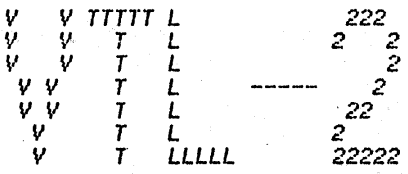
V V TTTTT L 222
V V T L 2 2
V V T L 2
V V T L ----- 2
V V T L 22
V T L 2
V T LLLLL 22222

V T L - 2

A VERY TINY LANGUAGE

FOR THE ALTAIR 8800

INTRODUCTION:

THE STATEMENTS THAT MAY BE ENTERED AS INPUT TO THE VTL-2 INTERPRETER ARE OF TWO TYPES.

1. DIRECT STATEMENTS, WHICH HAVE NO LINE NUMBER, AND ARE EXECUTED IMMEDIATLY AFTER THEY ARE ENTERED.

2. PROGRAM STATEMENTS, WHICH ARE USED TO BUILD A PROGRAM, AND ARE NOT EXECUTED UNTIL THE PROGRAM IS RUN. PROGRAM STATEMENTS MUST HAVE LINE NUMBERS IDENTIFIYING THEIR LOCATION IN THE PROGRAM.

VTL-2 IS SIMPLE ENOUGH FOR THE BEGINNER TO USE EASILY, AND YET POWERFUL ENOUGH TO SERVE THE NEEDS OF THE MOST ADVANCED USERS. THE SUBSCRIPTED MEMORY REFERENCE COMMANDS, AND FULL INPUT-OUTPUT FORMAT CONTROL, MAKE VTL-2 A VERSATILE LANGUAGE SUITIBLE FOR SOLVING A WIDE RANGE OF COMPUTER PROBLEMS.

PRELIMINARY CONCEPTS:

LINE NUMBERS MUST PRECEDE EACH PROGRAM STATEMENT, AND MUST BE SEPARATED FROM THAT STATEMENT BY A SINGLE BLANK SPACE. THESE NUMBERS MUST BE IN THE RANGE OF 1-65535. LINE NUMBER ZERO IS NOT PERMITTED. EACH LINE ENDS WITH A CARRIAGE RETURN, AND MUST BE LESS THAN 73 CHARACTERS LONG.

IT IS RECOMMENDED THAT LINES BE NUMBERED IN STEPS OF TEN (10, 20, 30,.... ETC.) SO THAT NEW STATEMENTS MAY BE INSERTED IF NECESSARY.

VARIABLES MAY BE REPRESENTED BY ANY SINGLE ALPHABETIC OR SPECIAL CHARACTER (EG. PUNCTUATION MARK. !"#$%&'()=-+*:;?/>.<,[]). MOST OF THESE ARE AVAILABLE FOR THE USER TO DEFINE AS HE WISHES. A FEW OF THE VARIABLE NAMES HOWEVER, HAVE BEEN SET ASIDE FOR SPECIAL PURPOSES. THESE SO-CALLED "SYSTEM VARIABLES" WILL BE DISCUSSED IN DETAIL BELOW.

THE VALUE ASSIGNED TO A VARIABLE MAY BE EITHER A NUMERIC VALUE IN THE RANGE 0-65535, OR A SINGLE ASCII CHARACTER (INCLUDING CONTROL-CHARACTERS). NUMERIC AND STRING VALUES MAY BE FREELY INTERCHANGED, IN WHICH CASE, THE CHARACTERS ARE EQUIVALENT TO THE DECIMAL VALUE OF THEIR ASCII CODE REPRESENTATION. THUS, IT BECOMES POSSIBLE TO ADD 1 TO THE LETTER "A", GIVING AS A RESULT, THE LETTER "B".

THE ARITHMETIC OPERATIONS PERMITTED FOR USE IN EXPRESSIONS ARE:

- \+ (ADDITION)
- \- (SUBTRACTION)

```
*  (MULTIPLICATION)
/  (DIVISION)
=  (TEST FOR EQUALITY)
>  (TEST FOR GREATER THAN OR EQUAL TO)
<  (TEST FOR LESS THAN)
```

THE TEST OPERATIONS, EQUAL TO, GREATER THAN OR EQUAL TO, AND LESS THAN, ALL RETURN A VALUE OF ZERO IF THE TEST FAILS, AND A VALUE OF ONE IF THE TEST IS SUCCESSFUL.

EXPRESSIONS IN VTL-2 MAY CONTAIN ANY NUMBER OF VARIABLES OR NUMERIC VALUES (LITERALS) CONNECTED BY ANY OF THE ABOVE OPERATION SYMBOLS. PARENTHESES MAY BE USED TO ALTER THE ORDER OF EXECUTION OF THE OPERATIONS. IF NO PARENTHESES ARE INCLUDED, THE OPERATIONS PROCEED IN STRICTLY RIGHT TO LEFT ORDER.

THE VALUE RESULTING FROM THE EXPRESSION MUST BE ASSIGNED TO SOME VARIABLE NAME. THIS IS DONE WITH THE EQUAL SIGN. NOTE THAT THE SYMBOL HAS TWO MEANINGS DEPENDING ON WHERE IT OCCURS IN THE EXPRESSION. THE EXPRESSION "A=B=C" MEANS TEST B AND C FOR EQUALITY. IF THEY ARE EQUAL, PUT A ONE IN A; IF THEY ARE UNEQUAL, PUT A ZERO IN A.

SOME EXAMPLES OF VALID ARITHMETIC EXPRESSIONS WOULD BE:

```
Y=A*(X*X)+B*X+C          WITH LEFT TO RIGHT EXECUTION THIS IS
                         EQUIVELANT TO:  Y=(A*X*X+B)*X+C
Y=(A*X*X)+(B*X)+C        WHICH IS EQUIVALENT TO:  AX↑2+BX+C
```

NOTICE HOW THE ABSENCE OF PARENTHESES AROUND THE QUUANTITY B*X IN THE FIRST EXPRESSION HAS COMPLETELY ALTERED ITS MEANING. KEEP THE LEFT TO RIGHT ORDER IN MIND, AND WHEN IN DOUBT, USE PARENTHESES TO CONTROL THE ORDER OF EVALUATION.

SYSTEM VARIABLES:

IN ORDER TO CONSERVE SPACE, AND TO PROVIDE A MORE CONSISTENT SYNTAX, VTL-2 USES "SYSTEM VARIABLES" TO ACCOMPLISH FUNCTIONS USUALY DONE WITH SPECIAL KEY WORDS IN OTHER LANGUAGES. THIS CONVENTION IS PROBABLY THE SINGLE MOST IMPORTANT REASON FOR ITS TINY SIZE.

THESE SPECIAL VARIABLES ARE USED FOR SUCH FUNCTIONS AS THE BASIC 'PRINT, GOTO, GOSUB, RETURN, IF, AND RANDOM' FUNCTIONS.

THE SYSTEM VARIABLE "NUMBER" OR "POUND SIGN" (#) REPRESENTS THE LINE NUMBER OF THE LINE BEING EXECUTED. UNTIL THE STATEMENT HAS BEEN COMPLETED, IT WILL CONTAIN THE CURRENT LINE NUMBER. SO THAT THE STATEMENT:

```
100 A=#
```

IS EQUIVELENT TO SIMPLY WRITING "100 A=100". AFTER COMPLETION OF A LINE, THIS VARIABLE WILL CONTAIN THE NUMBER OF THE NEXT LINE TO BE EXECUTED. IF NOTHING IS DONE TO THE VARIABLE, THIS WILL BE THE NEXT LINE IN THE PROGRAM TEXT. IF A STATEMENT CHANGES #, HOWEVER, THE NEXT LINE EXECUTED WILL BE THE LINE WITH THE NUMBER THAT MATCHES THE VALUE OF #. THUS THE VARIABLE # MAY BE USED TO TRANSFER CONTROL TO A DIFFERENT PART OF THE PROGRAM.

THIS IS THE VTL-2 EQUIVELANT OF THE BASIC "GOTO" STATEMENT. FOR EXAMPLE:

```
#=300   MEANS   "GOTO 300"
```

IF THE # VARIABLE SHOULD EVER BE SET TO ZERO BY SOME STATEMENT, THIS VALUE WILL BE IGNORED, AND THE PROGRAM WILL PROCEED AS IF NO CHANGE HAD TAKEN PLACE. THIS FACT ALLOWS US TO WRITE "IF" STATEMENTS IN VTL-2. CONSIDER THE FOLLOWING EXAMPLE:

```
10 X=1            SET X EQUAL TO 1
20 #=(X=25)*50    IF X=25 GOTO 50
30 X=X+1          ADD 1 TO X
40 #=20           GOTO 20
50 .....          CONTINUE
```

NOTICE THAT THE QUANTITY (X=25) WILL HAVE THE VALUE ONE, IF IT IS TRUE THAT X IS EQUAL TO 25, AND THE VALUE ZERO IF IT IS FALSE. WHEN THIS LOGICAL VALUE IS MULTIPLIED TIMES 50, THE RESULT WILL BE EITHER ZERO, OR 50. IF IT IS 50 THE STATEMENT CAUSES A "GOTO 50" TO OCCUR. IF THE VALUE IS ZERO, A "GOTO 0" WHICH IS A DUMMY OPERATION, CAUSES THE NEXT STATEMENT DOWN (NUMBER 30) TO BE EXECUTED.

TAKING ADVANTAGE OF LEFT-TO-RIGHT EVALUATION, TWO BYTES OF MEMORY COULD BE SAVED BY WRITING:

```
20 #=X=25*50
```

EACH AND EVERY TIME THE VALUE OF # IS CHANGED BY A PROGRAM STATEMENT, THE OLD-VALUE+1 IS SAVED IN THE SYSTEM VARIABLE "EXCLAMATION POINT" (!). IN OTHER WORDS AFTER EXECUTING A GOTO, THE LINE NUMBER OF THE LINE THAT FOLLOWS THE GOTO IS SAVED SO THAT A SUBROUTINE WILL KNOW WHICH PROGRAM STATEMENT CALLED IT, AND WILL KNOW WHERE TO RETURN WHEN FINISHED. THUS THE # VARIABLE IS USED FOR BOTH GOTO AND GOSUB OPERATIONS. FOR EXAMPLE:

```
10 X=1
20 #=100
30 X=2
40 #=100
50 X=3
60 #=100
....
100 X=X*X
110 #=!          (GOTO BACK WHERE YOU CAME FROM)
```

IN THIS EXAMPLE, CONTROL PROCEEDS FROM LINE 20 TO LINE 100. AFTER THAT, LINE 110 CAUSES CONTROL TO RETURN TO LINE 30. WHEN LINE 40 IS EXECUTED, THE SUBROUTINE AT 100 WILL RETURN TO LINE 50.

THE ACTUAL VALUE STORED IN THE ! VARIABLE IS (OLD LINE NUMBER+1) BUT, VTL-2, IF IT DOES NOT FIND THE EXACT LINE NUMBER IT IS SEARCHING FOR, WILL TAKE THE NEXT HIGHER LINE NUMBER. THEREFORE, IF A PROGRAM STATEMENT SAYS "#=52" AND THERE ARE LINES NUMBERED 50 AND 60 WITH NOTHING IN BETWEEN, CONTROL PASSES TO THE NEXT HIGHER LINE NUMBER, 60.

