

A^{File}

BOB WISEMAN
118 ST. ANDREWS DR.
CINCINNATI, OHIO 45245

JAN 12, 1981

DEAR ROBERT FABRIS,

ENCLOSED YOU SHOULD FIND A TAPE WITH ONE PROGRAM ON EACH SIDE. SIDE ONE HAS A NEW LANGUAGE ON IT, BALLY FORTH. THIS IS A SIMPLE FORTH-LIKE LANGUAGE WHICH IS INTERPRETED BY BALLY BASIC. IT IS VERY DIFFERENT FROM BASIC AND SOME MENTAL REPROGRAMMING MAY BE NECESSARY. THE OTHER SIDE OF THE TAPE CONTAINS THE CLASSIC TOWER OF HANOI PUZZLE.

I HOPE THAT YOU AND YOUR READERS ENJOY THESE AS MUCH AS I DO.

YERZ,

WISEMAN

I just received your note regarding the (LONG-AWAITED) ASTROVISION ADD-ON. I am most interested in more information.

I feel that I have excellent qualifications for putting such a device through its paces and reporting back on its abilities.

W. _____

to check for content

```

*****
***                                     ***
***           B A L L Y   F O R T H           ***
***                                     ***
*****

```

I HOPE THAT I OFFEND NO ONE BY CALLING THIS "FORTH". THE SIMILARITIES ARE PRESENT BUT ARE LIMITED. PLEASE UNDERSTAND THAT THIS IS A EENSY-WEENSY VERSION.

NONETHELESS, FOR THOSE OF YOU WISHING A NEW CHALLENGE, A CHANGE OF PACE FROM THE VULGARITIES OF TINY BASIC, HERE IS A NEW LANGUAGE THAT YOU CAN FRET OVER.

THIS LANGUAGE OPERATES ON A STACK PRINCIPLE. YOU DO NOT HAVE ANY DATA NAMES (LIKE "A"), YOU HAVE A STACK OF DATA, MUCH LIKE A STACK OF PLAYING CARDS. FIRST YOU ENTER YOUR PROGRAM AND YOUR PROGRAM CREATES AND MANIPULATES THIS STACK OF NUMBERS.

EACH NUMBER IN THE STACK MUST BE A NUMBER THAT IS VALID IN BALLY BASIC. THESE ARE STORED IN THE AT-SIGN STRING, BUT MORE ABOUT THIS LATER. FIRST, LET US LOOK AT THE OPERATIONS THAT ARE AVAILABLE.

MY KEYBOARD DOES NOT CONTAIN MANY OF THE SYMBOLS USED IN BALLY FORTH. PLEASE PARDON THIS INCONVENIENCE.

THE FOLLOWING COMMANDS ADD AN ELEMENT TO THE TOP OF THE STACK. EACH OF THESE INCREASES THE SIZE OF THE STACK BY ONE.

- V (DOWN ARROW) PUTS THE NUMBER ZERO ON THE TOP OF THE STACK.
- Q PUTS THE SIZE OF THE STACK ON THE TOP OF THE STACK.
- I TEMPORARY HALT. REQUEST THAT A NUMBER BE ENTERED THROUGH THE KEYPAD AND PLACED ON THE TOP OF THE STACK.
- D DUPLICATES THE TOP OF THE STACK ON THE TOP OF THE STACK.

THE FOLLOWING COMMANDS DELETE AN ELEMENT FROM THE TOP OF THE STACK. EACH OF THESE DECREASES THE SIZE OF THE STACK BY ONE.

- ^ (UP ARROW) DELETE THE TOP OF THE STACK.
- + REPLACE TOP TWO ELEMENTS OF THE STACK WITH THEIR SUM.
- * (MUTIPLY SIGN) REPLACE TOP TWO ELEMENTS OF THE STACK WITH THEIR PRODUCT
- / (DIVIDE SIGN) REPLACE TOP TWO ELEMENTS OF THE STACK WITH THE TOP-1 DIVIDED BY THE TOP ELEMENT.
- (SUBTRACT) REPLACE THE TOP TWO ELEMENT OF THE STACK WITH THE TOP-1 MINUS THE TOPMOST ELEMENT.

THESE COMMANDS CHANGE THE TOP ELEMENT OF THE STACK. THE NUMBER OF STACK ELEMENTS REMAINS THE SAME.

- X EXCHANGE THE TOP TWO ELEMENTS OF THE STACK.

- B TAKE THE TOP ELEMENT OF THE STACK. COUNT THAT MANY ELEMENTS BACK INTO THE STACK AND PUT THAT VALUE ON THE TOP OF THE STACK REPLACING WHATEVER WAS THERE.
- O THRU 9 MULTIPLY THE TOP OF THE STACK BY 10 AND ADD THE NUMBER.
- T WHEN A T IS ENCOUNTERED, NO ACTION IS TAKEN.

