



AUGUST 1986 ISSUE NO. 11

FOR THE RECORD

by Ed Bittner
Recording Secretary

The July meeting of the West Penn Users Group started off with a bang as we almost lost Scott Coleman's computer again. Had he brought a case of pop as asked in the June Newsletter we could rule out sabotage. Scott opened the meeting with a good news / bad news report that J. Willforth was out of town. (You decide.) The meeting went smoothly without any interruptions.

Hot news items from the president included a computer show in Horsehead, N.Y. featuring the new Myarc system, the consecutive issues of the Smart Programmer, that Games and Gadgets (Cent III Mall) was selling cartridges (from \$2-20), and that our membership continues to grow. Several avenues for a new meeting place are still being explored with the Penns Woods Community Center being voted down. Monday night meeting dates were still a favorite. Some members questioned the reliability of some hardware/software firms. Without mentioning them here the best advise is to talk to other members in the group prior to making major purchases.

Scott asked and received a favorable vote (unanimous) for the club to purchase 500 diskettes and a dozen data cases to be sold to club members at \$.50 each (lots of ten) and \$8.00 ,respectively. Support your club !

After much confusion, dismay, and enlightenment (I think in that order), about drives, controllers, jumpers, splicer cards, and termination packs, I still would like to see someone from our group write an article on some of the above. Meanwhile we're in need of raffle prizes (suggestions, donations or ?) would suffice. Norm X. showed a neat inclusion of graphics from TI artist with a TI- writer output. Impressive !

Preview of coming attractions. In the August newsletter and at the August meeting look for : Articles on TI-writer by Stan X , a report on the new Myarc computer by X X , and don't miss the return of yes , J. Willforth, our editor in chief, with a demo using a homemade expansion box (\$ 35.00 ,kit form). Be There !!!

Submitted late apologetically,

Scoops Bittner

P.S. Following a suggestion by the president, I will bring one case of cold pop (soda) to be sold for \$.50/can. Proceeds will, of course, go to the club. I must get to know peoples last names, using X's shows 0 class.

John Willforth asked me if I would write a column on T. I. Writer for the Newsletter. I told him that I did not feel like an expert on the program but I would try. If anyone has any comments please feel free to make them because I can learn from you also.

The best place to start is at the beginning. When I first unpacked T. I. Writer it was quite intimidating, a cartridge, a disk and that BIG manuel. It is not as bad as it seems and it really is an easy program to learn to use. One must have the following 1) Consol, 2) 32K memory expansion, 3) RS232 card, 4) disk drive (2 disk drives are best but you can do very nicely with one) and a printer. (The printer should be Epson compatible).

First make a copy of the program and use the original as a backup. (Very important.)

Place the cartridge in the port and the program disk in the drive. (I am assuming one disk drive.) Power up the system and get T. I. Writer from the menu using the English version. Next you will see 1. EDITOR 2. FORMATTER 3. UTILITY. We will discuss the Editor first because all writing is done here. Press (1) and the disk will now run putting the editor program in memory. A line will show across the top with a cursor (you are in Command Mode). The line says Edit, Tabs, Files, Lines, Search, RecoverEdit, ShowDirectory, Quit. We will now just discuss Tabs and Edit.

The line starting with "Edit" is called the Command Mode. Let us now set our tabs (and margins). Press T and the cursor will move to a numbered line. First hit the space bar and remove the L then with the "Fctn D" move the cursor to the 10 space and type L then move the cursor with "Fctn D" to 70 and type an R, press enter and you are in the editing mode and you can type anything you want with the margins set at 10 and 70. (If you want different margins just enter them as described). You do not have to press the enter key at the end of the line unless you want to start an new paragrph. You are in Word-wrap mode.

If you want to print out your document at this stage just press Fctn 9 and you are back in Command Mode. Now press F (files) and you will see more selections such as LoadF, SaveF, PrintF, DeleteF, Purge or ShowDirectory. At this point type PF and you will now see "PRINT FILE, enter devicename:", at this point if you have a parallel printer enter P10 and your document will be printed out as you typed it.

Try using the word processor and reading the instructions. You will find it easy to use. More next time.

THE INCOMPLETE AND ABRIDGED
COMPUTER TERM LEXICON

From the Washington, DC 'Gazette'
of July 13, 1983.

ANALOG - An hors d'oeuvre, usually made from cheese and covered with crushed nuts.
APPLE - Typically a device used to seduce men, usually equipped with display screens and/or worms.
BAR CHART - A list of places to go when it is Miller time.
BASIC - A form of motion sickness that occurs only after waiting for two hours to cross the bridge and continue on to Ocean City.
BINARY - Possessing the ability to have friends of both sexes.
BIPOLAR - A descriptive term referring to someone who has homes in Gnome, Alaska and Buffalo, New York.
BIT - Similar to a nibble. Commonly eight nibbles in a mouthfull. See byte.
BUBBLE MEMORY - A derogatory term, usually referring to a person's intelligence. See also "vacuum tube."
BUFFER - A process through which computers are treated to prevent stomach upset.
BUG - Small living things that small living boys throw on small living girls.
BYTE - A mouthfull, as in "How many bytes in a Big Mac?"
CARRIAGE RETURN - The act of returning a vehicle to the rental counter.
CASSETTE - A very small petite cass...usually achieved through exercise.
CHARACTER DENSITY - The number of very weird people in your computer club.
CHIP - Any number of small crunchy objects often served with onion dip.
CLOSED LOOP - A method of execution no longer in vogue except in Iran.
CODE - Usually lasts about three to five days, accompanied by sore throat, runny nose and fever.
COMMAND - Statement presented by a human and accepted by a computer in such a manner as to make the human feel as if he is in control.
COMPILE - A heap of decomposing vegetable matter.
CONVERSATIONAL MODE - Describes the typical office on a Monday after a Steelers game.
COUPLING - An activity usually preceded by marriage, but not necessarily.
CPU - A juvenile way of telling your dog he missed the paper.
CRT - A movie about a little alien who forgets his telephone number and must write home.
CURSOR - An expert in the use of four letter words.
DEBUG - The act of placing shoe leather against a small creeping creature.

DISC DRIVE - Propulsion method developed by a well known foreign car manufacturer.
DOWNTIME - Title of a movie song popularized by Petula Clark.
DUMP - The EPA's answer to health.
EPROM - Movie actor Eprom Zimbalist, Jr.
ERROR - Something only humans can commit.
FIFO - A cute name for a dog.
FLOPPY DISCS - A defect occurring in all 1982 disc drives, necessitating a factory recall.
GIGO - A movie industry acronym referring to the numerous Gidget Goes **** movies, i.e., GIGO Hawaiian, GIGO Crazy, GIGO Surfing, etc., etc.
GLITCH - Scientific name for the little balls of fuzz that collect in navels.
HARDWARE - Typically boots, leather and chains. Contrast with software.
INTEGRATED CIRCUIT - The end result of busing.
INTERFACE - The opposite of "Getouttamyface."
K - A term used in employment ads to disguise how much they are really willing to pay.
KEYPUNCHING - An activity similar in most aspects to cowpoking.
MEGAHERTZ - A very large car rental company.
MODEM - A contraction. Commonly used as in "Give me some modem cookies."
NETWORK - The occupation of a fisherman.
ON LINE - A statement shouted at tennis judges in response to serves being called out.
OUTPUT - What people who talk backwards do to their cat.
PROGRAM - What commercials try to do to us.
RAM - A male sheep with horns.
REAL TIME - Here and now, as opposed to fake time which only occurs there and then.
RECURSIVE - See "recursive."
ROM - A RAM after a delicate operation.
SEMICONDUCTOR - A person hired to lead an orchestra before he has graduated from the famous director's school.
SERIAL PROCESSING - The procedure through which corn flakes are arrived at.
SNOBOL - A small white round object thrown in the winter.
SOFTWARE - Typically silk nighties, nylons, garter belts. Contrast with hardware.
STRING - A object a kitten will play with but if given to a cat will cause him to leave home.
TERMINAL - What most people have to be before consenting to see a doctor.
TRANSISTOR - A sibling, i.e., a transbrother.
TYPEWRITER - A contradiction in terms.
VACUUM TUBE - A derogatory term. See "bubble memory."

DRAW



NIBBLES BYTES

CASSETTES---You can't beat the quality and price of a 2-pack of C-60, normal bias, Type D tapes made by IDK Corp. The 2-pack runs \$3.00 regular price and \$2.00 on sale at most discount stores. These tapes are much better than Radio Shack's Realistic Low Noise cassettes. If you switch to a disk system, the tapes can be reused to make high quality stereo recordings; they're that good.

MODEMS---According to an article in July 1986 "LOTUS" Magazine, the price of auto-answer, auto-dial, 1200 baud modems should drop to about \$100 because of new chip technology that reduce the number of chips needed to make a modem. Two chips now replace the 5 to 9 chips used in modems in the past.

TI PARTS---Radio Shack is still selling TI 99/4A keyboards for \$3.95, Stk. No. 277-1023, advertised in the Shack's August flyer.

R. P. Sadusky, West Penn 99ers

GETTING THE MOST FROM YOUR CASSETTE SYSTEM
BY MICKEY SCHMITT
NUMBER 4
KEEPING YOUR CASSETTE TAPES AND PROGRAMS ORGANIZED
PART II

THIS MONTH I AM CONTINUING WITH THE TOPIC OF KEEPING YOUR CASSETTE TAPES AND PROGRAMS ORGANIZED - USING THE INFORMATION GENERATED BY LAST MONTH'S 3 X 5 INDEX CARDS - AS THE FOUNDATION FOR THE FOLLOWING PROGRAM.

