruction sheet. However, there is often a frustrating trade-off between the program instructions we would like to display, and the memory capacity we finally need for a good operating program.

The procedure described below will allow you to display a title at the beginning of the tape load, and up to nine lines of title and/or instructions at the end of the tape load, without using any memory space. The specified input command (e.g., :INPUT 2) is used to control this process.

Loading the program. When the operating program is ready to store permanently onto tape, set NT=1 (type in NT=1 and press GO). Then enter the following:

```
:PRINT; TV=0; TV=1; PRINT; PRINT "PROGRAM TITLE"; PRINT "BY PROGRAM AUTHOR";
PRINT; LIST; PRINT; PRINT "STANDBY-"; PRINT; PRINT ":RETURN; :INPUT 2"
```

Do not press GO at this point. With the above entered, start the tape recorder on record, and when the leader has gone by press GO. You will see ?? which are the TV numbers for the input number command, followed by the title and author, then the normal program listing, then STANDBY-, and then :RETURN; :INPUT 2. As soon as this last statement appears on the screen, turn the tape recorder off without entering anything else, and leave the tape cartridge at the position at which it was turned off.

The following should be noted in using the above procedure:

(a) the TV instruction shown is an example, since TV=0; TV=1 sets up an input code which will input only on the instruction :INPUT 1 (see ARCADIAN page 102). Use whatever TV code you want for your program input instruction.

(b) put whatever information you want in the PROGRAM TITLE and PROGRAM AUTHOR lines, within the limit of the buffer space. If the keypad entry locks out on you before you have entered the above material, then you have put too many characters in these two lines.

(c) the last instruction :RETURN; :INPUT 2 closes the IO port and then waits to re-open it when it finds the proper TV input code. The above sample uses 2 as an example. If more than one program is included on a tape, both of the input codes for each program should be unique values, used only once on the tape.

Loading the message. A separate program is now used to load the message onto the tape which you want to appear after the program load but before the program operation begins, such as the title and playing instructions. Press RESET to clear out the operating program, and then enter the following:

```
 10 CLEAR
 20 TV=0; TV=2                        (this should correspond to the value in the last
 30 PRINT; PRINT                           statement of the program load sequence)
 40 PRINT ".PROGRAM TITLE"
 50 PRINT ".PROGRAM AUTHOR"           (each line of message should contain no more than
 60 PRINT ".********************"          23 characters/spaces after the period)
 70 PRINT ".INSTRUCTIONS"
 80 PRINT ".INSTRUCTIONS"
 90 PRINT ".INSTRUCTIONS"
100 PRINT ".INSTRUCTIONS"
110 PRINT ".INSTRUCTIONS"
120 PRINT ".TO START PUSH KEY 1"                          (see note a)
130 PRINT ":RETURN; NT=0; BC=133; K=KP; GOTO 1            (see note b)
```

All message lines (Line 40 thru Line 110) must be used to fill the screen and scroll unwanted material out of sight. If you do not need all the lines, or want spaces between parts of the instructions, write PRINT only in that line.

When the message loading program is ready to record onto the tape, enter :PRINT; RUN without pressing GO. Then start the tape on record and press GO. As soon as the material in Line 130 appears on the screen, turn the tape recorder off without entering anything more.

The tape is now ready to input and display the message.

(a) Line 120 is based on the key-press in Line 130 being used to start the main program operation. This requires no memory space. However, I usually prefer to have the hand controller operate everything if possible, which requires a line in the operating program, as follows:

(1) change Line 120 in the above message program to

```
120 PRINT ".TO START PULL TRIGGER"
```

and eliminate K=KP in Line 130.