THE SYSTEM VARIABLE "QUESTION MARK" (?) REPRESENTS THE USER'S TERMINAL. IT CAN BE EITHER AN INPUT, OR AN OUTPUT, DEPENDING ON WHICH SIDE OF THE EQUAL SIGN IT APPEARS.

THE STATEMENT "?=A" IS INTERPRETED AS "PRINT A", AND THE STATEMETNT "X=?"IS INTERPRETED AS "INPUT X". NOTE THAT THE "?" MAY BE INCLUDED ANYWHERE IN THE EXPRESSION. FOR EXAMPLE THE PROGRAM:

```
10 ?="ENTER THREE VALUES"
20 A=(?+?+?)/3
30 ?="THE AVERAGE IS"
40 ?=A
```

WILL REQUEST THREE INPUTS WHILE EXECUTING LINE 20.

WHEN TYPING IN A REPLY TO A REQUEST FOR INPUT, THE USER MAY ENTER ANY ONE OF THREE DIFFERENT TYPES OF DATA:

1. A DECIMAL NUMBER
2. A VARIABLE NAME
3. ANY VALID VTL-2 EXPRESSION

THUS, FOR EXAMPLE, THE USER MAY REPLY WITH SUCH THINGS AS "1004" OR "A+B*(9/X)". IN EACH CASE THE EXPRESSION IS COMPLETELY EVALUATED BEFORE THE RESULT IS PASSED TO THE INPUT STATEMENT. THE ONLY EXCEPTION IS THAT YOU ARE NOT ALLOWED TO RESPOND WITH ANOTHER QUESTION MARK AS THIS WILL MESS UP THE LINE POINTER IN THE INTERPRETER, CAUSING IT TO RETURN AN IMPROPER VALUE.

IF A CARRIAGE-RETURN, WITH NO VALUE, IS TYPED IN RESPONSE TO A REQUEST FOR INPUT, THE INTERPRETER WILL RETURN SOME UNDEFINED VALUE. THEREFORE, THIS IS NOT RECOMMENDED.

WHEN THE QUESTION MARK IS ON THE LEFT SIDE OF THE FIRST EQUAL SIGN IT REPRESENTS A PRINT STATEMENT. WHEN THIS OCCURS, EITHER OF TWO DIFFERENT THINGS MAY BE ON THE RIGHT SIDE OF THE EQUAL SIGN:

1. ANY VALID VTL-2 EXPRESSION (AS DEFINED ABOVE)
2. A STRING OF CHARACTERS ENCLOSED IN QUOTE MARKS ("")

WHEN THE EXPRESSION IS A NUMERIC ONE, THE VALUE IS COMPUTED, AND PRINTED AS A LEFT ADJUSTED, UNSIGNED, DECIMAL INTEGER, WITH NO LEADING OR TRAILING BLANKS. A CARRIAGE RETURN NEVER FOLLOWS THE PRINTING OF A DECIMAL VALUE.

WHEN THE EXPRESSION IS A QUOTED CHARACTER STRING, THE ACTUAL STRING OF CHARACTERS IS PRINTED WITH NO LEADING OR TRAILING BLANKS. A CARRIAGE-RETURN LINE-FEED SEQUENCE WILL FOLLOW THE PRINTING OF A STRING UNLESS A SEMICOLON FOLLOWS THE CLOSING QUOTE.

THE OMMISION OF LEADING AND TRAILING BLANKS ALLOWS COMPLETE CONTROL OF FORMATTING PRINTED OUTPUT. FOR EXAMPLE THE PROGRAM:

```
10 ?=50/2
20 ?=",";
30 ?=265+3
40 ?=". ";
50 ?=1.6
```

WILL PRINT THE LINE: "25,268.16" WITH NO SPACES BETWEEN THE PIECES. THIS FEATURE IS MOST OFTEN USED IN FLOATING POINT, AND MULTIPLE PRECISION SUBROUTINES. (SEE "FACTORIALS" IN THE SAMPLE PROGRAM SECTION.)

IF, AT ANY TIME, IT IS DESIRED TO HAVE A CARRIAGE-RETURN LINE-FEED PRINTED, THE STATEMENT ?="" WILL ACOMPLISH THIS.

THE SYSTEM VARIABLE "PER-CENT" (%) CONTAINS THE VALUE OF THE REMAINDER OF THE LAST DIVIIDE OPERATION. THIS VALUE WILL REMAIN THE SAME UNTIL THE NEXT DIVIDE OPERATION.

THE SYSTEM VARIABLE "APOSTROPHE" (') REPRESENTS A RANDOM NUMBER. THIS NUMBER WILL HAVE AN UNPREDICTABLE VALUE IN THE RANGE 0-65535. IF CALLED TWICE ON THE SAME LINE, THE SAME VALUE WILL BE RETURNED BOTH TIMES. THE VALUE OF THE VARIABLE IS SCRAMBLED EACH TIME ANY STATEMENT IS EXECUTED. THEREFORE, FOR BEST RESULTS, IT IS HIGHLY RECOMMENDED THAT AT LEAST ONE OTHER COMPUTATION BE PERFORMED BEFORE THE VALUE IS AGAIN CALLED FOR. THIS MAY EVEN BE A SIMPLE DUMMY STATEMENT SUCH AS "Z=Z+7". FOR AN EXAMPLE OF THIS SEE "DON'T LOSE YOUR AT" IN THE SAMPLE PROGRAMS SECTION.

IN ADDITION TO DECIMAL NUMERIC INPUT AND OUTPUT, THE SYSTEM VARIABLE "DOLLAR SIGN" ($) IS USED TO INPUT AND OUTPUT SINGLE CHARACTERS AS WITH THE QUESTION MARK VARIABLE, "A=$" MEANS "INPUT A SINGLE ASCII CHARACTER AND PLACE ITS NUMERIC VALUE IN A". SIMILARLY, "$=X" MEANS "PRINT THE SINGLE ASCII CHARACTER WHOSE VALUE IS STORED IN X". FOR EXAMPLE THE PROGRAM:

```
10 A=65
20 $=A
30 A=A+1
40 #=A<91*20
50 ?=""
```

WILL PRINT OUT, AS ONE CONTINUOUS STRING, ALL THE LETTERS OF THE ALPHABET: ABCDEFGHIJKLMNOPQRSTUVWXYZ. IF YOU WISH TO FIND OUT WHAT DECIMAL VALUES CORRESPOND TO WHICH CHARACTERS, THESE CAN BE FOUND IN THE 8800 REFERENCE MANUAL, OR SIMPLY COMPUTED BY TYPING THE DIRECT STATEMENT: "?=$" AND THEN ENTERING THE CHARACTER WHOSE DECIMAL VALUE IS TO BE FOUND

THE SYSTEM VARIABLE "ASTERISK" (*) REPRESENTS THE MEMORY SIZE OF YOUR COMPUTER. FOR A 1K SYSTEM THIS WOULD BE 1024, FOR A 17K SYSTEM IT WOULD BE 17*1024. WHEN THE MACHINE IS FIRST TURNED ON, AND VTL-2 IS CALLED FOR THE FIRST TIME, THE USER MUST TYPE IN THE VALUE OF THE * VARIABLE AS A DIRECT STATEMENT. "*=1024" FOR EXMAPLE.

IF A PERSON WISHES TO ALLOT SPACE FOR USER DEFINED MACHINE LANGUAGE SUBROUTINES, THEN THE VARIABLE * IS SET EQUAL TO THE BOTTON OF THE FIRST BYTE REQUIRED BY THE USER DEFINED ROUTINE.

IN THE 8800 VERSION ONLY, THE SYSTEM VARIABLE PERIOD (.) IS USED TO CONTROL THE TERMINAL. IF THE NUMBER IS ODD, (BIT 0 HIGH) THEN THE COMPUTER WILL 'ECHO' OR PRINT. IF THE NUMBER IS EVEN, THE COMPUTER WILL NOT ECHO OR PRINT. IF THE NUMBER DIVIDED BY 2 IS ODD, (BIT 1 HIGH) THEN THE COMPUTER WILL ACCEPT INPUT FROM THE CASSETTE. IF THE NUMBER DIVIDED BY FOUR IS ODD (BIT 2 HIGH) THEN THE COMPUTER WILL OUTPUT TO THE CASSETTE IN PARRALEL WITH THE TERMINAL.

ALSO IN THE 8800 VERSION ONLY, THE SYSTEM VARIABLE COMMA (,) REPRESENTS THE NUMBER OF NULLS THE COMPUTER WILL PUT OUT TO THE TERMINAL AFTER EVERY CARRIAGE-RETURN LINE-FEED.

IN GENERAL, ON THE 8800 VERSION, IN NORMAL OPERATION, COMMA IS ZERO ",=0" AND PERIOD IS 1 ".=1". HOWEVER TO READ FROM CASSETTE, YOU WOULD SET PERIOD EQUAL TO 2 AND COMMA EQUAL TO 0. ".=2",=0" THIS IS SO THAT THE TERMINAL DOES NOT SLOW UP THE COMPUTER IN READING THE TAPE. HOWEVER, IF THE TERMINALS SPEED IS GREATER THAN 300 BAUD, THEN PERIOD MAY BE SET TO 3 SO THAT YOU CAN SEE THE PROGRAM AS IT COMES IN.

TO STORE A PROGRAM ON CASSETTE, SET THE NULLS TO 4 ",=4" SET THE ECHO OFF, AND THE CASSETTE OUTPUT ON, ".=4" THEN TYPE "0" BUT DO NOT HIT THE CARRIAGE RETURN. START THE CASSETE TAPE RUNNING ON RECORD, AND THEN HIT RETURN. THE COMPUTER WILL THEN SAVE ITS CURRENT PROGRAM ON TAPE.

TO READ A PROGRAM IN FROM CASSETTE, SET THE NULLS TO 0 ",=0" TURN THE ECHO OFF, AND THE CASSETE INPUT ON, ".=2" TYPE "1" WITHE NO CARRIAGE RETURN, AND START THE TAPE RECORDER.

TO SAVE A PROGRAM ON PAPER TAPE, SET ".=1",",=3", TYPE "0" PUNCH 20 OR MORE NULLS, AND THEN HIT THE CARRIAGE-RETURN.

TO READ IN FROM PAPER TAPE, SET ".=1",",=0", TYPE "1" , START THE READER ON THE NULLS OF THE PAPER TAPE AND START THE READER TO RUN.

FINALY ON THE 8800 VERSION ONLY, THE SYSTEM VARIABLE UP-ARROW (↑) IS THE NUMBER OF THE TERMINAL OPTION CURRENTLY RUNNING. IF THE NUMBER IS ODD (BIT 0 HIGH) THEN THE TERMINAL WILL BE SET FOR A 2SIO BOARD AT PORT 18, ELSE IF THE NUMBER DIVIDED BY 8 IS ODD (BIT 3 HIGH) THE TERMINAL WIL BE A 2SIO BORD AT PORT 16. FINALLY IF NEITHER SITUATION IS TRUE, (BIT 3 AND BIT 0 LOW) THE THE TERMINAL WILL BE SET FOR A SINGLE-SIO BOARD AT PORT 0.