ALTHOUGH THIS PROGRAM WILL WORK AS WRITTEN - YOU ARE ENCOURAGED TO MAKE ANY CHANGES THAT YOU MAY WANT IN ORDER TO MEET YOUR OWN PERSONAL NEEDS. DON'T BE AFRAID TO DO A LITTLE EXPERIMENTING. IT CAN'T HURT AND YOU JUST MAY LEARN A THING OR TWO IN THE PROCESS.

THIS PARTICULAR PROGRAM WAS CREATED WITH THE INTENT OF GIVING YOU THE FOLLOWING OPTIONS: YOU MAY EITHER TYPE IN THE FOLLOWING PROGRAM AS LISTED - FILLING IN THE BLANKS AS THEY APPEAR OR YOU COULD JUST TYPE IN THE INFORMATION THAT WOULD APPEAR IN THE BLANK AREA AND FORGET ABOUT TYPING IN ALL THE "FORMAL TITLES". PERSONALLY I LIKE THE LATTER CHOICE MYSELF AS IT SAVES ALOT OF UNNECESSARY REPETITIVE TYPING AND IT KEEPS MY SCREEN INFORMATION DOWN TO A BARE MINIMUM WHEN I RUN THE PROGRAM.

NEXT MONTH'S TOPIC WILL BE CASSETTE TIPS - TRICKS - AND TIDBITS. IT SHOULD PROVE TO BE QUITE INTERESTING - AS I PASS ALONG WHAT I'VE FOUND OUT THE HARD WAY - AND WHAT I'VE LEARNED FROM MY FELLOW T.I. FRIENDS. I GUARANTEE THAT YOU'LL ENJOY A FEW GOOD LAUGHS - AT MY OWN EXPENSE!

IN THE MEAN TIME - IF YOU NEED ANY HELP WITH THIS PROGRAM JUST GIVE ME A CALL (412-335-0163) AND I'LL TRY TO HELP.

MICKEY SCHMITT

```

100 REM *****
110 REM *      PROGRAM LISTING FOR A CASSETTE TAPE CATALOG IN T.I. BASIC      *
120 REM *****
130 CALL CLEAR
140 PRINT "CASSETTE TITLE:_____":
150 PRINT "CASSETTE NUMBER:_____":
160 PRINT "CASSETTE SIDE:_____":
170 PRINT "COUNTER READING:_____":
180 PRINT "LANGUAGE USED:_____":
190 PRINT "PERIPHERALS NEEDED:_____":
200 PRINT "PROGRAM NAME:_____":
210 PRINT "PROGRAM DESCRIPTION:_____":
220 GOSUB 10000
230 CALL CLEAR
240 REM TO CATALOG MORE THAN ONE PROGRAM - FOLLOW THE SAME FORMAT AS USED IN
250 REM LINE NUMBERS 170 - 230. CONTINUE USING THIS SAME FORMAT TILL ALL OF
260 REM YOUR PROGRAMS HAVE BEEN CATALOGED.
270 REM CAUTION: AFTER YOUR FINAL ENTRY - REMEMBER TO USE AN "END" STATEMENT
280 REM RIGHT AFTER YOUR FINAL "CALL CLEAR" STATEMENT.
290 REM FOLLOWING THIS FORMAT WILL HELP KEEP ALL OF YOUR PROGRAMMING
300 REM INFORMATION UNIFORM AND EASIER TO FOLLOW ON YOUR MONITOR OR TV SCREEN.
310 END
10000 PRINT "PRESS: ANY KEY TO CONTINUE"
10010 CALL KEY(0,K,S)
10020 IF S=0 THEN 10010
10030 RETURN

```


PROTECTING YOUR COMPUTER SYSTEM AGAINST VOLTAGE TRANSIENTS

Computer systems and other a.c. powered solid-state electronic equipment are susceptible to voltage transients coming through your a.c. wall outlets. These transients are also called "spikes" or "surges" and are caused by electrical storms, turning switches on and off, and by starting appliance motors. A voltage spike can be as high as 6000 volts and lasts less than 100 microseconds. According to West Penn Power Company, these transients are responsible for 50 percent of personal computer malfunctions. It is also possible for a voltage spike to cause permanent damage to your computer system. Your local power company can't supply you computer grade power so you must protect your system yourself.

How can you protect your computer equipment from these transients? By using a simple, inexpensive device called a MOV (metal oxide varistor). A MOV is a bi-directional voltage surge suppressor: it reacts to voltage spikes by absorbing their energy and dissipating the energy as heat. MOV's range in values from 4 to 2800 volts a.c. For normal line voltage of 115-120 VAC, the 130 volt MOV is used.

How does a MOV operate? For line voltages less than 130 VAC, the 130 volt MOV is inactive; it behaves as if it isn't in the circuit. If the voltage climbs above 130 VAC, then the MOV starts to conduct current, clamping the voltage to safe levels in both the positive and negative directions. MOV's react to transients in only a few nanoseconds, making them very fast and very effective for spike protection, whereas fuses and circuit breakers don't react to transients at all.

MOV's are made by General Electric and Panasonic. They can be obtained from most mail order electronic companies and local Radio Shack stores. Radio Shack has two sizes of 130 volt MOV's that are made by Panasonic:

Cat. No. 276-570, 0.6 watt, 35 joules, 2500 amps, \$1.59 each and

Cat. No. 276-568, 1.0 watt, 70 joules, 4000 amps, \$1.69 each.

I would recommend using the 1.0 watt MOV because of its higher energy absorbing characteristics.

The easiest way to connect MOV's to your computer's power source is to wire them into a 3-prong a.c. plug and then insert the plug into the computer system's power strip if you are using one (use the outlet closest to the switch or line cord) or insert the MOV plug into the wall outlet that your computer is connected to. The other method is to wire the MOV's inside the power strip if there is room and the power strip is able to be disassembled. You can't wire them into the TI 99/4A console because the a.c. voltages coming in are low (18 and 8.5 VAC), having been stepped down by the external transformer. You could wire lower value MOV's into the console to match the low a.c. voltages, but then only the console would be protected and not your whole computer system.

Commercial surge protected power strips with 4-8 outlets are available at costs of \$30 to \$100. Radio Shack has a 6 outlet, 15 amp power strip with a lighted on-off switch, circuit breaker, EMI-RFI noise filter, and two MOV's for surge protection for \$29.95, Cat. No. 61-2780. Also, Radio Shack has a 3-prong

plug device with MOV surge protection that can be plugged into an existing power strip or wall outlet for \$7.95, Cat. No. 61-2791. Some commercial devices may only have one MOV installed, in the differential mode (the MOV is wired from the hot wire to the neutral wire). For full protection you need three MOV's: one in the differential mode and two in the common mode (one wired from the hot wire to the ground wire and the other wired from the neutral wire to the ground wire).

Whether you buy a commercial product or make your own surge protector, you should protect your computer system from the problems of electrical transients. For the price of a few dollars you can protect yourself from lost data, glitches and permanent damage.