(2) add this line to the operating program:

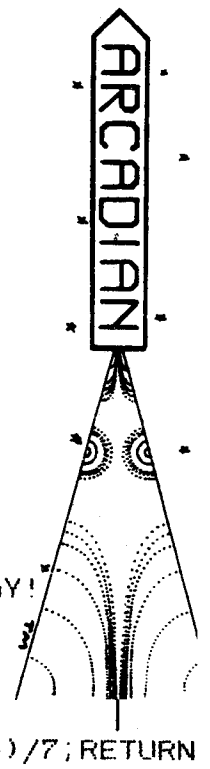
```
1 BOX 0,-32,159,7,2; IF TR(1)#1 GOTO 1
```

The BOX instruction is not necessary, but it blanks out the material which was printed by Line 130 in the message program, for a nice screen display. Line 1 in the program simply loops on itself while the player is reading the message, until he pulls the trigger. A CLEAR instruction will probably be needed at the next operating line.

(b) Line 130 contains some items as examples. If you need NT set at a specific value, or a color set, a value set (e.g., A=200) or other item once at the beginning of the operating program, it can be done as part of Line 130 without using program memory space. Be careful about the length of Line 130, since it will cause the message to scroll out of sight if it is too long. You could use two lines for this if needed.

```
  1  .
  2  . DAY-OF-THE-WEEK
  3  .        AND
  4  . CALENDAR PROGRAM
  5  .
  6  .BY KIRK GREGG
  7  .SEPTEMBER '80
  8  .
  9  :RETURN
 10  CLEAR;LIST 2,3
 20  PRINT;PRINT " ENTER ANY DATE BETWEEN
 30  PRINT " 1-1-1583 & 12-31-9999
 40  PRINT;INPUT " MONTH      (1-12)?"M
 50  IF (M<1)+(M>12)GOTO 810
 60  N=31;IF M=2 N=29
 70  IF (M=9)+(M=4)+(M=6)+(M=11) N=30
 80  PRINT #1," DAY      (1-",N,;INPUT ")?"D
 90  IF (D<1)+(D>N)GOTO 840
100  INPUT "  YEAR(1583-9999)?"Y
110  IF (Y<1583)+(Y>9999)GOTO 870
120  IF M#2 GOTO 150
130  X=Y/4;IF RM#0 GOTO 140
135  X=Y/100;IF RM#0 GOTO 150
136  X=Y/400;IF RM#0 GOTO 900
137  GOTO 150
140  N=N-1;IF D=29 GOTO 900
150  Z=Y;K=M;E=D;IF M<3 M=M+12;Y=Y-1
160  GOSUB 510
170  I=RM;IF I=0 I=7
180  D=1;GOSUB 510
190  L=RM;IF L=0 L=7
200  W=1;IF L=1 W=0
201  .
202  .PRINT CALENDAR
203  .
210  NT=0;CLEAR;FC=RND(244);BC=FC+12
220  CX=-40;CY=39;GOSUB 600+K
230  PRINT #6,Z;BOX 0,39,141,9,3
240  X=(L-1)x20-62;Y=28
250  FOR C=1 TO N
260  CX=X;CY=Y;PRINT #2,C;
270  IF X=-62 W=W+1
280  L=L+1;X=X+20
290  IF L>7 X=-62;Y=Y-11;L=1
300  NEXT C
301  .
302  .DRAW MATRIX
303  .
310  X=70;Y=23
320  FOR C=1 TO W
330  LINE -X,Y,0
340  LINE X,Y,1
350  Y=Y-11;NEXT C
360  Y=34-Wx11
370  FOR C=-70 TO 70 STEP 20
380  LINE C,34,0
390  LINE C,Y,1;NEXT C
400  .
401  .PRINT DAY NAME
402  .
410  CX=-40;CY=-40
420  PRINT #2,E," => ",;GOSUB 700+I
430  .
431  .ASK IF DONE
432  .
440  K=KP;CLEAR;PRINT;PRINT
450  PRINT "  ANY MORE (Y/N) ? ";K=KP
460  IF K#78 CLEAR;PRINT;NT=3;GOTO 40
470  .
471  .END OF JOB
472  .
480  PRINT;PRINT;PRINT " HAVE A BEAUTIFUL DAY!
490  STOP
500  .
501  .CALCULATE D.O.W.
502  .
510  X=(D+2xM+Y+6x(M+1)/10+2-Y/100+Y/400+Y/4)/7;RETURN
```