THE SYSTEM VARIABLE "AMPERSAND" (&) REPRESENTS THE NEXT AVAILABLE BYTE OF MEMORY IN THE PROGRAM BUFFER. WHEN FIRST CALLING VTL-2, OR WHEN IT IS DESIRED TO ERASE THE PRESENT PROGRAM, THIS MUST BE INITIALIZED TO THE VALUE 264. "&=264" AS A DIRECT STATEMENT.

AT ANY GIVEN TIME, THE USER MAY FIND OUT HOW MUCH OF HIS MEMORY STILL REMAINS UNUSED BY TYPING "?=*-&". THIS WILL CAUSE THE SYSTEM TO RESPOND WITH THE NUMBER OF BYTES REMAINING. A MINIMUM OF AT LEAST 3 BYTES ARE NEEDED FOR ANY LINE OF VTL-2. THE LINE NUMBERS ARE SAVED IN BINARY, AND REQUIRE TWO BYTES REGARDLESS OF THEIR DECIMAL VALUES. THE LINES "1 X=Y" AND "65000 X=Y" BOTH TAKE UP AN IDENTICAL 7 BYTES OF MEMORY, AND ARE EXAMPLES OF THE NORMAL MINIMUM VALID VTL-2 LINE.

ANY MEMORY REMAINING PAST THE END OF A PROGRAM MAY BE USED AS ARRAY STORAGE. THIS ARRAY STORAGE MAY BE USED FOR SAVING NUMERIC OR STRING VALUES. THE ARRAY DOES NOT HAVE A NAME, SINCE THERE IS ONLY ONE, BUT, IT CAN BE DIVIDED UP INTO SEVERAL PIECES AND USED FOR DIFFERENT GROUPS OF DATA. (SEE "CIPHER" IN THE SAMPLE PROGRAMS SECTION.) A SUBSCRIPT EXPRESSION IS IDENTIFIED BY A COLON AND A RIGHT PARENTHESES. THE COLON MARKS THE BEGINNING OF THE EXPRESSION AND THE RIGHT PARENTHESES MARKS THE END. THUS, FOR EXAMPLE, ":1)=0" PLACES A ZERO IN THE FIRST TWO BYTE WORD PAST THE END OF THE PROGRAM, AND ":2+7)=A" PLACES THE VALUE OF A IN THE 9TH TWO-BYTE WORD PAST THE END OF THE PROGRAM.

SUBSCRIPTS SHOULD NOT BE ALLOWED TO BE LESS THAN ONE (1) AS THIS WILL POINT THE SUBSCRIPT INTO THE PROGRAM AND COULD CAUSE IT TO BE WIPED OUT.

SUBSCRIPT EXPRESSIONS MAY BE ANY VALID VTL-2 NUMERIC EXPRESSIONS THIS EXMAPLE SHOULD CLARIFY THE USE OF SUBSCRIPT EXPRESSIONS:

```
10 I=1            SET POINTER
20 :I)=$          INPUT A CHARACTER TO NEXT ARRAY WORD
30 #=:I)=13*60    GOTO 60 IF ITS A CARRIAGE RETURN CHARACTER
40 I=I+1          POINT TO NEXT ARRAY WORD
50 #=20           GO GET ANOTHER CHARACTER
60 ?=""           PRINT CARRIAGE RETURN-LINE FEED
70 I=1            RESET POINTER
80 $=:I)          PRINT ITH CHARACTER
90 #=:I)=13*120 IF CARRIAGE RETURN THEN GOTO 120
100 I=I+1         POINT TO NEXT CHARACTER
110 #=80          GO GET NEXT CHARACTER
120 ?=""          PRINT CARRIAGE RETURN-LINE FEED
```

THE ABOVE EXAMPLE WILL READ IN ANY STRING OF CHARACTERS TYPED BY THE USER, SUCH AS A SENTANCE, OR PARAGRAPH, UNTIL A CARRIAGE RETURN IS TYPED. IT WILL THEN ECHO BACK THE COMPLETE STRING AS IT WAS TYPED.

FOR FURTHER EXAMPLES, STUDY THE GAME PROGRAMS WHICH USE CHARACTER INPUT AND THOSE THAT HAVE ARRAYS REPRESENTING THE PLAYING BOARD. THESE WILL BE FOUND IN THE SAMPLE PROGRAMS SECTION.

SINCE SUBSCRIPTS REFER TO TWO BYTE WORDS, AND SINCE VALUES AS LARGE AS 65535 ARE ALLOWED AS SUBSCRIPTS, IT IS POSSIBLE THAT LARGE VALUES IN THE SUBSCRIPT EXPRESSION MAY "WRAP AROUND" THE END OF MEMORY AND REACH LOCATIONS WITHIN THE PROGRAM TEXT. THEREFORE, THERE IS A DANGER THAT VTL-2 PROGRAMS USING COMPUTED SUBSCRIPTS MAY "CLOBBER" THEMSELVES. ON THE OTHER HAND, THIS ALSO MEANS THAT A VTL-2 PROGRAM MAY MODIFY ITSELF, ALTHOUGH THIS PRACTICE IS NOT RECOMMENDED.

THE SYSTEM VARIABLE "GREATER THAN" (>) IS USED TO PASS A VALUE TO A MACHINE LANGUAGE SUBROUTINE. WHEN ENCOUNTERED ON THE LEFT SIDE OF THE EQUAL SIGN, THE EXPRESSION IS EVALUATED, THE VALUE PLACED AS A 16 BIT INTEGER IN THE B AND C REGISTERS, AND A "CALL ZERO" COMMAND GENERATED. (SEE 8800 MANUAL ON SUBROUTINE HANDLING AND RESTARTS) AT THE CONCLUSION OF THE MACHINE LANGUAGE SUBROUTINE, A RETURN INSTRUCTION RETURNS CONTROL TO VTL-2, AND PLACES THE VALUE FOUND IN THE B & C REGISTERS INTO THE SYSTEM VARIABLE >.

THERE IS NO "END" STATEMENT IN VTL-2. THE INTERPRETER SINPLY CONTINUES SEQUENTIALY THROUGH THE PROGRAM UNTIL IT RUNS OUT OF LINES TO EXECUTE, OR UNTIL A STATEMENT IS ENCOUNTERED WHICH WILL TRY TO TRANSFER CONTROL TO A LINE THAT IS GREATER IN NUMBER THAN ANY IN THE PROGRAM.

OPERATIONAL CHARACTERISTICS

WHEN THE 8800 IS FIRST TURNED ON THE FOLLOWING THINGS MUST BE DONE BEFORE ANY VTL-2 PROGRAM MAY BE ENTERED. FIRST LIFT THE STOP & THE RESET SWITCHES SIMUTANEOUSLY, THE RELEASE THE RESET SWITCH THEN THE STOP SWITCH. NEXT SET THE ADDRESS SWITCHES TO ADDRESS F800 HEX. A15 THROUGH A11 UP, A10 THROUGH A0 DOWN. THEN LIFT THE EXAMINE SWITCH.

FIANLY, SET THE TERMINAL OPTION ON THE SENSE SWITCHES, IF YOU HAVE A SINGLE-SIO BOARD AT PORT 0 THEN SET A11 AND A8 DOWN. IF YOU HAVE A 2SIO BOARD, THEN SET A11 HIGH, AND A8 HIGH IF YOU WANT PORT 18 OR LOW IF YOU WANT PORT 16.

AFTER SETTING THE PORT OPTION, PRESS "RUN".

ONCE VTL-2 IS IN CONTROL, THE MESSAGE "OK" WILL BE PRINTED. THE NEXT STEP IS TO SET YOUR MEMORY SIZE. THIS IS DONE BY TYPING *=1024 FOR A 1K SYSTEM, *=1024*17 FOR A 17K SYSTEM, AND SO ON.

FINALY, THE USER MUST SET THE "END OF PROGRAM" POINTER. THIS IS DONE BY TYPING &=320. THIS NUMBER WILL BE THE SAME FOR ALL 8800 SYSTEMS REGARDLESS OF MEMORY SIZE.

VTL-2 IS NOW READY TO BEGIN ACCEPTING PROGRAMS AND COMMANDS. IF AT ANY TIME IT IS DESIRED TO ERASE THE PROGRAM IN MEMORY, REPERT THE LAST TWO STEPS GIVEN ABOVE. THIS WILL RE-INITIALIZE THE VTL-2 PROGRAM BUFFER SPACE.

WHEN A PROGRAM LINE IS ENTERED, IT WILL BE INSERTED INTO ITS PROPER PLACE IN THE PROGRAM TEXT. THE LINE NUMBER INDICATES WHERE IT WILL BE INSERTED. IF THE LINE JUST TYPED HAS THE SAME LINE NUMBER AS A LINE ALREADY IN THE TEXT, THE OLD LINE WILL BE REPLACED BY THE NEW LINE. IF THE LINE NUMBER ONLY IS TYPED, FOLLOWED IMMEDIATLY BY A CARRIAGE RETURN, THE LINE WITH THAT NUMBER WILL BE DELETED.

WHILE TYPING IN PROGRAM LINES, THE SYSTEM SHOULD SINGLE SPACE, AND MAKE NO REPLIES TO LINES ENTERED. IF, AFTER TYPING A LINE, THE SYSTEM DOUBLE SPACES DOWN, AND PRINTS "OK" THAT INDICATES THAT THERE WAS NOT ENOUGH MEMORY AVAILABLE TO INSERT THE NEW LINE JUST TYPED.

THE USER MAY CHECK TO SEE HOW MUCH MEMORY IS STILL AVAILABLE BY TYPING THE DIRECT STATEMENT (NO LINE NUMBER) "?=*-&". THE SYSTEM WILL RESPOND WITH THE NUMBER OF UNUSED BYTES REMAINING.

WHILE TYPING IN A LINE, THE BACK-ARROW KEY (SHIFT-O ON SOME TERMINALS, OR UNDERLINE ON OTHERS) WILL CAUSE THE LAST CHARACTER TYPED TO BE DELETED FROM THE INPUT BUFFER. THE CHARACTER WILL STILL APPEAR ON THE SCREEN OR PRINTOUT, BUT WILL NO LONGER BE IN MEMORY. THUS THE LINE "A=B*C__+N" GOES IN AS "A=B+N", WHERE THE "*C" WAS ERASED IN MEMORY BY TWO BACK-ARROW CHARACTERS.

AT ANY TIME BEFORE HITTING RETURN, THE ENTIRE LINE MAY BE ERASED FROM MEMORY BY TYPING THE AT-SIGN CHARACTER (@) (SHIFT-P OR "CANCEL" ON SOME TERMINALS.)

TYPING THE SINGLE CHARACTER ZERO (0) FOLLOWED BY A CARRIAGE RETURN, CAUSES THE SYSTEM TO PRINT OUT A LISTING OF THE PROGRAM.