R. P. Sadusky, West Penn 99'ers

THE ABOVE PROGRAM IS AN EVEN MORE IMPROVED DISK CAT PROGRAM, BUT WAS DESIGNED TO RUN A GEMINI 10X OR THE LIKE. THANKS TO MARK HARMS OF THE KANKAKEE IL. U.G. FOR THIS VERSION THAT HE WROTE IN RESPONSE TO THE ONE I PRINTED IN LAST MONTHS ISSUE OF WPUG NEWSLETTER. LET'S SEE HOW GOOD AND VERSATILE THIS PROGRAM CAN GET!

```

100 : *****CAT PRINT*****
110 : by Warren Agee
120 :
130 : CDS# 11,2550
140 :
150 : This is a very simple program which will print a catalog of
160 : any disk in teeny-tiny print which will fit
170 : on a mailing label (4" x 7 1/8" is best).
180 : CALL SCREEN(15) : GOTO 150 : *****ITSY-BITSY CAT-PR.NT*****
190 : IF A$(127) = "A" THEN GOTO 1 : IF A$(127) = "E" THEN GOTO 2 : IF A$(127) = "P" THEN GOTO 3 : IF A$(127) = "D" THEN GOTO 4 : IF A$(127) = "V" THEN GOTO 5 : IF A$(127) = "I" THEN GOTO 6 : IF A$(127) = "R" THEN GOTO 7 : IF A$(127) = "G" THEN GOTO 8 : IF A$(127) = "L" THEN GOTO 9 : IF A$(127) = "K" THEN GOTO 10 : IF A$(127) = "T" THEN GOTO 11 : IF A$(127) = "Y" THEN GOTO 12 : IF A$(127) = "S" THEN GOTO 13 : IF A$(127) = "P" THEN GOTO 14 : IF A$(127) = "D" THEN GOTO 15 : IF A$(127) = "V" THEN GOTO 16 : IF A$(127) = "I" THEN GOTO 17 : IF A$(127) = "R" THEN GOTO 18 : IF A$(127) = "G" THEN GOTO 19 : IF A$(127) = "L" THEN GOTO 20 : IF A$(127) = "K" THEN GOTO 21 : IF A$(127) = "T" THEN GOTO 22 : IF A$(127) = "Y" THEN GOTO 23 : IF A$(127) = "S" THEN GOTO 24 : IF A$(127) = "P" THEN GOTO 25 : IF A$(127) = "D" THEN GOTO 26 : IF A$(127) = "V" THEN GOTO 27 : IF A$(127) = "I" THEN GOTO 28 : IF A$(127) = "R" THEN GOTO 29 : IF A$(127) = "G" THEN GOTO 30 : IF A$(127) = "L" THEN GOTO 31 : IF A$(127) = "K" THEN GOTO 32 : IF A$(127) = "T" THEN GOTO 33 : IF A$(127) = "Y" THEN GOTO 34 : IF A$(127) = "S" THEN GOTO 35 : IF A$(127) = "P" THEN GOTO 36 : IF A$(127) = "D" THEN GOTO 37 : IF A$(127) = "V" THEN GOTO 38 : IF A$(127) = "I" THEN GOTO 39 : IF A$(127) = "R" THEN GOTO 40 : IF A$(127) = "G" THEN GOTO 41 : IF A$(127) = "L" THEN GOTO 42 : IF A$(127) = "K" THEN GOTO 43 : IF A$(127) = "T" THEN GOTO 44 : IF A$(127) = "Y" THEN GOTO 45 : IF A$(127) = "S" THEN GOTO 46 : IF A$(127) = "P" THEN GOTO 47 : IF A$(127) = "D" THEN GOTO 48 : IF A$(127) = "V" THEN GOTO 49 : IF A$(127) = "I" THEN GOTO 50 : IF A$(127) = "R" THEN GOTO 51 : IF A$(127) = "G" THEN GOTO 52 : IF A$(127) = "L" THEN GOTO 53 : IF A$(127) = "K" THEN GOTO 54 : IF A$(127) = "T" THEN GOTO 55 : IF A$(127) = "Y" THEN GOTO 56 : IF A$(127) = "S" THEN GOTO 57 : IF A$(127) = "P" THEN GOTO 58 : IF A$(127) = "D" THEN GOTO 59 : IF A$(127) = "V" THEN GOTO 60 : IF A$(127) = "I" THEN GOTO 61 : IF A$(127) = "R" THEN GOTO 62 : IF A$(127) = "G" THEN GOTO 63 : IF A$(127) = "L" THEN GOTO 64 : IF A$(127) = "K" THEN GOTO 65 : IF A$(127) = "T" THEN GOTO 66 : IF A$(127) = "Y" THEN GOTO 67 : IF A$(127) = "S" THEN GOTO 68 : IF A$(127) = "P" THEN GOTO 69 : IF A$(127) = "D" THEN GOTO 70 : IF A$(127) = "V" THEN GOTO 71 : IF A$(127) = "I" THEN GOTO 72 : IF A$(127) = "R" THEN GOTO 73 : IF A$(127) = "G" THEN GOTO 74 : IF A$(127) = "L" THEN GOTO 75 : IF A$(127) = "K" THEN GOTO 76 : IF A$(127) = "T" THEN GOTO 77 : IF A$(127) = "Y" THEN GOTO 78 : IF A$(127) = "S" THEN GOTO 79 : IF A$(127) = "P" THEN GOTO 80 : IF A$(127) = "D" THEN GOTO 81 : IF A$(127) = "V" THEN GOTO 82 : IF A$(127) = "I" THEN GOTO 83 : IF A$(127) = "R" THEN GOTO 84 : IF A$(127) = "G" THEN GOTO 85 : IF A$(127) = "L" THEN GOTO 86 : IF A$(127) = "K" THEN GOTO 87 : IF A$(127) = "T" THEN GOTO 88 : IF A$(127) = "Y" THEN GOTO 89 : IF A$(127) = "S" THEN GOTO 90 : IF A$(127) = "P" THEN GOTO 91 : IF A$(127) = "D" THEN GOTO 92 : IF A$(127) = "V" THEN GOTO 93 : IF A$(127) = "I" THEN GOTO 94 : IF A$(127) = "R" THEN GOTO 95 : IF A$(127) = "G" THEN GOTO 96 : IF A$(127) = "L" THEN GOTO 97 : IF A$(127) = "K" THEN GOTO 98 : IF A$(127) = "T" THEN GOTO 99 : IF A$(127) = "Y" THEN GOTO 100 : IF A$(127) = "S" THEN GOTO 101 : IF A$(127) = "P" THEN GOTO 102 : IF A$(127) = "D" THEN GOTO 103 : IF A$(127) = "V" THEN GOTO 104 : IF A$(127) = "I" THEN GOTO 105 : IF A$(127) = "R" THEN GOTO 106 : IF A$(127) = "G" THEN GOTO 107 : IF A$(127) = "L" THEN GOTO 108 : IF A$(127) = "K" THEN GOTO 109 : IF A$(127) = "T" THEN GOTO 110 : IF A$(127) = "Y" THEN GOTO 111 : IF A$(127) = "S" THEN GOTO 112 : IF A$(127) = "P" THEN GOTO 113 : IF A$(127) = "D" THEN GOTO 114 : IF A$(127) = "V" THEN GOTO 115 : IF A$(127) = "I" THEN GOTO 116 : IF A$(127) = "R" THEN GOTO 117 : IF A$(127) = "G" THEN GOTO 118 : IF A$(127) = "L" THEN GOTO 119 : IF A$(127) = "K" THEN GOTO 120 : IF A$(127) = "T" THEN GOTO 121 : IF A$(127) = "Y" THEN GOTO 122 : IF A$(127) = "S" THEN GOTO 123 : IF A$(127) = "P" THEN GOTO 124 : IF A$(127) = "D" THEN GOTO 125 : IF A$(127) = "V" THEN GOTO 126 : IF A$(127) = "I" THEN GOTO 127 : IF A$(127) = "R" THEN GOTO 128 : IF A$(127) = "G" THEN GOTO 129 : IF A$(127) = "L" THEN GOTO 130 : IF A$(127) = "K" THEN GOTO 131 : IF A$(127) = "T" THEN GOTO 132 : IF A$(127) = "Y" THEN GOTO 133 : IF A$(127) = "S" THEN GOTO 134 : IF A$(127) = "P" THEN GOTO 135 : IF A$(127) = "D" THEN GOTO 136 : IF A$(127) = "V" THEN GOTO 137 : IF A$(127) = "I" THEN GOTO 138 : IF A$(127) = "R" THEN GOTO 139 : IF A$(127) = "G" THEN GOTO 140 : IF A$(127) = "L" THEN GOTO 141 : IF A$(127) = "K" THEN GOTO 142 : IF A$(127) = "T" THEN GOTO 143 : IF A$(127) = "Y" THEN GOTO 144 : IF A$(127) = "S" THEN GOTO 145 : IF A$(127) = "P" THEN GOTO 146 : IF A$(127) = "D" THEN GOTO 147 : IF A$(127) = "V" THEN GOTO 148 : IF A$(127) = "I" THEN GOTO 149 : IF A$(127) = "R" THEN GOTO 150 : IF A$(127) = "G" THEN GOTO 151 : IF A$(127) = "L" THEN GOTO 152 : IF A$(127) = "K" THEN GOTO 153 : IF A$(127) = "T" THEN GOTO 154 : IF A$(127) = "Y" THEN GOTO 155 : IF A$(127) = "S" THEN GOTO 156 : IF A$(127) = "P" THEN GOTO 157 : IF A$(127) = "D" THEN GOTO 158 : IF A$(127) = "V" THEN GOTO 159 : IF A$(127) = "I" THEN GOTO 160 : IF A$(127) = "R" THEN GOTO 161 : IF A$(127) = "G" THEN GOTO 162 : IF A$(127) = "L" THEN GOTO 163 : IF A$(127) = "K" THEN GOTO 164 : IF A$(127) = "T" THEN GOTO 165 : IF A$(127) = "Y" THEN GOTO 166 : IF A$(127) = "S" THEN GOTO 167 : IF A$(127) = "P" THEN GOTO 168 : IF A$(127) = "D" THEN GOTO 169 : IF A$(127) = "V" THEN GOTO 170 : IF A$(127) = "I" THEN GOTO 171 : IF A$(127) = "R" THEN GOTO 172 : IF A$(127) = "G" THEN GOTO 173 : IF A$(127) = "L" THEN GOTO 174 : IF A$(127) = "K" THEN GOTO 175 : IF A$(127) = "T" THEN GOTO 176 : IF A$(127) = "Y" THEN GOTO 177 : IF A$(127) = "S" THEN