Kirk Gregg
P.O. Box 232
Waimea, HI 96796



```
598   .
599   .MONTH NAMES
600   .
601   PRINT "JANUARY",;RETURN
602   PRINT "FEBRUARY",;RETURN
603   PRINT "MARCH",;RETURN
604   PRINT "APRIL",;RETURN
605   PRINT "MAY",;RETURN
606   PRINT "JUNE",;RETURN
607   PRINT "JULY",;RETURN
608   PRINT "AUGUST",;RETURN
609   PRINT "SEPTEMBER",;RETURN
610   PRINT "OCTOBER",;RETURN
611   PRINT "NOVEMBER",;RETURN
612   PRINT "DECEMBER",;RETURN
698   .
699   .DAY NAMES
700   .
701   PRINT "SUNDAY",;RETURN
702   PRINT "MONDAY",;RETURN
703   PRINT "TUESDAY",;RETURN
704   PRINT "WEDNESDAY",;RETURN
705   PRINT "THURSDAY",;RETURN
706   PRINT "FRIDAY",;RETURN
707   PRINT "SATURDAY",;RETURN
800   .
801   .ERROR MESSAGES
802   .
810   PRINT;PRINT " INVALID MONTH!!
820   PRINT " 1 TO 12 ONLY PLEASE!!
830   FOR C=1 TO 450;NEXT C;CLEAR;PRINT;GOTO 40
840   PRINT;PRINT " INVALID DAY!!  ",
850   GOSUB 600+M
860   PRINT;PRINT #3," HAS ONLY",N," DAYS!!";GOTO 80
870   PRINT;PRINT " INVALID YEAR!!
880   PRINT " 1583 TO 9999 ONLY PLEASE!
890   PRINT;GOTO 100
900   PRINT;PRINT " INVALID DATE!!
910   PRINT;PRINT #6,Y," IS NOT A LEAP YEAR!!
920   GOTO 830
```

TUTORIAL: TAPING MEMORY

by Dave Ibach

Here is an alternate way to put your program on tape:

Dave Ibach
19553 Dartmouth Pl.
Northville, MI 48167

```
>A=-24576
>B=%{20050}
>PRINT B
    nnnnnn ——————————→use it here ——→
>NT=1;:PRINT;PRINT ▽.▽;PRINT ▽A=-24576;B=nnnnnn;
FOR N=A TO B;%{N}=KP;NEXT N;%{20050}=B;
:RETURN;RUN▽;FOR N=A TO B;CY=40;TV=%{N};
NEXT N;:RETURN get tape moving then press GO
```

The program can be loaded from tape to memory by using the standard :INPUT command.

Why would you want to do it this way? First, the program doesn't list on the screen during loading. If you have a program that holds surprises for those who play, such as the printing of clever remarks, you might not want to reveal the program listing during loading.

Second, you can easily include the contents of the string variable with this approach. Do it this way:

```
>A=-24576
>B=%{20050}
>C=B+2x{the number of string variables desired
            including @{0}}
>PRINT B
    nnnnnn ——————————→use it here ——————————→
>PRINT C
    mmmmmm ——————→ use it here ——→
>NT=1
>:PRINT;PRINT ▽.▽;PRINT ▽A=-24576;C=mmmmmm;
FOR N=A TO C;%{N}=KP;NEXT N;%{20050}=nnnnnn;
:RETURN;RUN▽;FOR N=A TO C;CY=40;TV=%{N};
NEXT N;:RETURN get tape moving then press GO
```

A third reason for taping your programs in this fashion is to save memory. The remainder of this tutorial will explain how.

In your program, you use statement numbers for three reasons:

1. As the object of a GOTO or GOSUB
2. To terminate the scope of an IF statement
3. The input buffer doesn't allow a statement to be greater in length than 102 keystrokes.