WHILE PRINTING IS TAKING PLACE, WHETHER AS A PROGRAM LISTING, OR AS OUTPUT FROM A PROGRAM, THE OPERATION MAY BE CANCELLED, AND CONTROL RETURNED TO VTL-2 BY PRESSING CONTROL-C. WHEN THIS IS DONE, THE SYSTEM COMPLETES IT'S CURRENT PRINT STATEMENT, AND THEN PRINTS "OK" TO ACKNOWLEDGE THE INTERUPTION.

IN ADDITION TO THIS, ANY OTHER KEY (PREFERABLY A NON-PRINTING CONTROL CHARACTER SUCH AS CONTROL-A) MAY BE PRESSED. THIS WILL CAUSE THE SYSTEN TO TEMPORARILY SUSPEND OPERATION, AND WAIT FOR ANOTHER KEY TO BE PRESSED. (AGAIN PREFERABLY A NON PRINTING CHARACTER.)

THIS FEATURE ALLOWS USERS WITH VIDEO TERMINALS TO LIST THEIR PROGRAMS A SECTION AT A TIME, HITTING CONTROL-A TO STOP THE LISTING, AND HITTING IT AGAIN TO RESUME LISTING.

NOTE THAT THESE CHARACTERS ALSO AFFECT PRINTING BEING DONE BY A PROGRAM. YOU MAY TEMPORARILY HALT YOUR PROGRAM WITH A CONTROL-A, AND START IT UP AGAIN WITH ANOTHER CONTROL-A. THESE KEYS WORK ONLY DURING PRINTING WHICH USES THE QUESTION MARK SYSTEM VARIABLE. STRING PRINTING WITH THE DOLLAR SIGN VARIABLE WILL NOT INTERUPT IN THIS MANNER. THIS ALLOWS THE USER THE OPTION OF MAKING HIS PROGRAM INTERUPTABLE OR NON-INTERUPTABLE.

SHOULD AN UNINTERUPTABLE PROGRAM BECOME "LOCKED UP" IN A LOOP, THE ONLY WAY TO BREAK OUT IS WITH THE FRONT PANEL RESET SWITCH. WHEN THIS IS DONE, THE JFC00 MUST BE TYPED TO RETURN TO VTL-2, BUT THE REMAINING STEPS LISTED ABOVE TO CLEAR THE PROGRAM MUST NOT BE PERFORMED!

TO RUN A PROGRAM, THE USER SIMPLY TYPES THE DIRECT COMMAND "#=1" (GOTO 1). THIS CAUSES THE SYSTEM TO FIND THE LOWEST NUMBERED LINE AND BEGIN EXECUTING THERE. IF IT IS DESIRED TO BEGIN EXECUTING AT SOME OTHER LINE NUMBER, SAY 1000, SIMPLY TYPE "#=1000", OR WHATEVER LINE NUMBER IS DESIRED.

COMMENTS MAY BE INSERTED ON ANY LINE BY PRECEEDING THEM WITH A RIGHT PARENTHESES. THIS SYMBOL MUST FOLLOW THE EXPRESSION ON THE LINE IMMEDIATELY, WITH NO BLANKS IN BETWEEN. THIS CAUSES THE SYSTEM TO STOP

TYPING THE SINGLE CHARACTER ZERO (0) FOLLOWED BY A CARRIAGE RETURN, CAUSES THE SYSTEM TO PRINT OUT A LISTING OF THE PROGRAM.

WHILE PRINTING IS TAKING PLACE, WHETHER AS A PROGRAM LISTING, OR AS OUTPUT FROM A PROGRAM, THE OPERATION MAY BE CANCELLED, AND CONTROL RETURNED TO VTL-2 BY PRESSING CONTROL-C. WHEN THIS IS DONE, THE SYSTEM COMPLETES IT'S CURRENT PRINT STATEMENT, AND THEN PRINTS "OK" TO ACKNOWLEDGE THE INTERUPTION.

IN ADDITION TO THIS, ANY OTHER KEY (PREFERABLY A NON-PRINTING CONTROL CHARACTER SUCH AS CONTROL-A) MAY BE PRESSED. THIS WILL CAUSE THE SYSTEM TO TEMPORARILY SUSPEND OPERATION, AND WAIT FOR ANOTHER KEY TO BE PRESSED. (AGAIN PREFERABLY A NON PRINTING CHARACTER.)

THIS FEATURE ALLOWS USERS WITH VIDEO TERMINALS TO LIST THEIR PROGRAMS A SECTION AT A TIME, HITTING CONTROL-A TO STOP THE LISTING, AND HITTING IT AGAIN TO RESUME LISTING.

NOTE THAT THESE CHARACTERS ALSO AFFECT PRINTING BEING DONE BY A PROGRAM. YOU MAY TEMPORARILY HALT YOUR PROGRAM WITH A CONTROL-A, AND START IT UP AGAIN WITH ANOTHER CONTROL-A. THESE KEYS WORK ONLY DURING PRINTING WHICH USES THE QUESTION MARK SYSTEM VARIABLE. STRING PRINTING WITH THE DOLLAR SIGN VARIABLE WILL NOT INTERUPT IN THIS MANNER. THIS ALLOWS THE USER THE OPTION OF MAKING HIS PROGRAM INTERUPTABLE OR NON-INTERUPTABLE.

SHOULD AN UNINTERUPTABLEPROGRAM BECOME "LOCKED UP" IN A LOOP, THE ONLY WAY TO BREAK OUT IS WITH THE FRONT-PANEL SWITCHES. LIFT THE STOP SWITCH, SET THE ADDRESS SWITCHES TO F81C HEX (A15 THROUGH A11 UP, A10 THROUGH A5 DOWN, A4, A3, A2 UP, A1, A0 DOWN.) LIFT EXAMINE, PRESS "RUN" THIS WILL RETURN YOU TO VTL-2 WITHOUT RESETING ANY OF THE POINTERS.

TO RUN A PROGRAM, THE USER SIMPLY TYPES THE DIRECT COMMAND "#=1" (GOTO 1). THIS CAUSES THE SYSTEM TO FIND THE LOWEST NUMBERED LINE AND BEGIN EXECUTING THERE. IF IT IS DESIRED TO BEGIN EXECUTING AT SOME OTHER LINE NUMBER, SAY 1000, SIMPLY TYPE "#=1000", OR WHATEVER LINE NUMBER IS DESIRED.

COMMENTS MAY BE INSERTED ON ANY LINE BY PRECEEDING THEM WITH A RIGHT PARENTHESES. THIS SYMBOL MUST FOLLOW THE EXPRESSION ON THE LINE IMMEDIATELY, WITH NO BLANKS IN BETWEEN. THIS CAUSES THE SYSTEM TO STOP EVALUATING THE LINE AND GO ON TO THE NEXT LINE. IF A LINE IS TO CONTAIN ONLY A COMMENT, THE FIRST CHARACTER ON THE LINE MUST BE A RIGHT PARENTHESES.

THERE ARE NO ERROR MESSAGES IN VTL-2. IF AN EXPRESSION IS WRONG THE RESULTS OF EVALUATING THAT EXPRESSION WILL ALSO BE WRONG. IN OTHER WORDS, VTL-2 ASSUMES THAT YOU KNOW WHAT YOU ARE DOING, AND WILL DO ITS BEST TO EXECUTE ANY STATEMENT THAT YOU GIVE IT. THIS LEAVES WIDE LATITUDE FOR TRYING VARIOUS PROGRAMMING "TRICKS", BUT ALSO LEAVES THE RESPONSIBILITY OF VERIFYING PROGRAM ACCURACY WITH THE PROGRAMMER.

LIST OF FEATURES

VARIABLES

VARIABLE	MEANING
A-Z	COMMON VARIABLES USE FREELY FOR STORING VALUES

SYSTEM VARIABLES

!	RETURN ADDRESS
"	POINTS TO THE LINE # AFTER THE LAST #= STATEMENT
#	POINTER FOR LITERAL PRINT STATEMENTS
$	LINE NUMBER
%	SINGLE CHARACTER STRING (INPUT OR OUTPUT)
&	REMAINDER AFTER THE LAST DIVIDE OPERATION
'	POINTS TO THE LAST BYTE OF PROGRAM
(	RANDOM NUMBER
)	SETS START OF PARENTHESIZED EXPRESSION
	END
	SETS END OF LINE
	SETS END OF PARENTHESIZED EXPRESSION
	SETS END OF ARRAY DESCRIPTION
*	USED ALSO FOR REMARK STATEMENT
>	POINTS TO END OF MEMORY
?	MACHINE LANGUAGE SUBROUTINE
	PRINT STATEMENT WHEN ON LEFT OF EQUAL SIGN
	INPUT STATEMENT WHEN ON RIGHT OF EQUAL SIGN
[	DEFINES START OF ARRAY DESCRIPTION
	WHEN FOLOWING A LITERAL PRINT STATEMENT,
	SAYS DO NOT PRINT CARRIAGE-RETURN LINE-FEED
. -=; +, </\]\[	MAY BE USED FREELY AS STANDARD VARIABLES BUT USE IS NOT RECOMMENDED FOR LEGIBILTY REASONS

OPERATORS

+	ADD TO PREVIOUS VALUE
-	SUBTRACT FROM PREVIOUS VALUE
*	MULTIPLY TIMES PREVIOUS VALUE
/	DIVIDE PREVIOUS VALUE BY
=	IS PREVIOUS VALUE EQUAL TO (YES = 1, NO = 0)
<	IS PREVIOUS VALUE LESS THAN (YES = 1, NO = 0)
>	IS PREVIOUS VALUE EQUAL TO OR GREATER THAN (Y=1, N=0)

THE DEFAULT OPERATOR IS THE LESS THAN TEST.