THESE COMMANDS CAN CHANGE WHICH OPERATION IS DONE NEXT. THESE DO NOT AFFECT THE STACK.

- ? (QUESTION MARK) THIS TESTS THE TOP OF THE STACK FOR ZERO. IF THE TOP OF THE STACK IS ZERO, THE NEXT INSTRUCTION IS THE ONE IMMEDIATELY FOLLOWING THE QUESTION MARK. IF THE TOP OF THE STACK IS NON-ZERO, THE NEXT INSTRUCTION IS SKIPPED.
- < (LEFT ARROW) THE NEXT INSTRUCTION IS THE FIRST "T" TO THE LEFT OF THE CURRENT INSTRUCTION. THIS IS A GO TO.
- > (RIGHT ARROW) GO TO THE FIRST "T" TO THE RIGHT. THE "T" CHARACTER CAN BE THOUGHT OF AS A "TAB STOP" FOR JUMPING RIGHT OR LEFT.
- S STOPS THE PROGRAM
- P PRINT THE TOP OF THE STACK WHERE IT SAYS: DATA.
- . STOPS THE PROGRAM WHEN ENCOUNTERED. THIS ALSO STOPS THE ENTRY SECTION OF THE PROGRAM AND BEGINS EXECUTION.

WE ALWAYS BEGIN WITH A ZERO ON THE TOP OF THE STACK WITH A STACK DEPTH OF ONE. AS EACH INSTRUCTION IS EXECUTED, ITS OPERATION IS PRINTED BESIDE THE "OP". ITS POSITION IN THE PROGRAM (RELATIVE TO 1) IS PRINTED BESIDE THE "IN#". AFTER THE INSTRUCTION IS COMPLETE, THE DEPTH OF THE STACK IS PRINTED BESIDE THE "ST#", AND THE DATA ON THE TOP OF THE STACK IS PRINTED BESIDE THE "TOP".

ENTER THE PROGRAM. IT MUST END WITH A PERIOD. IT WILL RUN. WHEN IT STOPS, HIT ANY KEY, AND YOU CAN ENTER ANOTHER PROGRAM. THOSE OF YOU WHO ARE SO INCLINED, MIGHT WANT TO CHANGE THIS TO ALLOW EDITING AND MODIFICATION OF THE LAST PROGRAM. YOU MIGHT ALSO WANT TO DEVISE A "SINGLE-STEP" MODE FOR DEBUGGING. SINCE EACH "GOSUB 100" EXECUTES THE NEXT INSTRUCTION, THIS WOULD NOT BE DIFFICULT.

PERHAPS SOME EXAMPLES WOULD MORE EASILY SHOW HOW THIS LANGUAGE FUNCTIONS. HERE ARE A FEW SIMPLE ONES TO GIVE YOU AN IDEA HOW THE OPERATIONS CAN BE MADE INTO A PROGRAM.

EXAMPLE ONE: INPUT A NUMBER, ADD ONE AND PRINT THE RESULT.

IV1+P.

EXAMPLE TWO: INPUT A NUMBER, SQUARE IT PRINT THE RESULT.

ID*.

EXAMPLE THREE: INPUT A NUMBER, IF IT IS ZERO, STOP. IF IT IS NOT ZERO ~~PRINT ITS RECIPROCAL~~

TRIPLE + PRINT

I?S↓3×P.

~~EXAMPLE FOUR:~~

EXAMPLE FOUR: INPUT A NUMBER, IF IT IS EVEN, PRINT 0. IF IT IS ODD PRINT 1.

IDV2/V2*-?>V1PSTP.

EXAMPLE FIVE: INPUT A NUMBER, ADD TO IT ALL NUMBER LESS THAN IT.

IDTXV1-?>XV2B+<TXP.

EXAMPLE SIX: MYSTERY PROGRAM.

VITDV1-?><T^T*X?><TXPS.

PLEASE REMEMBER THAT I HAVE USED THE FOLLOWING REPLACEMENTS:

V = DOWN ARROW
< = LEFT ARROW
> = RIGHT ARROW
^ = UP ARROW
* = MULTIPLY SIGN
/ = DIVIDE SIGN

HERE ARE SOME NOTES ON THE GUTS FOR THOSE OF YOU WHO IMPATIENT TO CHANGE THE PROGRAM.