Each time you are constrained by #3, you are throwing away two memory locations. In addition, if you are storing data in

the first few lines (as George Moses does in his music tapes), it can be particularly annoying to have to deal with statement numbers that recur too frequently. To get rid of these extraneous statement numbers:

1. Make a handwritten list of the statement numbers that exist for this reason.
2. Rewrite each one of these statements inserting a semi-colon after the statement number. This is done so that we can concatenate the text to the previous statement. This step is unnecessary for data lines.
3. Squeeze out statement number x like this:

```
>FOR N=-24576 TO 0:IF %(N)#xNEXT N
>FOR I=N-1 TO -22775STEP 2:%(I)=%(I+3):NEXT I:
 %(20050)=%(20050)-3
```

Hint: to avoid the tedium of doing step 3 repeatedly, make these the first two statements of your program, even if you have to borrow the space by temporarily eliminating other lines:

```
1 INPUT "STMT # TO ELIMINATE?"A:FOR
 N=-24576 TO 0:IF %(N)#A NEXT N
2 FOR I=N-1 TO -22775STEP 2:%(I)=%(I+3):NEXT I:
 %(20050)=%(20050)-3:STOP
```

Now simply RUN once for each statement number you wish to eliminate.

4. Et voila, it's done! If you followed the previous hint, take out lines 1 and 2 and restore any lines you may have temporarily removed.

Now your program is listable, executable, and smaller in size. PRINT SZ if you are skeptical.

But you can't write it to tape and reload using the normal technique, because some of the statements are now too big for the buffer. You'll have to write it to tape using one of the techniques given at the beginning of this tutorial.

HORSERACE by Paul Slezak

```
  5 NT=0;CLEAR ;FC=155;BC=160;N=0;FOR Z=1TO 4;@(Z+20)=500;NEXT Z;&(9)=180
 10 CY=5;PRINT "WELCOME TO ARLINGDUNG PARK";INPUT "# OF PLAYERS? (1-4)"A;IF A>
GOTO 10
 15 GOSUB 290
 20 CLEAR ;CX=-20;PRINT "RACE #",;PRINT #1,N;T=0;S=0;GOSUB 600
 30 FOR Z=1TO 8;PRINT #2,Z,;@(Z+8)=RND (5)+RND (3)+1;CX=38;PRINT #1,@(Z+8),;CX=
45;PRINT ":1",;CX=-50;GOTO 30+Z
 31 PRINT "ALBADEXTER";NEXT Z
 32 PRINT "JO L IN";NEXT Z
 33 PRINT "C BICUSPID";NEXT Z
 34 PRINT "SEATLE SLEWED";NEXT Z
 35 PRINT "WOMAN O'WAR";NEXT Z
 36 PRINT "DUSTY PAUL";NEXT Z
 37 PRINT "DIABLO";NEXT Z
 38 PRINT "SECRETARYLESS";NEXT Z
 40 FOR Z=1TO A;IF @(Z+20)=0@(Z+16)=0;NEXT Z;GOTO 100
 50 CY=-32;PRINT "PLAYER #",;PRINT #1,Z,;CX=-20;INPUT B;CY=-32;CX=10;INPUT C
 60 IF B>0IF B<10GOTO 80
 70 GOTO 50
 80 IF C>@(Z+20)CY=-32;PRINT "SORRY,I'M NO LOAN SHARK";GOSUB 610;CY=-32;FOR D=1
TO 23;PRINT " ",;NEXT D;CX=-78;GOTO 50
 90 @(Z+16)=B;@(Z+20)=@(Z+20)-C;@(Z+24)=C;NEXT Z
100 FC=160;NT=5;PRINT "146 641 641 641000000";CLEAR ;NT=0;FC=155;CY=35
110 FOR Z=1TO 10;CX=70;GOTO 110+Z
111 PRINT "X";NEXT Z
112 PRINT "X";NEXT Z
113 PRINT "F";NEXT Z
114 PRINT "I";NEXT Z
115 PRINT "N";NEXT Z
116 PRINT "I";NEXT Z
117 PRINT "S";NEXT Z
118 PRINT "H";NEXT Z
119 PRINT "X";NEXT Z
120 PRINT "X";NEXT Z
130 CY=25;FOR Z=1TO 8;PRINT #2,Z,;CX=-60;PRINT ":";NEXT Z;GOSUB 610
140 CY=25;FOR Z=1TO 8;CX=-60;PRINT "..";NEXT Z
150 NT=1;&(20)=50;&(21)=205;FOR Z=1TO 35;MU="V";NEXT Z;NT=0;&(20)=0;&(21)=0;CY=
25;FOR Z=1TO 8;PRINT "     ";NEXT Z
155 CY=0;PRINT "..AND THEY'RE OFF!!";GOSUB 610;CY=0;PRINT "                   "
160 GOSUB 610
170 CY=25;FOR Z=1TO 8;@(Z)=@(Z)+((RND (4)+8+RND (5))-(RND (@(Z+8))));CX=@(Z);PR
INT #1,Z;IF @(Z)>65S=1
175 NEXT Z;GOSUB 610
180 IF S=1GOTO 200
190 CY=25;GOSUB 620;GOTO 160
200 G=0;FOR Z=1TO 8;IF G>@(Z)NEXT Z;GOTO 220
210 X=Z;G=@(Z);M=@(Z+8);NEXT Z
220 B=0;FOR Z=1TO 8;IF @(Z)>65B=B+1
230 NEXT Z;IF B>1GOSUB 500
240 CLEAR ;CY=0;PRINT "THE WINNER IS #",;PRINT #2,X;GOSUB 610
250 FOR Z=1TO A;IF @(Z+16)=X@(Z+20)=@(Z+24)bM+@(Z+20)
260 NEXT Z
290 CLEAR ;CX=-30;PRINT "YOU HAVE"
```