HURKLE

```
100 ?=""
110 ?="A HURKLE IS HIDING ON A"
120 ?="10 BY 10 GRID.  HOMEBASE"
130 ?="ON THE GRID IS POINT 00"
140 ?="AND A GRIDPOINT IS ANY"
150 ?="PAIR OF WHOLE NUMBERS"
160 ?="TRY TO GUESS THE HURKLE'S"
170 ?="GRIDPOINT.  YOU GET 5 GUESSES"
180 ?=""
190 R=?/100*0+%
200 A=R/10
210 B=%
220 K=1
230 ?="GUESS #";
240 ?=K
250 ?="  ?";
260 X=?/10
270 Y=%
280 ?=""
290 #=X*10+Y=R*540
300 K=K+1
310 #=K=6*440
320 ?="GO ";
330 #=Y=B*370+(Y<B*360)
340 ?="SOUTH";
350 #=370
360 ?="NORTH";
370 #=X=A*410+(X<A*400)
380 ?="WEST";
390 #=410
400 ?="EAST";
410 ?=""
420 ?=""
430 #=230
440 ?=""
450 ?="SORRY THAT'S 5 GUESSES"
460 ?="THE HURKLE IS AT ";
470 ?=A
480 ?=B
490 ?=""
500 ?=""
510 ?="LETS PLAY AGAIN."
520 ?="HURKLE IS HIDING"
530 #=180
540 ?="YOU FOUND HIM IN ";
550 ?=K
560 ?=" GUESSES"
570 #=490
```

TIME OF DAY DIGITAL CLOCK

FOR 300 BAUD TERMINALS

```
10 ?="HOUR ?";
20 H=?
30 ?="MINUTE ?";
40 M=?
50 ?="SECOND ?";
60 S=?
70 ?="READY"
80 A=$
90 S=S+1
100 M=S/60+M
110 S=%
120 H=M/60+H
130 M=%
140 H=H/24*0+%
150 ?="TIME:  ";
160 ?=H/10
170 ?=%
180 ?=":";
190 ?=M/10
200 ?=%
210 ?=":";
220 ?=S/10
230 ?=%
240 $=13
250 A=B
260 T=31
270 T=T-1
280 #=T=0*90
290 #=270
```

FOR 110 BAUD TERMINALS

```
10 ?="HOUR ?";
20 H=?
30 ?="MINUTE ?";
40 M=?
50 ?="SECOND ?";
60 S=?
70 ?="READY"
80 A=$
90 S=S+1
100 M=S/60+M
110 S=%
120 H=M/60+H
130 M=%
140 H=H/24*0+%
150 ?=H/10
160 ?=%
170 ?=":";
180 ?=M/10
190 ?=%
200 ?=":";
210 ?=S/10
220 ?=%
230 $=13
240 A=B
250 A=B
260 A=B
270 A=B+B
280 T=14
290 T=T-1
300 #=T=0*90
310 #=290
```

FACTORIALS

CALCULATES FACTORIALS UNTIL IT RUNS OUT OF MEMORY
FOR 1K OF MEMORY THIS IS ABOUT 208!

```
10 A=1
20 L=2
30 :1)=1
40 I=2
50 :I)=0
60 I=I+1
70 #=L>I*50
80 ?=""
90 ?=""
100 ?=A
110 ?="! ="
120 ?=""
130 I=L+1
140 I=I-1
150 #=:I)=0*140
160 ?=:I)
170 I=I-1
180 #=I=0*220
190 ?=:I)/10
200 ?=%
210 #=170
220 A=A+1
230 I=1
240 C=0
250 X=:I)
260 :I)=A*X
270 #=:I)<X*320
280 :1)=:I)+C
290 C=:I)/100
300 :I)=%
310 I=I+1
320 #=L>I*250
330 #=C=0*80
340 L=L+1
350 #=*-&/2<L*300
360 :I)=C
370 #=290
```

WEEKDAY

```
10 #=440
20 ?="DAY OF THE WEEK"
30 ?=""
40 ?="MONTH?  ";
50 M=?
60 #=M>13*40
70 #=M=0*40
80 ?=DAY OF MONTH?  ";
90 D=?
100 ?="YEAR?  "
110 Y=?
120 #=Y>1800*230
130 #=Y<100*150
140 #=70
150 ?=""
160 ?="IS THAT 19";
170 ?=Y
180 ?="?  ";
190 K=$
200 #=K=89=0*70
210 ?="ES"
220 Y=Y+1900
230 C=Y/100
240 Y=%
250 #=Y/4*0+%=0*280
260 :1)=6
270 :2)=2
280 W=Y/4+Y+D+:M)+(2*(C=18))/7*0+%
290 #=300+(20*W)
300 ?="SUN";
310 #=430
320 ?="MON";
330 #=430
340 ?="TUES";
350 #=430
360 ?="WEDNES";
370 #=340
380 ?="THURS";
390 #=430
400 ?="FRI";
410 #=430
420 ?="SATUR";
430 ?="DAY"
440 :1)=0
450 :2)=3
460 :3)=3
470 :4)=6
480 :5)=1
490 :6)=4
500 :7)=6
510 :8)=2
520 :9)=5
530 :10)=0
540 :11)=3
550 :12)=5
560 #=20
```

STARSHOOTER

```
10 I=0
20 I=I+1
30 :I)=46
40 #=I<41*20
50 :25)=42
60 I=8
70 J=1
80 $=I-1/7+64
90 ?=" - ";
100 S=I+J
110 $=:S)
120 J=J+1
130 #=J=6*160
140 ?="  ";
150 #=100
160 I=I+7
170 ?=""
180 ?=""
190 #=I<43*70
200 ?=""
210 ?="    1  2  3  4  5"
220 ?=""
230 ?="YOUR MOVE --";
240 I=42
250 I=I+1
260 :I)=$
270 #=:I)=13*320
280 #=:I)=3*580
290 #=:I)=95=0*250
300 I=I-1
310 #=260
320 A=:43)-64
330 ?=""
340 #=A>6*230
350 B=:44)-48
360 #=B>6*230
370 S=A*7+1+B
380 ?=""
390 #=:S)=42*420
400 ?="THAT'S NOT A STAR!"
410 #=230
420 :S)=46
430 C=S-7
440 #=520
450 C=S-1
460 #=520
470 C=S+1
480 #=520
490 C=S+7
500 #=520
510 #=60
520 ↑=!
530 #=:C)=42*560
540 :C)=42
550 #=↑
560 :C)=46
570 #=↑
```

OBJECT OF THE GAME IS TO CHANGE THIS:

```
A - . . . . .
B - . . . . .
C - . . * . .
D - . . . . .
E - . . . . .
    1 2 3 4 5
```

TO THIS:

```
A - * * * * *
B - *       *
C - *       *
D - *       *
E - * * * * *
    1 2 3 4 5
```

FACTORS OF A NUMBER

```
10 ?="NUMBER?  ";
20 N=?
30 X=N
40 $=22
50 ?=N
60 ?=" IS ";
70 #=N/2*80+%=0*140
80 D=3
90 Q=N/D
100 #=%=0*160
110 #=D>Q*300
120 D=D+2
130 #=90
140 ?="EVEN. "
150 #=10
160 ?=""
170 ?=D
180 N=Q
190 Q=N/D
200 #=%=0*220
210 #=120
220 ?="↑";
230 P=1
240 N=Q
250 Q=N/D
260 P=P+1
270 #=%=0*240
280 ?=P
290 #=120
300 #=N=1*340
310 #=N=X*390
320 ?=""
330 ?=N
340 ?=""
350 ?=""
360 ?="DONE"
370 ?=""
380 #=10
390 ?="PRIME. "
400 ?=""
410 #=340
```

DON'T LOSE YOUR AT!

BY

ED VERNER

ADAPTED TO VTL-2 BY GARY SHANNON

(A GAME SIMILAR TO "BAGLES")

THE OBJECT OF THE GAME IS TO GUESS THE SECRET NUMBER PICKED BY THE COMPUTER. THE NUMBER HAS THREE DIGITS, NO ZEROES, AND NO DIGIT IS REPEATED. AFTER YOU TYPE YOUR GUESS, THE COMPUTER WILL PRINT AN "IT" FOR EVERY CORRECT DIGIT IN THE WRONG POSITION, AND AN "AT" FOR EVERY CORRECT DIGIT IN THE RIGHT POSITION. YOU WIN WHEN YOU GET 3 "AT'S".

EACH TIME THAT YOU GUESS INCORRECTLY, YOU LOSE 5% OF THE POINTS YOU HAVE LEFT.

```
10 T=0
20 L=0
30 ?="DON'T LOSE YOUR 'AT'"
40 X='/9*0+%+1
50 Y='/9*0+%+1
60 #=X=Y*40
70 Z='/9*0+%+1
80 #=X=Z*40
90 #=Y=Z*40
100 ?="I'VE GOT A NUMBER."
105 L=L+1
110 P=10000
120 ?=""
130 ?="YOU HAVE ";
140 ?=P/100
150 ?=". ";
160 ?=%/10
170 ?=%
180 ?=" POINTS LEFT"
190 ?=""
200 ?="WHAT'S YOUR GUESS? -- ";
210 G=?
220 A=G/100
230 B=%/10
240 C=%
260 S=600
270 #=A=Y*S
280 #=A=Z*S
290 #=B=X*S
300 #=B=Z*S
310 #=C=X*S
320 #=C=Y*S
330 K=0
340 S=620
350 #=A=X*S
360 #=B=Y*S
370 #=C=Z*S
380 #=K<3*580
390 ?=""
400 ?="YOU WIN ";
410 ?=P/100
420 ?=". ";
430 ?=%/10
440 ?=%
450 ?=" POINTS FOR A TOTAL OF ";
460 T=T+P
490 ?=T/100
500 ?=". ";
510 ?=%/10
520 ?=%
540 ?=" POINTS IN ";
550 ?=L
560 ?=" GAMES."
570 #=30
580 P=P/20*19
590 #=120
600 ?="IT ";
610 #=!
620 ?="AT ";
630 K=K+1
640 #=!
```

********** HAVE FUN! **********

CRAPS!

```
10 T=100
20 $=22
30 ?="CRAPS!"
40 ?=""
50 ?="HOW MUCH DO YOU BET? - ";
60 B=?
70 #=B=0*90
80 ?="GOOD LUCK!"
90 #=B=0*480
100 #=T>B*160
110 ?="TOO MUCH!"
120 ?="YOU HAVE $";
130 ?=T
140 ?=" LEFT. "
150 #=40
160 ?=""
170 ?="ROLL-";
180 A=?
190 $=22
200 ?="FIRST ROLL:  ";
210 #=500
220 #=R=7*360
230 #=R=11*360
240 #=R<4*390
250 #=R=12*390
260 ?=""
270 ?=R
280 ?=" IS YOUR POINT"
290 P=R
300 ?="ROLL-";
310 A=$
320 #=500
330 #=R=7*390
340 #=R=F*360
350 #=300
360 ?="YOU WIN"
370 T=T+B
380 #=120
390 T=T-B
400 ?="YOU LOSE"
410 #=T=0*430
420 #=120
430 ?="YOU ARE BUSTED!"
440 ?="MOVE OVER AND LET THE NEXT"
450 ?="SUCKER TRY. "
460 ?=""
470 #=10
480 ?="BE SERIOUS"
490 #=40
500 R=⁄⁄6*0+%+1
510 ?=R
520 X=X+11213
530 ?=" AND ";
540 S=⁄⁄6*0+%+1
550 X=X*56001
560 ?=S
570 ?="  (";
580 R=R+S
590 ?=R
600 ?=")"
610 #=!
```

CIPHER GAME

```
10 I=0
20 I=I+1
30 :I)=I+64
40 #=I<26*20
50 I=1
60 ?=""
70 M='/26*0+%+1
80 H=:M)
90 :M)=:I)
100 :I)=H
110 I=I+1
120 #=I<27*70
130 ?="TEXT?"
140 ?=""
150 I=27
160 :I)=$
170 #=:I)=13*220
180 #=:I)=95=0*200
190 I=I-2
200 I=I+1
210 #=160
220 ?=""
230 I=27
240 #=:I)<64*270
250 T=:I)-64
260 :I)=:T)
270 I=I+1
280 #=:I)>14*240
290 ?=""
300 ?="CODE:"
310 ?=""
320 I=27
330 $=:I)
340 #=:I)=13*370
350 I=I+1
360 #=330
370 ?=""
380 ?="SWITCH? - ";
390 A=$
400 B=$
410 #=B=64*370
420 I=27
430 #=:I)=A*490
440 #=:I)=B=0*460
450 :I)=A
460 I=I+1
470 #=:I)=13*290
480 #=430
490 :I)=B
500 #=460
```