GOTO 178 : IF A$(127) = "P" THEN GOTO 179 : IF A$(127) = "D" THEN GOTO 180 : IF A$(127) = "V" THEN GOTO 181 : IF A$(127) = "I" THEN GOTO 182 : IF A$(127) = "R" THEN GOTO 183 : IF A$(127) = "G" THEN GOTO 184 : IF A$(127) = "L" THEN GOTO 185 : IF A$(127) = "K" THEN GOTO 186 : IF A$(127) = "T" THEN GOTO 187 : IF A$(127) = "Y" THEN GOTO 188 : IF A$(127) = "S" THEN GOTO 189 : IF A$(127) = "P" THEN GOTO 190 : IF A$(127) = "D" THEN GOTO 191 : IF A$(127) = "V" THEN GOTO 192 : IF A$(127) = "I" THEN GOTO 193 : IF A$(127) = "R" THEN GOTO 194 : IF A$(127) = "G" THEN GOTO 195 : IF A$(127) = "L" THEN GOTO 196 : IF A$(127) = "K" THEN GOTO 197 : IF A$(127) = "T" THEN GOTO 198 : IF A$(127) = "Y" THEN GOTO 199 : IF A$(127) = "S" THEN GOTO 200 : IF A$(127) = "P" THEN GOTO 201 : IF A$(127) = "D" THEN GOTO 202 : IF A$(127) = "V" THEN GOTO 203 : IF A$(127) = "I" THEN GOTO 204 : IF A$(127) = "R" THEN GOTO 205 : IF A$(127) = "G" THEN GOTO 206 : IF A$(127) = "L" THEN GOTO 207 : IF A$(127) = "K" THEN GOTO 208 : IF A$(127) = "T" THEN GOTO 209 : IF A$(127) = "Y" THEN GOTO 210 : IF A$(127) = "S" THEN GOTO 211 : IF A$(127) = "P" THEN GOTO 212 : IF A$(127) = "D" THEN GOTO 213 : IF A$(127) = "V" THEN GOTO 214 : IF A$(127) = "I" THEN GOTO 215 : IF A$(127) = "R" THEN GOTO 216 : IF A$(127) = "G" THEN GOTO 217 : IF A$(127) = "L" THEN GOTO 218 : IF A$(127) = "K" THEN GOTO 219 : IF A$(127) = "T" THEN GOTO 220 : IF A$(127) = "Y" THEN GOTO 221 : IF A$(127) = "S" THEN GOTO 222 : IF A$(127) = "P" THEN GOTO 223 : IF A$(127) = "D" THEN GOTO 224 : IF A$(127) = "V" THEN GOTO 225 : IF A$(127) = "I" THEN GOTO 226 : IF A$(127) = "R" THEN GOTO 227 : IF A$(127) = "G" THEN GOTO 228 : IF A$(127) = "L" THEN GOTO 229 : IF A$(127) = "K" THEN GOTO 230 : IF A$(127) = "T" THEN GOTO 231 : IF A$(127) = "Y" THEN GOTO 232 : IF A$(127) = "S" THEN GOTO 233 : IF A$(127) = "P" THEN GOTO 234 : IF A$(127) = "D" THEN GOTO 235 : IF A$(127) = "V" THEN GOTO 236 : IF A$(127) = "I" THEN GOTO 237 : IF A$(127) = "R" THEN GOTO 238 : IF A$(127) = "G" THEN GOTO 239 : IF A$(127) = "L" THEN GOTO 240 : IF A$(127) = "K" THEN GOTO 241 : IF A$(127) = "T" THEN GOTO 242 : IF A$(127) = "Y" THEN GOTO 243 : IF A$(127) = "S" THEN GOTO 244 : IF A$(127) = "P" THEN GOTO 245 : IF A$(127) = "D" THEN GOTO 246 : IF A$(127) = "V" THEN GOTO 247 : IF A$(127) = "I" THEN GOTO 248 : IF A$(127) = "R" THEN GOTO 249 : IF A$(127) = "G" THEN GOTO 250 : IF A$(127) = "L" THEN GOTO 251 : IF A$(127) = "K" THEN GOTO 252 : IF A$(127) = "T" THEN GOTO 253 : IF A$(127) = "Y" THEN GOTO 254 : IF A$(127) = "S" THEN GOTO 255 : IF A$(127) = "P" THEN GOTO 256 : IF A$(127) = "D" THEN GOTO 257 : IF A$(127) = "V" THEN GOTO 258 : IF A$(127) = "I" THEN GOTO 259 : IF A$(127) = "R" THEN GOTO 260 : IF A$(127) = "G" THEN GOTO 261 : IF A$(127) = "L" THEN GOTO 262 : IF A$(127) = "K" THEN GOTO 263 : IF A$(127) = "T" THEN GOTO 264 : IF A$(127) = "Y" THEN GOTO 265 : IF A$(127) = "S" THEN GOTO 266 : IF A$(127) = "P" THEN GOTO 267 : IF A$(127) = "D" THEN GOTO 268 : IF A$(127) = "V" THEN GOTO 269 : IF A$(127) = "I" THEN GOTO 270 : IF A$(127) = "R" THEN GOTO 271 : IF A$(127) = "G" THEN GOTO 272 : IF A$(127) = "L" THEN GOTO 273 : IF A$(127) = "K" THEN GOTO 274 : IF A$(127) = "T" THEN GOTO 275 : IF A$(127) = "Y" THEN GOTO 276 : IF A$(127) = "S" THEN GOTO 277 : IF A$(127) = "P" THEN GOTO 278 : IF A$(127) = "D" THEN GOTO 279 : IF A$(127) = "V" THEN GOTO 280 : IF A$(127) = "I" THEN GOTO 281 : IF A$(127) = "R" THEN GOTO 282 : IF A$(127) = "G" THEN GOTO 283 : IF A$(127) = "L" THEN GOTO 284 : IF A$(127) = "K" THEN GOTO 285 : IF A$(127) = "T" THEN GOTO 286 : IF A$(127) = "Y" THEN GOTO 287 : IF A$(127) = "S" THEN GOTO 288 : IF A$(127) = "P" THEN GOTO 289 : IF A$(127) = "D" THEN GOTO 290 : IF A$(127) = "V" THEN GOTO 291 : IF A$(127) = "I" THEN GOTO 292 : IF A$(127) = "R" THEN GOTO 293 : IF A$(127) = "G" THEN GOTO 294 : IF A$(127) = "L" THEN GOTO 295 : IF A$(127) = "K" THEN GOTO 296 : IF A$(127) = "T" THEN GOTO 297 : IF A$(127) = "Y" THEN GOTO 298 : IF A$(127) = "S" THEN GOTO 299 : IF A$(127) = "P" THEN GOTO 300 : IF A$(127) = "D" THEN GOTO 301 : IF A$(127) = "V" THEN GOTO 302 : IF A$(127) = "I" THEN GOTO 303 : IF A$(127) = "R" THEN GOTO 304 : IF A$(127) = "G" THEN GOTO 305 : IF A$(127) = "L" THEN GOTO 306 : IF A$(127) = "K" THEN GOTO 307 : IF A$(127) = "T" THEN GOTO 308 : IF A$(127) = "Y" THEN GOTO 309 : IF A$(127) = "S" THEN GOTO 310 : IF A$(127) = "P" THEN GOTO 311 : IF A$(127) = "D" THEN GOTO 312 : IF A$(127) = "V" THEN GOTO 313 : IF A$(127) = "I" THEN GOTO 314 : IF A$(127) = "R" THEN GOTO 315 : IF A$(127) = "G" THEN GOTO 316 : IF A$(127) = "L" THEN GOTO 317 : IF A$(127) = "K" THEN GOTO 318 : IF A$(127) = "T" THEN GOTO 319 : IF A$(127) = "Y" THEN GOTO 320 : IF A$(127) = "S" THEN GOTO 321 : IF A$(127) = "P" THEN GOTO 322 : IF A$(127) = "D" THEN GOTO 323 : IF A$(127) = "V" THEN GOTO 324 : IF A$(127) = "I" THEN GOTO 325 : IF A$(127) = "R" THEN GOTO 326 : IF A$(127) = "G" THEN GOTO 327 : IF A$(127) = "L" THEN GOTO 328 : IF A$(127) = "K" THEN GOTO 329 : IF A$(127) = "T" THEN GOTO 330 : IF A$(127) = "Y" THEN GOTO 331 : IF A$(127) = "S" THEN GOTO 332 : IF A$(127) = "P" THEN GOTO 333 : IF A$(127) = "D" THEN GOTO 334 : IF A$(127) = "V" THEN GOTO 335 : IF A$(127) = "I" THEN GOTO 336 : IF A$(127) = "R" THEN GOTO 337 : IF A$(127) = "G" THEN GOTO 338 : IF A$(127) = "L" THEN GOTO 339 : IF A$(127) = "K" THEN GOTO 340 : IF A$(127) = "T" THEN GOTO 341 : IF A$(127) = "Y" THEN GOTO 342 : IF A$(127) = "S" THEN GOTO 343 : IF A$(127) = "P" THEN GOTO 344 : IF A$(127) = "D" THEN GOTO 345 : IF A$(127) = "V" THEN GOTO 346 : IF A$(127) = "I" THEN GOTO 347 : IF A$(127) = "R" THEN GOTO 348 : IF A$(127) = "G" THEN GOTO 349 : IF A$(127) = "L" THEN GOTO 350 : IF A$(127) = "K" THEN GOTO 351 : IF A$(127) = "T" THEN GOTO 352 : IF A$(127) = "Y" THEN GOTO 353 : IF A$(127) = "S" THEN GOTO 354 : IF A$(127) = "P" THEN GOTO 355 : IF A$(127) = "D" THEN GOTO 356 : IF A$(127) = "V" THEN GOTO 357 : IF A$(127) = "I" THEN GOTO 358 : IF A$(127) = "R" THEN GOTO 359 : IF A$(127) = "G" THEN GOTO 360 : IF A$(127) = "L" THEN GOTO 361 : IF A$(127) = "K" THEN GOTO 362 : IF A$(127) = "T" THEN GOTO 363 : IF A$(127) = "Y" THEN GOTO 364 : IF A$(127) = "S" THEN GOTO 365 : IF A$(127) = "P" THEN GOTO 366 : IF A$(127) = "D" THEN GOTO 367 : IF A$(127) = "V" THEN GOTO 368 : IF A$(127) = "I" THEN GOTO 369 : IF A$(127) = "R" THEN GOTO 370 : IF A$(127) = "G" THEN GOTO 371 : IF A$(127) = "L" THEN GOTO 372 : IF A$(127) = "K" THEN