Paul Slezak
1818 S. 59th Ave
Cicero, IL 60650

19 SPACES

```
 300 FOR Z=1TO A;PRINT "PLAYER #",;PRINT #1,Z,;PRINT #2," $",;PRINT #2,@(Z+20);N
EXT Z
 310 FOR Z=1TO 3000;NEXT Z
 320 IF N>8GOTO 400
 330 N=N+1;IF N>0GOTO 20
 340 RETURN
 400 CLEAR ;FC=79;BC=82;CY=16;CX=-35;PRINT "RACES OVER!!";PRINT ;PRINT
 410 PRINT "IF YOU WANT 9 MORE RACES    PRESS 1";A=KP;IF A#49STOP
 420 GOTO 5
 500 FOR Z=1TO 6;CLEAR ;FC=212;BC=209;CY=0;CX=-35;PRINT "PHOTO FINISH";NEXT Z;BC
=160;FC=155
 510 RETURN
 600 FOR Z=1TO 8;@(Z)=-78;NEXT Z;RETURN
 610 FOR F=1TO 300;NEXT F;RETURN
 620 FOR D=1TO 8;CX=@(D);PRINT " ";NEXT D;RETURN
```

REVIEW FORM is shown below. This will give you the code for the various numbers used and the categories.

BALLY PROGRAM REVIEW

Date:_____
Name of Program:_____Cassette Name_____
Description:_____
Source:_____Price_____
Reviewed by:_____Age:____

Circle score for each item 0 1 2 3 4 5 6 7 8 9 on scoring line.

PROGRAM DOCUMENTATION (PD)

No Instructions 0 1 2 3 4 5 6 7 8 9 Very Clear Description Listing, Flowchart, Instructions

PROGRAM POLISH (PP)

Sloppy, Amateurish 0 1 2 3 4 5 6 7 8 9 Professional

USE OF SPECIAL FEATURES (USF)

Minimal Used 0 1 2 3 4 5 6 7 8 9 Great Use of Features Graphics, Sound, Handles

LEVEL OF CHALLENGE (LC)

Not Challenging 0 1 2 3 4 5 6 7 8 9 Very Challenging

ORIGINALITY AND CREATIVITY (OC)

Adapted, Same Old Stuff 0 1 2 3 4 5 6 7 8 9 Totally Brilliant and Unique

LEVEL OF INTEREST (LI)