PHRASE SORT

```
10 $=22
20 I=0
30 I=I+1
40 :I)=$
50 L=:I)=95*2
60 I=I-L
70 #=:I)>14*30
80 ?=""
90 I=1
100 K=1
110 J=K
120 #=:K)=32*160
130 #=:J)=32*150
140 #=:K)>:J)*160
150 J=K
160 K=K+1
170 #=:K)>14*120
180 H=:I)
190 :I)=:J)
200 :J)=H
210 I=I+1
220 #=:I)>14*100
230 I=0
240 I=I+1
250 $=:I)
260 #=:I)>14*240
270 ?=""
```

LIFE

NEEDS 2K OF MEMORY FOR OPERATION

```
10 #=710
20 Z=!
30 #=X*Y=0*(X=20)+1/2*Z
40 #=Y=20*Z
50 L=Y-1*19+X
60 #=:L)=10+(:L)=32)*Z
70 S=S+1
80 #=Z
90 I=1
100 J=1
110 S=0
120 X=I-1
130 Y=J-1
140 #=T
150 X=I
160 #=T
170 X=I+1
180 #=T
190 Y=J
200 #=T
210 X=I-1
220 #=T
230 Y=J+1
240 #=T
250 X=I
260 #=T
270 X=I+1
280 #=T
290 K=J-1*19+I
300 #=S>3*340
310 #=:K)=32*400
320 :K)=S=2*42
330 #=400
340 #=S>4*380
350 #=:K)=42*400
360 :K)=10
370 #=400
380 #=:K)=32*400
390 :K)=0
400 J=J+1
410 #=J<20*110
420 I=I+1
430 #=I<20*100
440 ?=""
450 ?=""
460 P=0
470 ?=" ";
480 J=1
490 I=1
500 K=J-1*19+I
510 :K)=:K)+(:K)<30*32)
520 #=:K)=32*560
530 ?="<>";
540 P=P+1
550 #=570
560 ?="  ";
570 I=I+1
580 #=I<20*500
590 ?=""
600 ?=" ";
610 J=J+1
620 #=J<20*490
630 ?=""
640 ?="GEN= ";
650 ?=G
660 ?="     POP= ";
670 ?=P
680 ?="------------"
690 G=G+1
700 #=90
710 I=1
720 T=20
730 J=1
740 :I-1*19+J)=32
750 J=J+1
760 #=J<20*740
770 I=I+1
780 #=I<20*730
790 G=0
800 J=1
810 ?=""
820 I=1
830 $=10
840 #=J>10*860
850 ?=" ";
860 ?=J
870 ?=" ";
880 K=J-1*19+I
890 :K)=$
900 #=:K)=13*940
910 #=:K)=64*810
920 I=I+1
930 #=I<20*890
940 :K)=32
950 J=J+1
960 #=J<20*820
970 #=440
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

F800            0000 *VTL-2 FOR 8080
F800            0005 *VERSION 2.9
F800            0010 *BY FRANK MCCOY
F800            0015 *COPYRIGHT  1976
F800            0020 *
F800            0025 *BUFFER AREA
F800            0030             ORG   0
0000            0035 BRKPNT      EQU   $
0000            0040 RST0        DS    8
0008            0045 RST1        DS    8
0010            0050 RST2        DS    8
0018            0055 RST3        DS    8
0020            0060 RST4        DS    8
0028            0065 RST5        DS    8
0030            0070 RST6        DS    8
0038            0075 RST7        DS    8
0040            0080 AT          DS    2
0042            0085 VARS        DS    52
0076            0090 BRAK        DS    2
0078            0095 SAVE10      DS    2
007A            0100 BRIK        DS    2
007C            0105 UP          DS    2
007E            0110 SAVE11      DS    2
0080            0115 SAVE14      DS    2
0082            0120 EXCL        DS    2
0084            0125 QUOTE       DS    2
0086            0130 DOLR        DS    2
0088            0135 DOLLAR      DS    2
008A            0140 REMN1       DS    1
008B            0145 REMN2       DS    1
008C            0150 AMPR        DS    2
008E            0155 QUITE       DS    2
0090            0160 PAREN       DS    2
0092            0165 PARIN       DS    2
0094            0170 STAR        DS    2
0096            0175 PLUS        DS    2
0098            0180 COMA        DS    2
009A            0185 MINS        DS    2
009C            0190 PERD        DS    2
009E            0195 SLASH       DS    2
00A0            0200 SAVE0       DS    2
00A2            0205 SAVE1       DS    2
00A4            0210 SAVE2       DS    2
00A6            0215 SAVE3       DS    2
00A8            0220 SAVE4       DS    2
00AA            0225 SAVE5       DS    2
00AC            0230 SAVE6       DS    2
00AE            0235 SAVE7       DS    2
00B0            0240 SAVE8       DS    2
00B2            0245 SAVE9       DS    2
00B4            0250 COLN        DS    2
00B6            0255 SEMI        DS    2
00B8            0260 LESS        DS    2
00BA            0265 EQAL        DS    2
00BC            0270 GRAT        DS    2
00BE            0275 DECBUF      DS    4
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

00C2            0280 LASTD     DS    1
00C3            0285 DELIM     DS    1
00C4            0290 LINBUF    DS    73
010D            0295 *USER PROGRAM
010D            0300           ORG   140H
0140            0305 STACK     EQU   $
0140            0310 PRGM      EQU   $
0140            0315 *MAIN OPERATING SYSTEM
0140            0320           ORG   0F800H
F800 21 03 00   0325 RSTRT     LXI   H,3
F803 22 9C 00   0330           SHLD  PERD
F806 22 98 00   0335           SHLD  COMA
F809 7D         0340           MOV   A,L
F80A D3 10      0345           OUT   10H
F80C D3 12      0350           OUT   12H
F80E 3E 11      0355           MVI   A,11H
F810 D3 10      0360           OUT   10H
F812 D3 12      0365           OUT   12H
F814 DB FF      0370           IN    255
F816 E6 09      0375           ANI   9
F818 6F         0380           MOV   L,A
F819 22 7C 00   0385           SHLD  UP
F81C 31 40 01   0390 START     LXI   SP,STACK
F81F AF         0395           XRA   A
F820 21 64 FB   0400           LXI   H,OKM
F823 CD 44 FB   0405           CALL  STRNG
F826 21 00 00   0410 LOOP      LXI   H,0
F829 22 86 00   0415           SHLD  DOLR
F82C CD EB FA   0420           CALL  CVTLN
F82F D2 89 F8   0425           JNC   STMNT
F832 CD 78 F8   0430           CALL  EXEC
F835 CA 1C F8   0435           JZ    START
F838 CD C3 F8   0440 LOOP2     CALL  FIND
F83B D2 1C F8   0445 EQSTRT    JNC   START
F83E CD 6E F8   0450           CALL  LXHN
F841 22 86 00   0455           SHLD  DOLR
F844 2A 7E 00   0460           LHLD  SAVE11
F847 23         0465           INX   H
F848 23         0470           INX   H
F849 23         0475           INX   H
F84A CD 78 F8   0480           CALL  EXEC
F84D EB         0485           XCHG
F84E CA 61 F8   0490           JZ    LOOP3
F851 2A 7E 00   0495           LHLD  SAVE11
F854 CD 6E F8   0500           CALL  LXHN
F857 CA 61 F8   0505           JZ    LOOP3
F85A 23         0510           INX   H
F85B 22 82 00   0515           SHLD  EXCL
F85E C3 38 F8   0520           JMP   LOOP2
F861 E5         0525 LOOP3     PUSH  H
F862 2A 8C 00   0530           LHLD  AMPR
F865 44         0535           MOV   B,H
F866 4D         0540           MOV   C,L
F867 E1         0545           POP   H
F868 CD E2 F8   0550           CALL  FND3
F86B C3 3B F8   0555           JMP   EQSTRT
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

F86E 7E          0560 LXHN .    MOV   A,M
F86F 23          0565           INX   H
F870 66          0570           MOV   H,M
F871 6F          0575           MOV   L,A
F872 7D          0580 CPHD      MOV   A,L
F873 BB          0585           CMP   E
F874 C0          0590           RNZ
F875 7C          0595           MOV   A,H
F876 BA          0600           CMP   D
F877 C9          0605           RET
F878 22 AE 00    0610 EXEC      SHLD  SAVE7
F87B CD AD FA    0615           CALL  VAR2
F87E 23          0620           INX   H
F87F 7E          0625 SKIP      MOV   A,M
F880 CD E8 F8    0630           CALL  EVIL
F883 2A 86 00    0635 OUTX      LHLD  DOLR
F886 7C          0640           MOV   A,H
F887 B5          0645           ORA   L
F888 C9          0650           RET
F889 22 B0 00    0655 STMNT     SHLD  SAVE8
F88C 69          0660           MOV   L,C
F88D 60          0665           MOV   H,B
F88E 22 86 00    0670           SHLD  DOLR
F891 78          0675           MOV   A,B
F892 B1          0680           ORA   C
F893 C2 1F F9    0685           JNZ   SKP2
F896 2A 8C 00    0690           LHLD  AMPR
F899 EB          0695           XCHG
F89A 21 40 01    0700           LXI   H,PRGM
F89D CD 72 F8    0705 LST2      CALL  CPHD
F8A0 CA 1C F8    0710           JZ    START
F8A3 D5          0715           PUSH  D
F8A4 4E          0720           MOV   C,M
F8A5 23          0725           INX   H
F8A6 46          0730           MOV   B,M
F8A7 E5          0735           PUSH  H
F8A8 CD 91 F9    0740           CALL  PRNT2
F8AB E1          0745           POP   H
F8AC 23          0750           INX   H
F8AD CD C2 F9    0755           CALL  PNTMSG
F8B0 CD 4B FB    0760           CALL  CRLF
F8B3 D1          0765           POP   D
F8B4 C3 9D F8    0770           JMP   LST2
F8B7 2A 7E 00    0775 NXTXT     LHLD  SAVE11
F8BA 23          0780           INX   H
F8BB 23          0785 LOOKAG    INX   H
F8BC 7E          0790           MOV   A,M
F8BD A7          0795           ANA   A
F8BE C2 BB F8    0800           JNZ   LOOKAG
F8C1 23          0805 OUTD      INX   H
F8C2 C9          0810           RET
F8C3 2A 8C 00    0815 FIND      LHLD  AMPR
F8C6 4D          0820           MOV   C,L
F8C7 44          0825           MOV   B,H
F8C8 2A 86 00    0830           LHLD  DOLR
F8CB EB          0835           XCHG
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

F8CC 21 40 01   0840            LXI   H,PRGM
F8CF 22 7E 00   0845 FND2       SHLD  SAVE11
F8D2 79         0850            MOV   A,C
F8D3 BD         0855            CMP   L
F8D4 C2 DA F8   0860            JNZ   NXTUU
F8D7 78         0865            MOV   A,B
F8D8 BC         0870            CMP   H
F8D9 C8         0875            RZ
F8DA 7E         0880 NXTUU      MOV   A,M
F8DB 93         0885            SUB   E
F8DC 23         0890            INX   H
F8DD 7E         0895            MOV   A,M
F8DE 9A         0900            SBB   D
F8DF 2B         0905            DCX   H
F8E0 3F         0910            CMC
F8E1 D8         0915            RC
F8E2 CD B7 F8   0920 FND3       CALL  NXTXT
F8E5 C3 CF F8   0925            JMP   FND2
F8E8 FE 22      0930 EVIL       CPI   '"'