GOTO 373 : IF A$(127) = "T" THEN GOTO 374 : IF A$(127) = "Y" THEN GOTO 375 : IF A$(127) = "S" THEN GOTO 376 : IF A$(127) = "P" THEN GOTO 377 : IF A$(127) = "D" THEN GOTO 378 : IF A$(127) = "V" THEN GOTO 379 : IF A$(127) = "I" THEN GOTO 380 : IF A$(127) = "R" THEN GOTO 381 : IF A$(127) = "G" THEN GOTO 382 : IF A$(127) = "L" THEN GOTO 383 : IF A$(127) = "K" THEN GOTO 384 : IF A$(127) = "T" THEN GOTO 385 : IF A$(127) = "Y" THEN GOTO 386 : IF A$(127) = "S" THEN GOTO 387 : IF A$(127) = "P" THEN GOTO 388 : IF A$(127) = "D" THEN GOTO 389 : IF A$(127) = "V" THEN GOTO 390 : IF A$(127) = "I" THEN GOTO 391 : IF A$(127) = "R" THEN GOTO 392 : IF A$(127) = "G" THEN GOTO 393 : IF A$(127) = "L" THEN GOTO 394 : IF A$(127) = "K" THEN GOTO 395 : IF A$(127) = "T" THEN GOTO 396 : IF A$(127) = "Y" THEN GOTO 397 : IF A$(127) = "S" THEN GOTO 398 : IF A$(127) = "P" THEN GOTO 399 : IF A$(127) = "D" THEN GOTO 400 : IF A$(127) = "V" THEN GOTO 401 : IF A$(127) = "I" THEN GOTO 402 : IF A$(127) = "R" THEN GOTO 403 : IF A$(127) = "G" THEN GOTO 404 : IF A$(127) = "L" THEN GOTO 405 : IF A$(127) = "K" THEN GOTO 406 : IF A$(127) = "T" THEN GOTO 407 : IF A$(127) = "Y" THEN GOTO 408 : IF A$(127) = "S" THEN GOTO 409 : IF A$(127) = "P" THEN GOTO 410 : IF A$(127) = "D" THEN GOTO 411 : IF A$(127) = "V" THEN GOTO 412 : IF A$(127) = "I" THEN GOTO 413 : IF A$(127) = "R" THEN GOTO 414 : IF A$(127) = "G" THEN GOTO 415 : IF A$(127) = "L" THEN GOTO 416 : IF A$(127) = "K" THEN GOTO 417 : IF A$(127) = "T" THEN GOTO 418 : IF A$(127) = "Y" THEN GOTO 419 : IF A$(127) = "S" THEN GOTO 420 : IF A$(127) = "P" THEN GOTO 421 : IF A$(127) = "D" THEN GOTO 422 : IF A$(127) = "V" THEN GOTO 423 : IF A$(127) = "I" THEN GOTO 424 : IF A$(127) = "R" THEN GOTO 425 : IF A$(127) = "G" THEN GOTO 426 : IF A$(127) = "L" THEN GOTO 427 : IF A$(127) = "K" THEN GOTO 428 : IF A$(127) = "T" THEN GOTO 429 : IF A$(127) = "Y" THEN GOTO 430 : IF A$(127) = "S" THEN GOTO 431 : IF A$(127) = "P" THEN GOTO 432 : IF A$(127) = "D" THEN GOTO 433 : IF A$(127) = "V" THEN GOTO 434 : IF A$(127) = "I" THEN GOTO 435 : IF A$(127) = "R" THEN GOTO 436 : IF A$(127) = "G" THEN GOTO 437 : IF A$(127) = "L" THEN GOTO 438 : IF A$(127) = "K" THEN GOTO 439 : IF A$(127) = "T" THEN GOTO 440 : IF A$(127) = "Y" THEN GOTO 441 : IF A$(127) = "S" THEN GOTO 442 : IF A$(127) = "P" THEN GOTO 443 : IF A$(127) = "D" THEN GOTO 444 : IF A$(127) = "V" THEN GOTO 445 : IF A$(127) = "I" THEN GOTO 446 : IF A$(127) = "R" THEN GOTO 447 : IF A$(127) = "G" THEN GOTO 448 : IF A$(127) = "L" THEN GOTO 449 : IF A$(127) = "K" THEN GOTO 450 : IF A$(127) = "T" THEN GOTO 451 : IF A$(127) = "Y" THEN GOTO 452 : IF A$(127) = "S" THEN GOTO 453 : IF A$(127) = "P" THEN GOTO 454 : IF A$(127) = "D" THEN GOTO 455 : IF A$(127) = "V" THEN GOTO 456 : IF A$(127) = "I" THEN GOTO 457 : IF A$(127) = "R" THEN GOTO 458 : IF A$(127) = "G" THEN GOTO 459 : IF A$(127) = "L" THEN GOTO 460 : IF A$(127) = "K" THEN GOTO 461 : IF A$(127) = "T" THEN GOTO 462 : IF A$(127) = "Y" THEN GOTO 463 : IF A$(127) = "S" THEN GOTO 464 : IF A$(127) = "P" THEN GOTO 465 : IF A$(127) = "D" THEN GOTO 466 : IF A$(127) = "V" THEN GOTO 467 : IF A$(127) = "I" THEN GOTO 468 : IF A$(127) = "R" THEN GOTO 469 : IF A$(127) = "G" THEN GOTO 470 : IF A$(127) = "L" THEN GOTO 471 : IF A$(127) = "K" THEN GOTO 472 : IF A$(127) = "T" THEN GOTO 473 : IF A$(127) = "Y" THEN GOTO 474 : IF A$(127) = "S" THEN GOTO 475 : IF A$(127) = "P" THEN GOTO 476 : IF A$(127) = "D" THEN GOTO 477 : IF A$(127) = "V" THEN GOTO 478 : IF A$(127) = "I" THEN GOTO 479 : IF A$(127) = "R" THEN GOTO 480 : IF A$(127) = "G" THEN GOTO 481 : IF A$(127) = "L" THEN GOTO 482 : IF A$(127) = "K" THEN GOTO 483 : IF A$(127) = "T" THEN GOTO 484 : IF A$(127) = "Y" THEN GOTO 485 : IF A$(127) = "S" THEN GOTO 486 : IF A$(127) = "P" THEN GOTO 487 : IF A$(127) = "D" THEN GOTO 488 : IF A$(127) = "V" THEN GOTO 489 : IF A$(127) = "I" THEN GOTO 490 : IF A$(127) = "R" THEN GOTO 491 : IF A$(127) = "G" THEN GOTO 492 : IF A$(127) = "L" THEN GOTO 493 : IF A$(127) = "K" THEN GOTO 494 : IF A$(127) = "T" THEN GOTO 495 : IF A$(127) = "Y" THEN GOTO 496 : IF A$(127) = "S" THEN GOTO 497 : IF A$(127) = "P" THEN GOTO 498 : IF A$(127) = "D" THEN GOTO 499 : IF A$(127) = "V" THEN GOTO 500 : IF A$(127) = "I" THEN GOTO 501 : IF A$(127) = "R" THEN GOTO 502 : IF A$(127) = "G" THEN GOTO 503 : IF A$(127) = "L" THEN GOTO 504 : IF A$(127) = "K" THEN GOTO 505 : IF A$(127) = "T" THEN GOTO 506 : IF A$(127) = "Y" THEN GOTO 507 : IF A$(127) = "S" THEN GOTO 508 : IF A$(127) = "P" THEN GOTO 509 : IF A$(127) = "D" THEN GOTO 510 : IF A$(127) = "V" THEN GOTO 511 : IF A$(127) = "I" THEN GOTO 512 : IF A$(127) = "R" THEN GOTO 513 : IF A$(127) = "G" THEN GOTO 514 : IF A$(127) = "L" THEN GOTO 515 : IF A$(127) = "K" THEN GOTO 516 : IF A$(127) = "T" THEN GOTO 517 : IF A$(127) = "Y" THEN GOTO 518 : IF A$(127) = "S" THEN GOTO 519 : IF A$(127) = "P" THEN GOTO 520 : IF A$(127) = "D" THEN GOTO 521 : IF A$(127) = "V" THEN GOTO 522 : IF A$(127) = "I" THEN GOTO 523 : IF A$(127) = "R" THEN GOTO 524 : IF A$(127) = "G" THEN GOTO 525 : IF A$(127) = "L" THEN GOTO 526 : IF A$(127) = "K" THEN GOTO 527 : IF A$(127) = "T" THEN GOTO 528 : IF A$(127) = "Y" THEN GOTO 529 : IF A$(127) = "S" THEN GOTO 530 : IF A$(127) = "P" THEN GOTO 531 : IF A$(127) = "D" THEN GOTO 532 : IF A$(127) = "V" THEN GOTO 533 : IF A$(127) = "I" THEN GOTO 534 : IF A$(127) = "R" THEN GOTO 535 : IF A$(127) = "G" THEN GOTO 536 : IF A$(127) = "L" THEN GOTO 537 : IF A$(127) = "K" THEN GOTO 538 : IF A$(127) = "T" THEN GOTO 539 : IF A$(127) = "Y" THEN GOTO 540 : IF A$(127) = "S" THEN GOTO 541 : IF A$(127) = "P" THEN GOTO 542 : IF A$(127) = "D" THEN GOTO 543 : IF A$(127) = "V" THEN GOTO 544 : IF A$(127) = "I" THEN GOTO 545 : IF A$(127) = "R" THEN GOTO 546 : IF A$(127) = "G" THEN GOTO 547 : IF A$(127) = "L" THEN GOTO 548 : IF A$(127) = "K" THEN GOTO 549 : IF A$(127) = "T" THEN GOTO 550 : IF A$(127) = "Y" THEN GOTO 551 : IF A$(127) = "S" THEN GOTO 552 : IF A$(127) = "P" THEN GOTO 553 : IF A$(127) = "D" THEN GOTO 554 : IF A$(127) = "V" THEN GOTO 555 : IF A$(127) = "I" THEN GOTO 556 : IF A$(127) = "R" THEN GOTO 557 : IF A$(127) = "G" THEN GOTO 558 : IF A$(127) = "L" THEN GOTO 559 : IF A$(127) = "K" THEN GOTO 560 : IF A$(127) = "T" THEN GOTO 561 : IF A$(127) = "Y" THEN GOTO 562 : IF A$(127) = "S" THEN GOTO 563 : IF A$(127) = "P" THEN GOTO 564 : IF A$(127) = "D" THEN GOTO 565 : IF A$(127) = "V" THEN GOTO 566 : IF A$(127) = "I" THEN GOTO 567
```