Boring 0 1 2 3 4 5 6 7 8 9 Fascinating

EDUCATIONAL VALUE (EV)

Little 0 1 2 3 4 5 6 7 8 9 Really Learn Facts and Skills

EASE OF USE (EU)

Awkward, Inconvenient 0 1 2 3 4 5 6 7 8 9 Easy, Quick, Convenient

OVERALL VALUE (OV)

Almost Worthless 0 1 2 3 4 5 6 7 8 9 Everyone Should Buy

CASSETTE REVIEW Date: 8/30/80

CASSETTE NAME: Program Tape #2
PROGRAMS ON CASSETTE: Wumpus 3.1; Blackjack II; Life 4.2; File Create; Tape Input/Output; Text Editor; Biorhythm 3.0

CASSETTE PRICE: $10.00 LISTING PRICE: Not Available
SOURCE Name: Mark Keller
Address: 9536 Shumway Drive
City: Orangevale State: CA ZIP: 95662

Reviewed by Bill Rueger Age 31

* * * * * * * * *

PROGRAM NAME: Wumpus 3.1
DESCRIPTION: This is a very good version of Wumpus. Perhaps the best feature is that it allows you to program your own cave sequence and also to save it on tape for later use. This is done using the Tape Input/Output program that follows. This attribute, and the documentation, is worth the price of the cassette alone.

RATING % based on applicable rating items. 53/72= 73.6%
PD= 7 PP= 8 USF= 5 LC= 6 OC= 7 LI= 6 EV= XX EU= 7 OV= 7
Time to play 10-30 min. For ages All # of players 1

* * * * * * * * *

PROGRAM NAME: Blackjack II
DESCRIPTION: This program implements the game of Blackjack for one player vs. the computer. The rules are the same as the casinos with the exception that if you split your cards the dealer will play against each hand after you stand on it. The dealer also alternates between two decks and only deals the top 26 cards from each deck. If you do not already have the Blackjack Videocade, this version is very good, minus the graphics and multi-player capability. It also allows for insurance as do the casinos.

RATING % based on applicable rating items. 42/72 = 58.3%
PD= 7 PP= 8 USF= 1 LC= 5 OC= 5 LI= 5 EV= XX EU= 6 OV= 5
Time to play Variable For ages 10 up # of players 1

* * * * * * * * *

PROGRAM NAME: Life 4.2
DESCRIPTION: While not as fast and graphically "clear" as other versions I have encountered, this version does have some helpful features that make it a must for any Life addict to acquire. It has the ability to stop the program at the end of a generation and change or insert additional colonies. The program can then continue at the generation it was on. It also allows you to set up the program visually through keyboard entry, eliminating the need to use graph paper and co-ordinates.

RATING % based on applicable rating items. 38/ 63 = 60.3%
PD= 6 PP= 7 USF= 4 LC= XX OC= 5 LI= 5 EV= XX EU= 6 OV= 5
Time to play For ages All # of players

* * * * * * * * *

PROGRAM NAME: File Create
DESCRIPTION: Allows you to make files of string data. This is a very useful program which allows you to create data records for later use. For example, you can create a file of cave locations to be used with the Wumpus Program. This can be stored on tape and used whenever you wish. A most interesting program, with good documentation on how it's down.

RATING % based on applicable rating items. 50/ 63 = 79.4%
PD= 4 PP= 8 USF= XX LC= XX OC= 9 LI= 7 EV= 8 EU= 6 OV= 8
Time to play _____ For ages_____ # of players

* * * * * * * * *

PROGRAM NAME: Tape Input/ Output
DESCRIPTION: This program lets you make copies and also enter them into the computer of Data Files. It is used in conjunction with the File Create program. It is an excellent program with Documentation on "how it works".

RATING % based on applicable rating items. 49/ 54 = 90.7%
PD= 8 PP= 9 USF= XX LC= XX OC= 8 LI= XX EV= 8 EU= 8 OV= 8
Time to play _____For ages_____# of players

PROGRAM NAME: Text Editor
DESCRIPTION: This is another useful program that allows you to edit each line in your program to correct errors, make changes etc. It takes up only 3000 bytes so can be used with lengthy programs. It lets you access any line in a program and edit it without having to retype the entire line. A must for anyone who writes his own programs.