F8EA CA 43 FB   0935            JZ    STRNG-1
F8ED CD E2 F9   0940 EVALU      CALL  EVAL
F8F0 C5         0945            PUSH  B
F8F1 2A AE 00   0950            LHLD  SAVE7
F8F4 CD C4 FA   0955            CALL  CONVP
F8F7 C1         0960            POP   B
F8F8 FE 24      0965            CPI   '$'
F8FA C2 01 F9   0970            JNZ   ANDT
F8FD 79         0975            MOV   A,C
F8FE C3 B5 FB   0980            JMP   OUTCH
F901 D6 3F      0985 ANDT       SUI   '?'
F903 CA 91 F9   0990            JZ    PRNT2
F906 3C         0995            INR   A
F907 CC 00 00   1000            CZ    BRKPNT
F90A 71         1005            MOV   M,C
F90B 23         1010            INX   H
F90C 70         1015            MOV   M,B
F90D 21 8F 00   1020            LXI   H,QUITE+1
F910 7E         1025            MOV   A,M
F911 81         1030            ADD   C
F912 4F         1035            MOV   C,A
F913 2B         1040            DCX   H
F914 7E         1045            MOV   A,M
F915 88         1050            ADC   B
F916 71         1055            MOV   M,C
F917 23         1060            INX   H
F918 77         1065            MOV   M,A
F919 79         1070 CPBD       MOV   A,C
F91A BB         1075            CMP   E
F91B C0         1080            RNZ
F91C 78         1085            MOV   A,B
F91D BA         1090            CMP   D
F91E C9         1095            RET
F91F CD C3 F8   1100 SKP2       CALL  FIND
F922 D2 44 F9   1105            JNC   INSRT
F925 CD 6E F8   1110            CALL  LXHN
F928 C2 44 F9   1115            JNZ   INSRT
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

F92B CD B7 F8   1120            CALL  NXTXT
F92E EB         1125            XCHG
F92F 2A 7E 00   1130            LHLD  SAVE11
F932 CD 19 F9   1135 DELT       CALL  CPBD
F935 CA 3F F9   1140            JZ    FITIT
F938 1A         1145            LDAX  D
F939 77         1150            MOV   M,A
F93A 23         1155            INX   H
F93B 13         1160            INX   D
F93C C3 32 F9   1165            JMP   DELT
F93F 22 8C 00   1170 FITIT      SHLD  AMPR
F942 44         1175            MOV   B,H
F943 4D         1180            MOV   C,L
F944 2A B0 00   1185 INSRT      LHLD  SAVE8
F947 11 03 00   1190            LXI   D,3
F94A 7E         1195            MOV   A,M
F94B A7         1200            ANA   A
F94C CA 26 F8   1205            JZ    LOOP
F94F 13         1210 CNTLN      INX   D
F950 23         1215            INX   H
F951 7E         1220            MOV   A,M
F952 A7         1225            ANA   A
F953 C2 4F F9   1230            JNZ   CNTLN
F956 EB         1235 OPEN       XCHG
F957 09         1240            DAD   B
F958 EB         1245            XCHG
F959 21 94 00   1250            LXI   H,STAR
F95C 7B         1255            MOV   A,E
F95D 96         1260            SUB   M
F95E 23         1265            INX   H
F95F 7A         1270            MOV   A,D
F960 9E         1275            SBB   M
F961 D2 1C F8   1280            JNC   START
F964 EB         1285            XCHG
F965 22 8C 00   1290            SHLD  AMPR
F968 03         1295            INX   B
F969 23         1300            INX   H
F96A E5         1305            PUSH  H
F96B 2A 7E 00   1310            LHLD  SAVE11
F96E EB         1315            XCHG
F96F E1         1320            POP   H
F970 0B         1325 SLIDE      DCX   B
F971 2B         1330            DCX   H
F972 0A         1335            LDAX  B
F973 77         1340            MOV   M,A
F974 CD 19 F9   1345            CALL  CPBD
F977 C2 70 F9   1350            JNZ   SLIDE
F97A 2A 86 00   1355 DON        LHLD  DOLR
F97D 7D         1360            MOV   A,L
F97E 02         1365            STAX  B
F97F 03         1370            INX   B
F980 7C         1375            MOV   A,H
F981 02         1380            STAX  B
F982 2A B0 00   1385            LHLD  SAVE8
F985 2B         1390            DCX   H
F986 23         1395 MOVL       INX   H
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

F987 03         1400            INX   B
F988 7E         1405            MOV   A,M
F989 02         1410            STAX  B
F98A A7         1415            ANA   A
F98B C2 86 F9   1420            JNZ   MOVL
F98E C3 26 F8   1425            JMP   LOOP
F991 11 BE 00   1430 PRNT2      LXI   D,DECBUF
F994 21 D9 FA   1435            LXI   H,PWRS10
F997 D5         1440 CVD1       PUSH  D
F998 50         1445            MOV   D,B
F999 59         1450            MOV   E,C
F99A 46         1455            MOV   B,M
F99B 23         1460            INX   H
F99C 4E         1465            MOV   C,M
F99D 23         1470            INX   H
F99E E5         1475            PUSH  H
F99F EB         1480            XCHG
F9A0 CD 3E FA   1485            CALL  DIV
F9A3 EB         1490            XCHG
F9A4 7D         1495            MOV   A,L
F9A5 42         1500            MOV   B,D
F9A6 4B         1505            MOV   C,E
F9A7 E1         1510            POP   H
F9A8 D1         1515            POP   D
F9A9 C6 30      1520            ADI   '0'
F9AB 12         1525            STAX  D
F9AC 13         1530            INX   D
F9AD 7E         1535            MOV   A,M
F9AE FE 7E      1540            CPI   7EH
F9B0 C2 97 F9   1545            JNZ   CVD1
F9B3 21 BD 00   1550            LXI   H,DECBUF-1
F9B6 1B         1555            DCX   D
F9B7 1A         1560            LDAX  D
F9B8 F6 80      1565            ORI   128
F9BA 12         1570            STAX  D
F9BB 23         1575 ZRSUP      INX   H
F9BC 7E         1580            MOV   A,M
F9BD FE 30      1585            CPI   '0'
F9BF CA B8 F9   1590            JZ    ZRSUP
F9C2 AF         1595 PNTMSG     XRA   A
F9C3 32 C3 00   1600 STRTMSG    STA   DELIM
F9C6 47         1605            MOV   B,A
F9C7 7E         1610 OUTMSG     MOV   A,M
F9C8 23         1615            INX   H
F9C9 B8         1620            CMP   B
F9CA CA D3 F9   1625            JZ    CTLC
F9CD CD B3 FB   1630            CALL  OUTCH-2
F9D0 C3 C7 F9   1635            JMP   OUTMSG
F9D3 CD 69 FB   1640 CTLC       CALL  POLCAT
F9D6 D0         1645            RNC
F9D7 CD 82 FB   1650            CALL  INCH
F9DA FE 03      1655            CPI   3
F9DC CA 1C F8   1660            JZ    START
F9DF C3 82 FB   1665            JMP   INCH
F9E2 CD 84 FA   1670 EVAL       CALL  GETVAL
F9E5 7E         1675 NXTRM      MOV   A,M
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

F9E6 A7         1680            ANA   A
F9E7 C8         1685            RZ
F9E8 FE 29      1690            CPI   ')'
F9EA CA C1 F8   1695            JZ    OUTD
F9ED CD F8 F9   1700            CALL  TERM
F9F0 44         1705            MOV   B,H
F9F1 4D         1710            MOV   C,L
F9F2 2A A0 00   1715            LHLD  SAVE0
F9F5 C3 E5 F9   1720            JMP   NXTRM
F9F8 C5         1725 TERM       PUSH  B
F9F9 7E         1730            MOV   A,M
F9FA F5         1735            PUSH  PSW
F9FB 23         1740            INX   H
F9FC CD 84 FA   1745            CALL  GETVAL
F9FF 22 A0 00   1750            SHLD  SAVE0
FA02 F1         1755            POP   PSW
FA03 E1         1760            POP   H
FA04 FE 2B      1765            CPI   '+'
FA06 C2 0B FA   1770            JNZ   EVAL2
FA09 09         1775            DAD   B
FA0A C9         1780            RET
FA0B FE 2D      1785 EVAL2      CPI   '-'
FA0D C2 17 FA   1790            JNZ   EVAL3
FA10 7D         1795 HSUBB      MOV   A,L
FA11 91         1800            SUB   C
FA12 6F         1805            MOV   L,A
FA13 7C         1810            MOV   A,H
FA14 98         1815            SBB   B
FA15 67         1820            MOV   H,A
FA16 C9         1825            RET
FA17 FE 2A      1830 EVAL3      CPI   '*'
FA19 C2 31 FA   1835            JNZ   EVAL4
FA1C EB         1840 MULT       XCHG
FA1D 21 00 00   1845            LXI   H,0
FA20 3E 10      1850            MVI   A,16
FA22 F5         1855 MULT1      PUSH  PSW
FA23 29         1860            DAD   H
FA24 EB         1865            XCHG
FA25 29         1870            DAD   H
FA26 EB         1875            XCHG
FA27 D2 2B FA   1880            JNC   MULT2
FA2A 09         1885            DAD   B
FA2B F1         1890 MULT2      POP   PSW
FA2C 3D         1895            DCR   A
FA2D C2 22 FA   1900            JNZ   MULT1
FA30 C9         1905            RET
FA31 FE 2F      1910 EVAL4      CPI   '/'
FA33 C2 5F FA   1915            JNZ   EVAL5
FA36 CD 3E FA   1920 DIVIDE     CALL  DIV
FA39 22 8A 00   1925            SHLD  REMN1
FA3C EB         1930            XCHG
FA3D C9         1935            RET
FA3E EB         1940 DIV        XCHG
FA3F 21 00 00   1945            LXI   H,0
FA42 78         1950            MOV   A,B
FA43 B1         1955            ORA   C
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

FA44 C8         1960            RZ
FA45 3E 10      1965            MVI   A,16
FA47 F5         1970 DIV1       PUSH  PSW
FA48 29         1975            DAD   H
FA49 EB         1980            XCHG
FA4A 29         1985            DAD   H
FA4B EB         1990            XCHG
FA4C D2 50 FA   1995            JNC   DIV2
FA4F 23         2000            INX   H
FA50 CD 10 FA   2005 DIV2      CALL  HSUBB
FA53 13         2010            INX   D
FA54 D2 59 FA   2015            JNC   DIV3
FA57 09         2020            DAD   B
FA58 1B         2025            DCX   D
FA59 F1         2030 DIV3       POP   PSW