WOULD YOU LIKE TO FIX THE "SCORE" IN PINBALL ?

by JOHN WILLFORTH WPUG

HAVE YOU EVER WANTED TO FIX THE SCORE IN THE PINBALL . I'M NOT GOING TO LIST ALL THE REASONS THAT YOU MIGHT WANT TO DO THIS, BUT THERE **WILL** PROBABLY COME A TIME WHEN YOU WOULD LIKE TO DO SO. WELL I THOUGHT THAT I WOULD MAKE IT EASY FOR YOU. ALL YOU NEED IS A DISK EDITOR. YOU MAY USE "DISK SURGEON", "DISK FIXER", "DISCO", OR EVEN THE FORTH EDITOR. THE LATTER WILL REQUIRE MORE INFORMATION THAN I'M GOING TO PROVIDE HERE.

FIRST LOAD YOUR DISK EDITOR, AND THEN REMOVE THE DISK FROM THE DRIVE. BEFOR YOU CONTINUE, I THINK THAT YOU SHOULD REALIZE THAT IF YOU ARE NOT WORKING WITH A "**BACK-UP**" DISK, YOU MAY NEVER PLAY **PINBALL** , OR **ANYTHING** THAT IS ON THAT DISK AGAIN! SO FOR YOUR OWN SAKE, PLEASE **BACKUP** FIRST!

AFTER THE EDITOR IS LOADED, AND THE DISK CONTAINING **PINBALL** (WITH THE FILE **PBSCORE** ALSO PRESENT ON THE SAME DISK) IS INSERTED IN THE SELECTED DRIVE, BEGIN TO VIEW THE SECTORS, STARTING AT <1, AND CONTINUING UNTIL YOU SEE THE FILE "**PBSCORE** " DISPLAYED OR PRINTED IN THE INTERPRETED FIELD. NOW LOOK AT THIS SECTOR, PAYING PARTICULAR ATTENTION TO BYTE-PAIR <1C. THE CONTENTS OF THIS ADDRESS IS THE ADDRESS (SECTOR NUMBER) OF THE FIRST SECTOR IN THE FILE CALLED "**PBSCORE** " .

USING THE EDITOR, LOOK AT THE ADDRESS (SECTOR NUMBER) CALLED OUT FOR IN <1C. THE EXAMPLE ABOVE, SHOWS FIRST THAT ON MY DISK, **PBSCORE** IS FOUND TO START AT SECTOR NUMBER <75. IF YOU ARE LOOKING AT THE INTERPRETED SECTOR, YOU WILL SEE THE DATA THAT WILL REPRESENT THE SCORE AND THE NAME (INITIALS) OF THE PLAYER, AND THIS WILL BE SEEN TO REPEAT FOR A TOTAL OF TEN REPETITIONS. IF YOU HAD WRITTEN DOWN THE SCORES AND THE INITIALS BEFORE THIS EDITING, YOU WOULD EASILY SEE THE FORMAT HERE. THE SCORE IN THE ABOVE EXAMPLE, SHOWS A SCORE OF 67,300 FOR JFW. THE LEAST SIGNIFICANT TWO ZEROS ARE INSERTED BY THE PROGRAM, AND THERE WERE NO MORE PLAYERS.

USING THE PROCEEDURES THAT APPLY TO YOUR PARTICULAR DISK EDITOR, YOU CAN NOW CHANGE THE DATA IN THESE LOCATIONS TO MATCH WHATEVER YOU DESIRE. I HOPE THAT THIS SHORT ARTICLE WILL HELP SOME OF YOU TO ENJOY THE **PINBALL** GAME EVEN MORE. IMAGINE, YOU WITH A SCORE OF 93,673,428, AND NOT EVEN BREAKING A SWEAT!!

```
DISK SURGEON 99-----AMERISOFT INTERNATIONAL-----SECTOR 0008
>0 >2 >4 >6 >8 >A >C >E INTERPRETATION
>00: 5042 5143 4F52 4520 2020 0000 0000 0000 PBSCORE *****
>10: 6400 0000 0000 0000 0000 0000 0000 0000 *****
>20: 0000 0000 0000 0000 0000 0000 0000 0000 *****
>30: 0000 0000 0000 0000 0000 0000 0000 0000 *****
>40: 0000 0000 0000 0000 0000 0000 0000 0000 *****
>50: 0000 0000 0000 0000 0000 0000 0000 0000 *****
```

```
DISK SURGEON 99-----AMERISOFT INTERNATIONAL-----SECTOR 0075
>0 >2 >4 >6 >8 >A >C >E INTERPRETATION
>00: 0000 0000 3733 4A45 5720 0000 0000 0000 000673JFW 000000
>10: 0000 0000 0000 0000 0000 0000 0000 0000 000000 00
>20: 0000 0000 0000 0000 0000 0000 0000 0000 000000 000000
>30: 0000 0000 0000 0000 0000 0000 0000 0000 000000 000000
>40: 0000 0000 0000 0000 0000 0000 0000 0000 000000 000000
>50: 0000 0000 0000 0000 0000 0000 0000 0000 000000 000000
```

```
NO. OF ONE IND. *** DISK FIXER V1.0 ** SECTOR DUMP SECTOR ADDRESS 0008
425 = 0 1 2 3 4 5 6 7 8 9 A B C D E F INTERPRETED
```

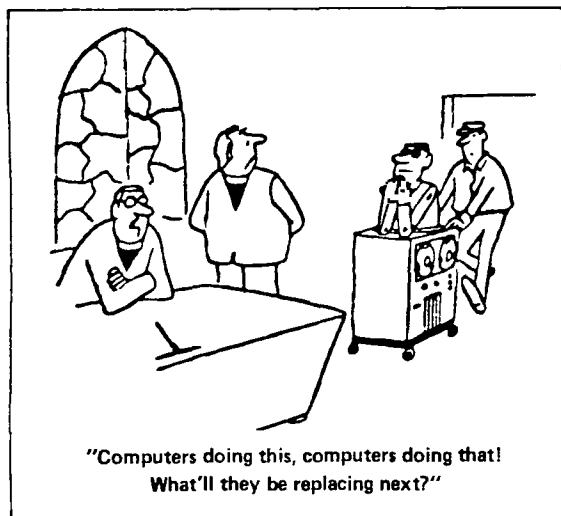
```
0000 = 5042 5143 4F52 4520 2020 0000 0000 0000 PBSCORE *****
0010 = 6400 0000 0000 0000 0000 0000 0000 0000 *****
0020 = 0000 0000 0000 0000 0000 0000 0000 0000 *****
0030 = 0000 0000 0000 0000 0000 0000 0000 0000 *****
0040 = 0000 0000 0000 0000 0000 0000 0000 0000 *****
0050 = 0000 0000 0000 0000 0000 0000 0000 0000 *****
```

```
NO. OF ONE IND. *** DISK FIXER V1.0 ** SECTOR DUMP SECTOR ADDRESS 0075
425 = 0 1 2 3 4 5 6 7 8 9 A B C D E F INTERPRETED
```

```
0000 = 0000 0000 3733 4A45 5720 0000 0000 0000 000673JFW 000000
0010 = 0000 0000 0000 0000 0000 0000 0000 0000 000000 00
0020 = 0000 0000 0000 0000 0000 0000 0000 0000 000000 000000
0030 = 0000 0000 0000 0000 0000 0000 0000 0000 000000 000000
0040 = 0000 0000 0000 0000 0000 0000 0000 0000 000000 000000
0050 = 0000 0000 0000 0000 0000 0000 0000 0000 000000 000000
```

THE NEXT WPUG MEETING

The next WPUG meeting will be held on MONDAY, AUGUST the 18th, at 7:00 P.M. The location will be the **NORWIN YMCA**, located in the OLD NIKE SITE, near NORTH IRWIN. If you need directions please call anyone of the officers, or members. my number is (412) 527-6656, Scott Coleman's number is (412) 271-6283. If you can't make it to this one, we have our meetings on the third Monday of every month, at 7:00 . Membership is \$15.00 per year for a full family membership, and \$10.00 per year, for an associate membership (newsletter only). The term of both memberships is from Jan. 1 of a year to Jan. 1 of the next year. If you join after July 1, the dues are half.



IF YOU HAVE THE 32K MEMORY EXPANSION, YOU CAN WRITE AN XB PROGRAM WHICH IS TOO LARGE TO STORE IN THE USUAL FORMAT. XB WILL STORE THESE LARGE PROGRAMS IN AN "INTERNAL VARIABLE 254" FILE. THE USUAL "SAVE" AND "OLD" COMMANDS ARE USED TO STORE AND LOAD THESE PROGRAMS.