RATING % based on applicable rating items. 50/ 54 = 92.6%
PD= 8 PP= 9 USF= XX LC= XX OC= 8 LI= XX EV= 9 EU= 9 OV= 8
Time to play _____For ages_____# of players

PROGRAM NAME: Biorhythm 3.0
DESCRIPTION: This program plots your particular Biorhythm for a given month and day. It does so by graphic display. It also includes "experimental" cycles of Health, Sex, Precognition, and Natural High. It is a very graphic program and easy to read.

RATING % based on applicable rating items. 49/ 63 = 77.8%
PD= 7 PP= 7 USF= 7 LC= XX OC= 7 LI= 7 EV= 0 EU= 7 OV= 7
Time to play _____For ages All # of players 1

PHILOSOPHY A bit of space otherwise unoccupied, so I shall inject a few words... There is no single place that one can go to for full information about the Bally Arcade, or its version of Palo Alto Tiny BASIC. There is no book that will convert from any other dialect of Tiny BASIC (upwards of 400) into Bally BASIC. Effort in this area requires a lot of cut-and-try, and certainly an understanding of what was intended by the originating programmer. That is, you have to know what he had in mind when he wrote the program. I have tried to fill the void of understanding how/why the Bally works the way it does with the ARCADIAN, and through the help of a lot of contributors, we have been able to document quite a bit, and to begin to understand its operation. All this material is included in the earlier issues of the ARCADIAN, from the first one to the latest. Those of you who are new subscribers should avail yourselves of the material that is in the first two volumes, because it is not my intent to reinvent the wheel every year and go over that material. Aside from the knowledge you will obtain, there are some really great programs in there, like Collins' Checkers, or Perkins' Ojello, to pick a couple. If someone were really ambitious, they could extract the tutorial material, and the game material, and produce a sort of digest, but not me. Dick Hauser has put a lot of games on tape, so part of that has been done already. Back issues are available at $10 for each year, 1979 and 1980.

ADS:

- Programs available: Mastermind(1 player);Spell and Score(2);Slot Machine(1); Craps (1); Checkbook Balancer; Russian Roulette (to 10); above 2.50 each. Dragon Hunt (1) 1.50 - try to find the invisible dragon on a 10x10 table. Prices are for listing or for recording on your tape. All old programs revised and improved, and all programs come with complete instructions,guaranteed bug-free. S.Walpole,11480 Beirut Ct. #204, Sappington, MO 63126
- Bally Software: Memory Maze (1-2 players) study the maze, then try to move thru it when it is invisible. scoring, color, music. Crazyface - Bally draws chinaman, foctball player,witch,singer, mountie; then you move hats, eyes, noses,mouths,necks to develop your own crazy faces. Hidden Word Finder - manipulates a hidden word puzzle. All three on tape plus full documentation 7.95: S.Walters,556 Langfield, Northville MI 48167

AN EXPANDED ISSUE this time as a number of good items came in. Also ,the expected postal rate increase is apparently a bit off, so I have some excess money which is going into the larger volume of printing and postage.

BUT I'm also drained out of material, so please keep the programs and tutorials coming in for our mutual understanding.

The SOURCE is mentioned a couple of times inside, it is composed of a giant computer having lots of data stored in it, such as the latest stock reports, UPI news reports on both national and local levels,a New York Times data bank, lots more. It also allows classified ads to be placed or read, private letters between subscribers, or 'face-to-face' talking. You can use it as a word processor or a computer, then store your work 'till another day. All you need is a computer terminal (Bally and TV screen) and a Modem to tie to the telephone lines. Full details in the next issue.

HAPPY HOLIDAYS to you all!!!

Bob Fabris

ARCADIAN

The SOURCE TCD 959
Robert Fabris, joyous at Noel
3626 Morrie Dr.
San Jose, CA 95127

FIRST CLASS