FA5A 3D         2035            DCR   A
FA5B C2 47 FA   2040            JNZ   DIV1
FA5E C9         2045            RET
FA5F 11 00 00   2050 EVAL5      LXI   D,0
FA62 CD 67 FA   2055            CALL  EVIL5
FA65 EB         2060            XCHG
FA66 C9         2065            RET
FA67 D6 3D      2070 EVIL5      SUI   '='
FA69 C2 74 FA   2075            JNZ   EVIL6
FA6C CD 10 FA   2080            CALL  HSUBB
FA6F C0         2085            RNZ
FA70 B5         2090            ORA   L
FA71 C0         2095            RNZ
FA72 13         2100            INX   D
FA73 C9         2105            RET
FA74 3D         2110 EVIL6      DCR   A
FA75 CA 7E FA   2115            JZ    EVIL7
FA78 CD 10 FA   2120            CALL  HSUBB
FA7B D0         2125            RNC
FA7C 13         2130            INX   D
FA7D C9         2135            RET
FA7E CD 10 FA   2140 EVIL7      CALL  HSUBB
FA81 D8         2145            RC
FA82 13         2150            INX   D
FA83 C9         2155            RET
FA84 CD EE FA   2160 GETVAL     CALL  CVBIN
FA87 D0         2165            RNC
FA88 FE 3F      2170            CPI   '?'
FA8A 23         2175            INX   H
FA8B C2 9B FA   2180            JNZ   VAR
FA8E 22 B2 00   2185            SHLD  SAVE9
FA91 CD 20 FB   2190            CALL  INLN
FA94 CD E2 F9   2195            CALL  EVAL
FA97 2A B2 00   2200            LHLD  SAVE9
FA9A C9         2205            RET
FA9B FE 24      2210 VAR        CPI   '$'
FA9D C2 A7 FA   2215            JNZ   VAR1
FAA0 CD 82 FB   2220            CALL  INCH
FAA3 4F         2225            MOV   C,A
FAA4 06 00      2230            MVI   B,0
FAA6 C9         2235            RET
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

FAA7 FE 28      2240 VAR1 .    CPI   '('
FAA9 CA E2 F9   2245           JZ    EVAL
FAAC 2B         2250           DCX   H
FAAD CD C4 FA   2255 VAR2      CALL  CONVP
FAB0 4E         2260           MOV   C,M
FAB1 23         2265           INX   H
FAB2 46         2270           MOV   B,M
FAB3 2A AC 00   2275           LHLD  SAVE6
FAB6 C9         2280           RET
FAB7 CD E2 F9   2285 ARRAY     CALL  EVAL
FABA 22 AC 00   2290           SHLD  SAVE6
FABD 2A 8C 00   2295           LHLD  AMPR
FAC0 09         2300           DAD   B
FAC1 09         2305           DAD   B
FAC2 F1         2310           POP   PSW
FAC3 C9         2315           RET
FAC4 7E         2320 CONVP     MOV   A,M
FAC5 23         2325           INX   H
FAC6 F5         2330           PUSH  PSW
FAC7 FE 3A      2335           CPI   ':'
FAC9 CA B7 FA   2340           JZ    ARRAY
FACC 22 AC 00   2345           SHLD  SAVE6
FACF 21 20 00   2350           LXI   H,20H
FAD2 E6 3F      2355           ANI   3FH
FAD4 85         2360           ADD   L
FAD5 6F         2365           MOV   L,A
FAD6 29         2370           DAD   H
FAD7 F1         2375 OUTB      POP   PSW
FAD8 C9         2380           RET
FAD9 27 10      2385 PWRS10    DD    10000,1000,100,10,1
FADB 03 E8
FADD 00 64
FADF 00 0A
FAE1 00 01
FAE3 7E         2390 TSTN      MOV   A,M
FAE4 FE 3A      2395           CPI   '9'+1
FAE6 3F         2400           CMC
FAE7 D8         2405           RC
FAE8 FE 30      2410           CPI   '0'
FAEA C9         2415           RET
FAEB CD 20 FB   2420 CVTLN     CALL  INLN
FAEE CD E3 FA   2425 CVBIN     CALL  TSTN
FAF1 D8         2430           RC
FAF2 01 00 00   2435 CONT      LXI   B,0
FAF5 7E         2440 CBLOOP    MOV   A,M
FAF6 D6 30      2445           SUI   '0'
FAF8 81         2450           ADD   C
FAF9 4F         2455           MOV   C,A
FAFA 3E 00      2460           MVI   A,0
FAFC 88         2465           ADC   B
FAFD 47         2470           MOV   B,A
FAFE 23         2475           INX   H
FAFF CD E3 FA   2480           CALL  TSTN
FB02 3F         2485           CMC
FB03 D0         2490           RNC
FB04 E5         2495           PUSH  H
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

FB05 60         2500            MOV   H,B
FB06 69         2505            MOV   L,C
FB07 29         2510            DAD   H
FB08 29         2515            DAD   H
FB09 09         2520            DAD   B
FB0A 29         2525            DAD   H
FB0B 44         2530            MOV   B,H
FB0C 4D         2535            MOV   C,L
FB0D E1         2540            POP   H
FB0E C3 F5 FA   2545            JMP   CBLOOP
FB11 FE 40      2550 INLN6      CPI   '@'
FB13 CA 1D FB   2555            JZ    NEWLIN
FB16 23         2560            INX   H
FB17 7D         2565            MOV   A,L
FB18 FE 0D      2570            CPI   LINBUF+73%256
FB1A C2 2A FB   2575            JNZ   INLN2
FB1D CD 4B FB   2580 NEWLIN     CALL  CRLF
FB20 21 C5 00   2585 INLN       LXI   H,LINBUF+1
FB23 2B         2590 INLN5      DCX   H
FB24 7D         2595            MOV   A,L
FB25 FE C3      2600            CPI   LINBUF-1%256
FB27 CA 1D FB   2605            JZ    NEWLIN
FB2A CD 82 FB   2610 INLN2      CALL  INCH
FB2D 77         2615            MOV   M,A
FB2E FE 5F      2620            CPI   5FH
FB30 CA 23 FB   2625            JZ    INLN5
FB33 FE 0D      2630 INLN3      CPI   0DH
FB35 DA 2A FB   2635            JC    INLN2
FB38 C2 11 FB   2640            JNZ   INLN6
FB3B AF         2645 INLN4      XRA   A
FB3C 77         2650            MOV   M,A
FB3D 21 C4 00   2655            LXI   H,LINBUF
FB40 C3 50 FB   2660            JMP   LF
FB43 23         2665            INX   H
FB44 CD C3 F9   2670 STRNG      CALL  STRTMSG
FB47 7E         2675            MOV   A,M
FB48 FE 3B      2680            CPI   ';'
FB4A C8         2685            RZ
FB4B 3E 0D      2690 CRLF       MVI   A,0DH
FB4D CD B5 FB   2695            CALL  OUTCH
FB50 3E 0A      2700 LF         MVI   A,0AH
FB52 CD B5 FB   2705            CALL  OUTCH
FB55 3A 98 00   2710            LDA   COMA
FB58 3C         2715            INR   A
FB59 3D         2720 NULL       DCR   A
FB5A C8         2725            RZ
FB5B F5         2730            PUSH  PSW
FB5C AF         2735            XRA   A
FB5D CD B5 FB   2740            CALL  OUTCH
FB60 F1         2745            POP   PSW
FB61 C3 59 FB   2750            JMP   NULL
FB64 0D         2755 OKM        DB    0DH,0AH
FB65 0A
FB66 4F 4B      2760            DT    'OK'
FB68 00         2765            DB    0
FB69 3A 7C 00   2770 POLCAT     LDA   UP
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

FB6C 0F         2775            RRC
FB6D DA 79 FB   2780            JC    INP12
FB70 E6 04      2785            ANI   4
FB72 CA 7D FB   2790            JZ    INP0
FB75 DB 10      2795 INP10      IN    10H
FB77 0F         2800            RRC
FB78 C9         2805            RET
FB79 DB 12      2810 INP12      IN    12H
FB7B 0F         2815            RRC
FB7C C9         2820            RET
FB7D DB 00      2825 INP0       IN    0
FB7F 0F         2830            RRC
FB80 3F         2835            CMC
FB81 C9         2840            RET
FB82 CD 69 FB   2845 INCH       CALL  POLCAT
FB85 DA 98 FB   2850            JC    INA
FB88 3A 9C 00   2855            LDA   PERD
FB8B E6 02      2860            ANI   2
FB8D CA 82 FB   2865            JZ    INCH
FB90 DB 06      2870            IN    6
FB92 0F         2875            RRC
FB93 DA 82 FB   2880            JC    INCH
FB96 DB 07      2885            IN    7
FB98 C3 C8 FB   2890            JMP   OUTE
FB9B 3A 7C 00   2895 INA        LDA   UP
FB9E 0F         2900            RRC
FB9F DA AC FB   2905            JC    IN12
FBA2 E6 04      2910            ANI   4
FBA4 CA B1 FB   2915            JZ    IN0
FBA7 DB 11      2920 IN10       IN    11H
FBA9 C3 B3 FB   2925            JMP   OUTCH-2
FBAC DB 13      2930 IN12       IN    13H
FBAE C3 B3 FB   2935            JMP   OUTCH-2
FBB1 DB 01      2940 IN0        IN    1
FBB3 E6 7F      2945            ANI   7FH
FBB5 F5         2950 OUTCH      PUSH  PSW
FBB6 3A 9C 00   2955            LDA   PERD
FBB9 E6 04      2960            ANI   4
FBBB CA CC FB   2965            JZ    OUTA
FBBE DB 06      2970 OUTC       IN    6
FBC0 3C         2975            INR   A
FBC1 CA CC FB   2980            JZ    OUTA
FBC4 07         2985            RLC
FBC5 DA BE FB   2990            JC    OUTC
FBC8 F1         2995            POP   PSW
FBC9 D3 07      3000            OUT   7
FBCB F5         3005 OUTE       PUSH  PSW
FBCC 3A 9C 00   3010 OUTA       LDA   PERD
FBCF 0F         3015            RRC
FBD0 D2 D7 FA   3020            JNC   OUTB
FBD3 3A 7C 00   3025            LDA   UP
FBD6 0F         3030            RRC
FBD7 DA EA FB   3035            JC    OUT12
FBDA E6 04      3040            ANI   4
FBDC CA F5 FB   3045            JZ    OUT0
FBDF DB 10      3050 OUT10      IN    10H
```

```
ADDR B1 B2 B3 E LINE  LABEL     OPCD  OPERAND

FBE1 E6 02      3055            ANI   2
FBE3 CA DF FB   3060            JZ    OUT10
FBE6 F1         3065            POP   PSW
FBE7 D3 11      3070            OUT   11H
FBE9 C9         3075            RET
FBEA DB 12      3080 OUT12      IN    12H
FBEC E6 02      3085            ANI   2
FBEE CA EA FB   3090            JZ    OUT12
FBF1 F1         3095            POP   PSW
FBF2 D3 13      3100            OUT   13H
FBF4 C9         3105            RET
FBF5 DB 00      3110 OUT0       IN    0
FBF7 07         3115            RLC
FBF8 DA F5 FB   3120            JC    OUT0
FBFB F1         3125            POP   PSW
FBFC D3 01      3130            OUT   1
FBFE C9         3135            RET
FBFF            3140 LASTM      EQU   $-1
```