LINES 10 - 99 MAINLINE LOGIC
100 - 499 HANDLES THE EXECUTION OF YOUR PROGRAM
500 - 700 BELLS AND WHISTLES. PRINTS TOP OF STACK.
700 - 800 ENTERS AND STORES YOUR PROGRAM
800 - 900 STARTUP, PRINTS SCREEN

WHEN YOU ENTER YOUR PROGRAM IT IS STORED IN ASCII IN THE AT-SIGN STRING STARTING IN LOCATION ZERO THRU "N". AT EXECUTION TIME, THE STACK IS BUILT ON THE AT-SIGN STRING FOLLOWING YOUR PROGRAM. THESE DATA NAMES ARE USED TO KEEP TRACK OF THIS.

T = ABSOLUTE TOP OF STACK POINTER
N = TOP OF PROGRAM POINTER
C = CURRENT OPERATION POINTER
A = CURRENT OPERATION (IN ASCII)
S = STOP FLAG. SET TO ONE TO STOP PROGRAM.

OK, SO NOW YOU SAY "ALL OF THIS IS FINE BUT I WANT TO ADD MY OWN COMMAND TO THIS". NO PROBLEM. THIS IS EASILY EXTENDED. FOR AN EXAMPLE, WE WILL CREATE THE COMMAND "=" (WHICH IS NOT IN THE LANGUAGE). "=" WILL MEAN THE FOLLOWING:

IF THE TOP TWO ELEMENTS OF THE STACK ARE EQUAL, DO THE NEXT OPERATION. OTHERWISE SKIP THE NEXT OPERATION.

WE FIND A BLANK LINE BETWEEN 100 AND 330 AND KEY IN:

XXX IF (A="=")*(&#(T)<>&#(T-1)):C=C+1:RETURN

XXX IF A=""RETURN

(FALLING THROUGH RESULTS IN THE PROGRAM STOPPING WITH AN ILLEGAL OPERATION CODE)

AGAIN PARDON THE USE OF # INSTEAD OF AT-SIGN BUT THAT IS ALL THAT I HAVE.

```
*****:
***                                     ***
***   T O W E R   O F   H A N O I   ***
***                                     ***
*****
```

YOU WILL BE ASKED HOW MANY BOXES YOU WANT IN THE STACK. PLEASE ANSWER THROUGH THE KEYBOARD WITH A NUMBER FROM 2 TO 7. THE OBJECT IS TO RESTACK THESE BOXES IN THE CENTER OR ON THE RIGHT.

THE RESTACKING IS DONE BY LIFTING A BOX AND SETTING IT DOWN. A LARGER BOX MAY NOT BE PLACED ON A SMALLER BOX. TO INDICATE WHICH STACK YOU WANT TO LIFT FROM OR PLACE ON USE CONTROL NUMBER ONE. JX = -1 INDICATES THE LEFT STACK, JY = +1 IS FOR THE CENTER, AND JX = +1 IS FOR THE RIGHT STACK. YOU WILL BE RATED AT THE END, AND YOU WILL BE GIVEN ONE MORE BOX TO PLAY WITH.

```
*****
***                                     ***
***   F I N I S H   ***
***                                     ***
*****
```

ARCADIAN

```

2 .
3 .
4 .
5 . BALLY FORTH
6 . BY BOB WISEMAN
7 . C 1981
10 GOSUB 800;GOSUB 700
25 T=N+1;@(T)=0;C=0;NT=0
30 GOSUB 100;IF SGOTO 60
50 GOSUB 500;GOTO 30
60 CY=0;CX=-54;PRINT "STOP
70 C=KP;GOTO 10
100 A=@(C);S=0
110 CY=24;CX=-61
120 PRINT #2,C+1,;CX=-12;TV=@(C)
130 PRINT ;C=C+1
140 IF (A)>=48)b(A<=57)@(T)=@(T)b10+A-48;RETURN
150 IF A=88A=@(T);@(T)=@(T-1);@(T-1)=A;RETURN
160 IF A=68T=T+1;@(T)=@(T-1);RETURN
170 IF A=84RETURN
180 IF A=83S=1;RETURN
190 IF A=80CY=8;CX=-12;PRINT #5,@(T);CY=16;RETURN
200 IF A=94T=T-1;RETURN
210 IF A=96T=T+1;@(T)=0;RETURN
220 IF A=43T=T-1;@(T)=@(T+1)+@(T);RETURN
230 IF A=45T=T-1;@(T)=@(T)-@(T+1);RETURN
240 IF A=98T=T-1;@(T)=@(T+1)b@(T);RETURN
250 IF A=99T=T-1;@(T)=@(T+1)c@(T);RETURN
260 IF A=73T=T+1;CY=0;CX=-54;INPUT "INPUT "@(T);CY=16;BOX -36,0,42,9,1;RETURN
270 IF (A=63)b(@(T)=0)RETURN
280 IF A=63C=C+1;RETURN
290 IF A=97GOTO 400
300 IF A=95GOTO 450
305 IF A=81T=T+1;@(T)=T-N-1;RETURN
310 IF A=66@(T)=@(T-@(T));RETURN
320 CX=0;CY=-16
330 PRINT "ILG OP"
340 S=1;RETURN
400 FOR C=CTO N;IF @(C)=84RETURN
410 NEXT C
420 C=0;RETURN
450 FOR C=C-2TO 0STEP -1;IF @(C)=84RETURN
460 NEXT C
470 RETURN
500 CX=-61
510 PRINT #2,T-N,;CX=-12;PRINT #5,@(T)
530 BOX 23+RND (50),RND (40),2,4,2
540 NT=3;MU=@(C);NT=0
550 RETURN
700 CY=-22;NT=3
710 PRINT "ENTER PROGRAM
720 A=KP;TV=A
730 IF A=31N=N-1;GOTO 720
735 IF A=46@(N)=83;RETURN
740 @(N)=A;N=N+1;GOTO 720
800 CLEAR ;C=0;N=0;NT=0
805 CY=37
810 PRINT " BALLY FORTH
815 CY=24
820 PRINT "IN# OP
830 PRINT "ST# TOP
840 PRINT " DATA
860 BOX 0,18,160,52,3
900 RETURN