THE THIRD FORM USED BY XB IS THE "MERGE FORMAT" STORED IN A "DISPLAY VARIABLE 163" FILE. THIS FORM IS CREATED WHEN THE "MERGE" OPTION IS SPECIFIED ON THE "SAVE" COMMAND, AND IS LOADED BY THE XB "MERGE" COMMAND. THE BEAUTY OF MERGE FORMAT IS THAT WHEN IT IS LOADED IT DOES NOT NECESSARILY OVERWRITE THE PROGRAM IN MEMORY. THE "MERGE" COMMAND DOES JUST THAT - IT MERGES THE NEW PROGRAM (OR PROGRAM SEGMENT) WITH THE PROGRAM IN MEMORY ACCORDING TO THE LINE NUMBERS.

NOW, WE GET TO THE GOOD STUFF, ASSEMBLER LANGUAGE PROGRAMS. THERE ARE THREE FORMS FOR AN ASSEMBLER PROGRAM: TAGGED OBJECT, COMPRESSED TAGGED OBJECT AND MEMORY IMAGE.

TAGGED OBJECT IS STORED IN A "DISPLAY FIXED 80" FILE ON DISK ONLY. ALL PROGRAM DATA IS IN HEXADECIMAL SO THAT IT CAN BE EDITED BY THE E/A EDITOR. TAGGED OBJECT CAN BE LOADED VIA "CALL LOAD" IN XB, OPTION 3 ON THE E/A MENU, OPTION 1 ON THE MM MENU OR BY "CALL LOAD" IN TI BASIC WHEN EITHER THE E/A OR MM MODULE IS USED. THE PROGRAM CAN BE "ABSOLUTE" OR "RELOCATABLE". AN ABSOLUTE PROGRAM MUST ALWAYS BE LOADED AT THE SAME PLACE IN MEMORY. A RELOCATABLE PROGRAM CAN BE LOADED ANY PLACE IN MEMORY. A TAGGED OBJECT PROGRAM MAY HAVE REFERENCES TO OTHER PROGRAMS OR SUBROUTINES. THE LOADER WILL RESOLVE THESE EXTERNAL REFERENCES, EXCEPT FOR THE XB LOADER.

COMPRESSED TAGGED OBJECT IS VERY NEARLY THE SAME AS TAGGED OBJECT EXCEPT THAT THE PROGRAM DATA IS STORED AS BYTES RATHER THAN AS HEXADECIMAL DIGITS. COMPRESSED TAGGED OBJECT LOADS FASTER THAN REGULAR TAGGED OBJECT AS YOU WOULD EXPECT. THE XB LOADER CANNOT LOAD COMPRESSED OBJECT.

TAGGED OBJECT, IN EITHER FORM, IS PRODUCED BY THE ASSEMBLER WHEN IT ASSEMBLES A SOURCE PROGRAM.

THE "MEMORY IMAGE" FORM OF ASSEMBLER PROGRAMS IS THE MOST COMPACT AND THE FASTEST LOADING. IT CAN BE STORED ON CASSETTE OR DISK. IT IS IDENTIFIED AS "PROGRAM" IN THE DISK CATALOG (JUST LIKE A BASIC PROGRAM). MEMORY IMAGE PROGRAMS CAN BE LOADED BY OPTION 5 ON THE E/A MENU OR OPTION 3 ON THE TIWRITER MENU (AND I ASSUME, BY MULTIPLAN, ALTHOUGH I HAVE NEVER TRIED SINCE I DON'T HAVE MULTIPLAN). IT SHOULD BE NOTED THAT THERE IS ONE SLIGHT BUT IMPORTANT DIFFERENCE BETWEEN HOW THE E/A CALLS A MEMORY IMAGE PROGRAM AND HOW TIWRITER DOES. TIWRITER BLANKS THE SCREEN JUST BEFORE CALLING THE PROGRAM AND THE E/A DOES NOT. THIS MEANS THE PROGRAM

MUST TURN THE SCREEN BACK ON OR NOTHING WILL SHOW. MEMORY IMAGE PROGRAMS ARE CREATED BY A UTILITY PROGRAM (ONE IS PROVIDED ON THE E/A DISK).

A PROGRAM FILE, CONTAINING AN ASSEMBLER MEMORY IMAGE OR A BASIC PROGRAM, CAN BE READ OR WRITTEN TO ANY INPUT/OUTPUT DEVICE WITH A SINGLE I/O OPERATION. THIS IS ONE OF THE REASONS THEY LOAD SO QUICKLY.

THERE IS A RESTRICTION ON THE SIZE OF AN ASSEMBLER MEMORY IMAGE PROGRAM OF 2400 BYTES (9216). HOWEVER, THE E/A AND TIWRITER MODULES WILL LOAD MULTIPLE MEMORY IMAGE FILES TO MAKE A PROGRAM OF ANY SIZE. THEY USE THE CONVENTION THAT THE FILE NAME OF THE SECOND AND THE FOLLOWING FILES IS OBTAINED BY INCREMENTING THE LAST DIGIT OR LETTER OF THE PREVIOUS FILE NAME. FOR EXAMPLE, THE TIWRITER EDITOR CONSISTS OF TWO MEMORY IMAGE FILES: EDITA1 AND EDITA2. AS A MATTER OF INTEREST, THE ADVENTURE, TUNNELS OF DOOM, PERSONAL RECORD KEEPING, STATISTICS AND PERSONAL REPORT GENERATOR, MODULES USE A MEMORY IMAGE OR "PROGRAM" FILE FOR THEIR DATA BASES. THE FACT THAT MEMORY IMAGES CAN BE SAVED OR LOADED WITH A SINGLE I/O OPERATION MAKES THEM ATTRACTIVE FOR SUCH USES.

A LOT OF THE ASSEMBLER LANGUAGE GAMES THAT ARE CIRCULATING AROUND ARE IN MEMORY IMAGE FORMAT SO LET'S LOOK CLOSER AT THEM. ASSEMBLER MEMORY IMAGE FILES HAVE A THREE WORD HEADER FOLLOWED BY THE DATA TO BE PLACED IN MEMORY. THE THREE HEADER WORDS ARE:

(1) THIS WORD IS A "FLAG". IF IT IS NOT ZERO (I.E. FFFF) THEN THIS FILE IS NOT THE LAST IN A MULTI-FILE PROGRAM. FOR EXAMPLE, THE FLAG WORD FOR EDITA1 IS FFFF INDICATING THAT THERE IS ANOTHER FILE CALLED EDITA2; THE FLAG WORD IN THE EDITA2 FILE IS 0000 INDICATING IT IS THE LAST FILE AND THERE IS NO EDITA3.

(2) THIS WORD IS THE LENGTH OF THE MEMORY IMAGE IN BYTES, INCLUDING THE SIX BYTE HEADER.

(3) THIS WORD IS THE CPU MEMORY ADDRESS WHERE THE MEMORY IMAGE IS TO BE LOADED.

EXECUTION OF A MEMORY IMAGE PROGRAM ALWAYS BEGINS AT THE FIRST BYTE OF THE FIRST SEGMENT LOADED.

FINALLY, THE SEVENTH FORM FOR PROGRAMS. THIS FORM IS CREATED AND LOADED BY "EASY BUG" OF THE MINI MEMORY MODULE. IT CAN BE WRITTEN ONLY TO CASSETTE AND IS A MEMORY IMAGE, BUT IS SLIGHTLY DIFFERENT FROM THE E/A MEMORY IMAGE FILE. THE EASY BUG MEMORY IMAGE PROGRAM CAN CONSIST OF ONLY ONE SEGMENT. THE HEADER ON THE EASY BUG FORMAT IS TWO WORDS, AS FOLLOWS:

(1) THIS WORD IS THE CPU MEMORY ADDRESS AT WHICH THE MEMORY IMAGE IS TO BE LOADED.

(2) THIS WORD IS THE LENGTH OF THE MEMORY DATA, NOT INCLUDING THE FOUR HEADER BYTES.

```
*****
: FILE TYPE : CONTENTS : MODULE : DSK : CS :
:*****:
: PROGRAM : BASIC PROGRAM : CONSOLE : YES : YES :
: PROGRAM : BASIC PROGRAM : XB : YES : YES :
: INTERNAL V 254: BASIC PROGRAM : XB : YES : NO :
: DISPLAY V 163: MERGE PROGRAM : XB : YES : NO :
:*****:
: DISPLAY F 80 : TAGGED OBJECT : XB : YES : NO :
: DISPLAY F 80 : TAGGED OBJECT : E/A : YES : NO :
: DISPLAY F 80 : TAGGED OBJECT : MM : YES : NO :
: DISPLAY F 80 : COMPRESSED OBJECT: E/A : YES : NO :
: DISPLAY F 80 : COMPRESSED OBJECT: MM : YES : NO :
: PROGRAM : E/A MEMORY IMAGE : E/A : YES : YES :
: PROGRAM : E/A MEMORY IMAGE : TIW : YES : YES :
: PROGRAM : MM MEMORY IMAGE : MM : NO : YES :
*****
```

THE MUSIC CORNER

by Jeff Gatlin

One of the things I have always thought was missing from the 4A's music capability was the lower range. The 4A gives us an almost infinite upper range, as far as we can hear, but its lower range stops at the frequency of 110. After listening to dozens of music programs, I noticed that some of them seemed to defy reason and belt out bass notes below what I call low "A" (the lowest note played by the 4A without magic). Curiosity and a greedy desire to hold this magic secret let me to examine these programs. I found that to create these magic tones, one must make use of a mysterious fourth voice which our beloved Extended BASIC manual describes as "noise". The "noise" is created by using a negative value between 1 and 8 in place of the note's frequency.

```
10 CALL SOUND(1000,-4,0)
```

This statement creates a mildly interesting noise if you have never heard it before. After hearing it once, no secrets were revealed. So what is the secret? Once again, our Extended BASIC manual sheds light upon us. It says, "-4 Periodic Noise that varies with the frequency of the third tone specified...." Ah ha! The secret is revealed! To create magic bass notes, one must use ALL THREE voices PLUS a -4 noise.

```
10 CALL SOUND(1000,330,0,392,0,523,0,-4,0)
```

Wala! But wait, something is amiss. The bass note created does not match the three voice chord. Experimentation revealed that if you use the note C (523) in the third voice, the bass note played is C# (sharp). Why? It doesn't really matter, at least to me. Below is a chart I made up of the frequencies to use in the third voice, to create the desired bass note.