```

```

1 .
2 .
3 .
4 .
5 .FORTH WISEMAN
10 GOSUB 800;GOSUB 700
25 T=N+1;@(T)=0;C=0;NT=0
30 GOSUB 100;IF SGOTO 60
50 GOSUB 500;GOTO 30
60 CY=0;CX=-54;PRINT "STOP
70 C=KP;GOTO 10
100 A=@(C);S=0
110 CY=24;CX=-61
120 PRINT #2,C+1,;CX=-12;TV=@(C)
130 PRINT ;C=C+1
140 IF (A>=48)b(A<=57)@(T)=@(T)b10+A-48;RETURN
150 IF A=88A=@(T);@(T)=@(T-1);@(T-1)=A;RETURN
160 IF A=68T=T+1;@(T)=@(T-1);RETURN
170 IF A=84RETURN
180 IF A=83S=1;RETURN
190 IF A=80CY=8;CX=-12;PRINT #5,@(T);CY=16;RETURN
200 IF A=94T=T-1;RETURN
210 IF A=96T=T+1;@(T)=0;RETURN
220 IF A=43T=T-1;@(T)=@(T+1)+@(T);RETURN
230 IF A=45T=T-1;@(T)=@(T)-@(T+1);RETURN
240 IF A=98T=T-1;@(T)=@(T+1)b@(T);RETURN
250 IF A=99T=T-1;@(T)=@(T+1)c@(T);RETURN
260 IF A=73T=T+1;CY=0;CX=-54;INPUT "INPUT "@(T);CY=16;BOX -36,0,42,9,1;RETURN
270 IF (A=63)b(@(T)=0)RETURN
280 IF A=63C=C+1;RETURN
290 IF A=97GOTO 400
300 IF A=95GOTO 450
305 IF A=81T=T+1;@(T)=T-N-1;RETURN
310 IF A=66@(T)=@(T-@(T));RETURN
320 CX=0;CY=-16
330 PRINT "ILG OP"
340 S=1;RETURN
400 FOR C=CTO N;IF @(C)=84RETURN
410 NEXT C
420 C=0;RETURN
450 FOR C=C-2TO 0STEP -1;IF @(C)=84RETURN
460 NEXT C
470 RETURN
500 CX=-61
510 PRINT #2,T-N,;CX=-12;PRINT #5,@(T)
530 BOX 23+RND(50),RND(40),2,4,2
540 NT=3;MU=@(C);NT=0
550 RETURN
700 CY=-22;NT=3
710 PRINT "ENTER PROGRAM
720 A=KP;TV=A
730 IF A=31N=N-1;GOTO 720
735 IF A=46@(N)=83;RETURN
740 @(N)=A;N=N+1;GOTO 720
800 CLEAR ;C=0;N=0;NT=0
805 CY=37
810 PRINT " BALLY FORTH
815 CY=24

```

```
820 PRINT "IN#      OP
830 PRINT "ST#      TOP
840 PRINT "          DATA
860 BOX 0,18,160,52,3
900 RETURN
```

>RUN

BALLY FORTH

~~IN# OP
ST# TOP~~

RUN

BALLY BASIC

>RUN

BALLY FORTH

IN# OP
ST# TOP
 DATA

ENTER PROGRAM

0123456789:;<=>?@ABCDEFGHIJKLMNPOQRSTUVWXYZ[\]↑ `abcdefghijklmnopqrsuvwxyz{|}~0