PLENTY LOW: REALLY LOW:

1661 -- A (same as A110)	831 -- A (one octave below 110)
1568 -- G#,Ab	784 -- G#,Ab
1480 -- G	740 -- G
1397 -- F#,Gb	698 -- F#,Gb
1319 -- F	659 -- F
1245 -- E	622 -- E
1175 -- D#,Eb	587 -- D#,Eb
1109 -- D	554 -- D
1047 -- C#,Db	523 -- C#,Db
988 -- C	494 -- C
932 -- B	466 -- B
880 -- A#,Bb	440 -- A#,Bb
831 -- A (one octave below A110)	415 -- A (two octaves below A110)

The bad news is this does not give you full control of FOUR voices, although if you're tricky enough, you can make use of four note chords. The reason is that the third tone and the noise work together as one tone. In almost every example I have come across, the third tone has a volume of 30, which makes it inaudible. However, this does not affect the volume of the bass note. It's volume is controlled by the value after -4.

With this basic information, you should be able to experiment using bass notes on your own. I hope it is helpful.

SHADED

VELVET

THIS PROGRAM USES THE MYSTERIOUS FOURTH VOICE OF THE 4A. REMEMBER! THERE IS ALWAYS AN INTERVAL OF THREE OCTAVES AND A MAJOR SEVENTH BETWEEN THE BASS AND THE THIRD VOICE. ENCE!!

10 ! "Shaded Velvet"

15 !An Original Composition

20 ! By Jeff Gatlin

25 !Copyright June 1986

30 CALL CLEAR :: CALL SCREEN (2):: FOR T=1 TO 14 :: CALL COLOR(T,15,2):: NEXT T

32 PRINT "This program demonstrates the ability to play four voices instead of three."

35 DISPLAY AT(5,5):"SHADED VELVET" :: DISPLAY AT(6,5):"An Original Composition"

40 DISPLAY AT(10,5):"By Jeff Gatlin" :: DISPLAY AT(12,5):"Copyright June 1986"

50 DIM A(60),B(60),C(60):: P=1 :: V=8

60 FOR R=1 TO 60 :: READ A(R) :: NEXT R

70 FOR R=1 TO 60 :: READ B(R) :: NEXT R

80 FOR R=1 TO 60 :: READ C(R) :: NEXT R

100 FOR R=1 TO 3 :: CALL SOUND(-999,A(R),V,B(R),V,C(R),V,-4,V-2):: P=P^1000000 :: P=P^1000000 :: V=V-2 :: NEXT R

110 FOR R=4 TO 8 :: CALL SOUND(-999,A(R),V,B(R),V,C(R),V,-4,V-2):: P=P^1000000 :: P=P^1000000 :: V=V+2 :: NEXT R

119 V=8

120 FOR R=9 TO 11 :: CALL SOUND(-999,A(R),V,B(R),V,C(R),V,-4,V-2):: P=P^1000000 :: P=P^1000000 :: V=V-2 :: NEXT R

130 FOR R=12 TO 16 :: CALL SOUND(-999,A(R),V,B(R),V,C(R),V,-4,V-2):: P=P^1000000 :: P=P^1000000 :: V=V+2 :: NEXT R

139 V=8

140 FOR R=17 TO 19 :: CALL SOUND(-999,A(R),V,B(R),V,C(R),V,-4,V-2):: P=P^1000000 :: P=P^1000000 :: V=V-2 :: NEXT R

150 FOR R=20 TO 24 :: CALL SOUND(-999,A(R),V,B(R),V,C(R),V,-4,V-2):: P=P^1000000 :: P=P^1000000 :: V=V+2 :: NEXT R

159 V=8

160 FOR R=25 TO 27 :: CALL SOUND(-999,A(R),V,B(R),V,C(R),V,-4,V-2):: P=P^1000000 :: P=P^1000000 :: V=V-2 :: NEXT R

170 FOR R=28 TO 32 :: CALL SOUND(-999,A(R),V,B(R),V,C(R),V,-4,V-2):: P=P^1000000 :: P=P^1000000 :: V=V+2 :: NEXT R

179 V=8

180 FOR R=33 TO 35 :: CALL SOUND(-999,A(R),V,B(R),V,C(R),V,-4,V-2):: P=P^1000000 :: P=P^1000000 :: V=V-2 :: NEXT R

190 FOR R=36 TO 40 :: CALL SOUND(-999,A(R),V,B(R),V,C(R),V,-4,V-2):: P=P^1000000 :: P=P^1000000 :: V=V+2 :: NEXT R

199 V=8

200 FOR R=41 TO 43 :: CALL SOUND(-999,A(R),V,B(R),V,C(R),V,-4,V-2):: P=P^1000000 :: P=P^1000000 :: V=V-2 :: NEXT R

210 FOR R=44 TO 48 :: CALL SOUND(-999,A(R),V,B(R),V,C(R),V,-4,V-2):: P=P^1000000 :: P=P^1000000 :: V=V+2 :: NEXT R

219 V=8

220 FOR R=49 TO 51 :: CALL SOUND(-999,A(R),V,B(R),V,C(R),V,-4,V-2):: P=P^1000000 :: P=P^1000000 :: V=V-2 :: NEXT R

230 FOR R=52 TO 56 :: CALL SOUND(-999,A(R),V,B(R),V,C(R),V,-4,V-2):: P=P^1000000 :: P=P^1000000 :: V=V+2 :: NEXT R

239 V=8

240 FOR R=57 TO 59 :: CALL SOUND(-999,A(R),V,B(R),V,C(R),V,-4,V-2):: P=P^1000000 :: P=P^1000000 :: V=V-2 :: NEXT R

245 R=60 :: FOR RP=1 TO 2 :: CALL SOUND(-999,A(R),V,B(R),V,C(R),V,-4,V-2):: NEXT RP

247 FOR RP=1 TO 8 :: CALL SOUND(-999,A(R),V,B(R),V,C(R),V,-4,V-2):: P=P^1000000 :: P=P^1000000 :: V=V+2 :: NEXT RP

250 CALL KEY(0,K,S):: IF S=0 THEN 250

998 END

999 ! FIRST VOICE

1000 DATA 40000,40000,40000,40000,40000,1568,1480,1760,1568,1976,1760,1480

1010 DATA 1480,1480,1568,1568,1760,1397,1319,1568,1568,1568,1661,1661

1020 DATA 1864,2094,1661,1864,1864,1864,1661,1661,1568,1661,1864,1661

1030 DATA 1661,1661,1864,1661,1568,1568,1568,1568,1568,1568,1760,1480

1040 DATA 1568,1568,1568,1568,1568,1568,1760,1480,1568,1568,1568,1568

1099 ! SECOND VOICE

1100 DATA 330,330,330,311,311,311,311,330,330,330,311,311,311,311,311

1110 DATA 440,440,440,415,415,415,523,523,523,349,349,349,349,349

1120 DATA 523,523,523,349,349,349,349,330,330,330,311,311,311,311,311

1130 DATA 330,330,330,311,311,311,311,330,330,330,330,330,330,330,330

1199 ! THIRD/BASS VOICE

1200 DATA 988,988,988,988,988,988,988,988,988,988,988,988,988,988

1210 DATA 1319,1319,1319,1319,1319,1319,1319,1319,1319,1319,1319,1319,1319,1319

1220 DATA 1568,1568,1568,1047,1047,1047,1047,1047,1047,1047,1047,1047,1047,1047

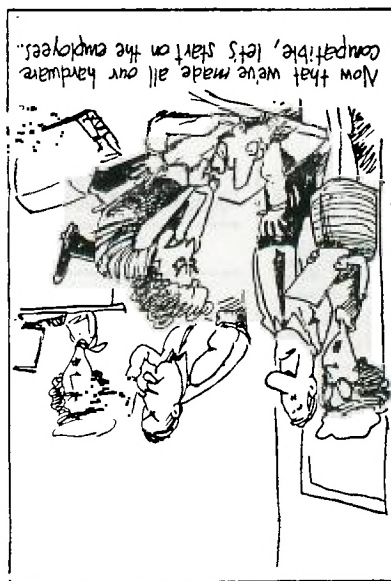
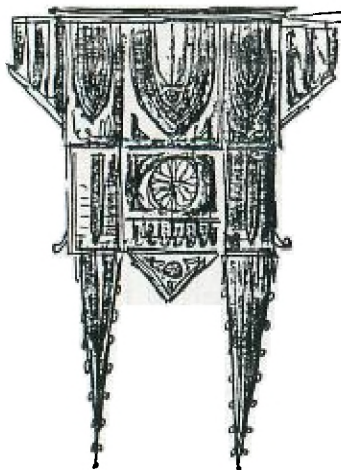
1230 DATA 1568,1568,1568,1047,1047,1047,1047,1047,1047,1047,1047,1047,1047,1047

1240 DATA 988,988,988,988,988,988,988,988,988,988,988,988,988,988

SOFTWARE



HARDWARE



THE WEST PENN 99'ERS
% JOHN F. WILLFORTH
R.D.# 1 BOX 73A
JEANNETTE, PA
15644

AUGUST ISSUE

