Brainfuck interpreter for ZX81

Language

Brainfuck has only few rules.

- There are only 8 commands `+ - > < [ ] , .`
- Each command is represented by a single letter.
- There are no parameters.
- The workspace is an array of cells.
- There is a pointer which selects the current cell.

On program-start all cells are 0 and the pointer selects the first cell.

Commands

In the following table all eight commands `+ - > < [ ] , .` are listed with comments.

Since there are no squared brackeds at the ZX81 we use round brackets instead.

classic	ZX81	
+	+	add 1 to the current cell
-	-	subtract 1 from the current cell
>	>	move the pointer to the next cell
<	<	move the pointer to the previous cell
[	(	beginning of loop
]	)	end of loop
.	.	PRINT current cell
,	,	INPUT current cell

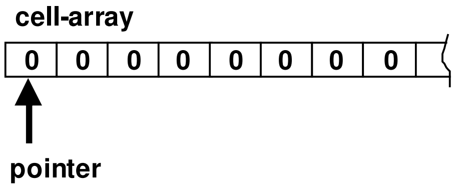
Comments

Any character that is not one of the eight commands is a comment and will be ignored by the interpreter. Therefore you can add comments at any position in the program.

Workspace

The workspace is a long array of cells with a pointer selecting the current cell.

On program-start all cells are 0 and the pointer selects the first cell.



Cells

Each cell has an integer signless numeric value represented by a 16 bit number between 0 and 65535

Add and subtract

With commands + and − the current cell will be added or subtracted by 1.

Adding above 65535 or subtracting below 0 will cause an error!

Pointer

The pointer can be incremented with the > an decremented with the < command.

Decrementing the pointer below the first cell will cause an error!

Loops

A program-loop is made with brackets. The part of the program inside the brackets will be either ignored or processed and optional repeated at the end. This is controlled by the value of the current cell.

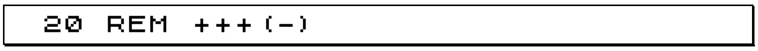
The loop will be ignored or not repeated if the value is 0.

In all other cases the loop will be processed or repeated at the end.

Loops may be nested. Opening and closing brackets have to be balanced!

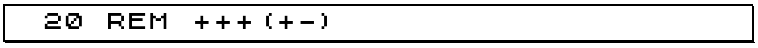
Examples

This loop will be processed 3 times:



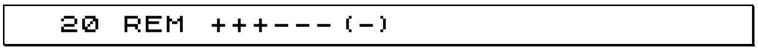
```
20 REM +++(-)
```

This loop will be processed infinitely:



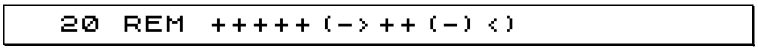
```
20 REM +++(+-)
```

This loop will be ignored:



```
20 REM +++---(-)
```

The inner loop will be processed 10 times:



```
20 REM +++++(->++(-)<)
```

Input and output

Input and output of characters will be in ASCII-code because of compatility reasons.

Any value of a cell will be output as ASCII-character. The software converts each character to the ZX81 character set. Characters that are not within the ZX81 character set will be shown as a questionmark.

The software converts any entered key to an ASCII-character and inputs this value to the current cell. Therefore you can only enter ASCII-values that are represented by a ZX81-key.

Entering the Enter-key will give in the value 0.

ZX81-implementation

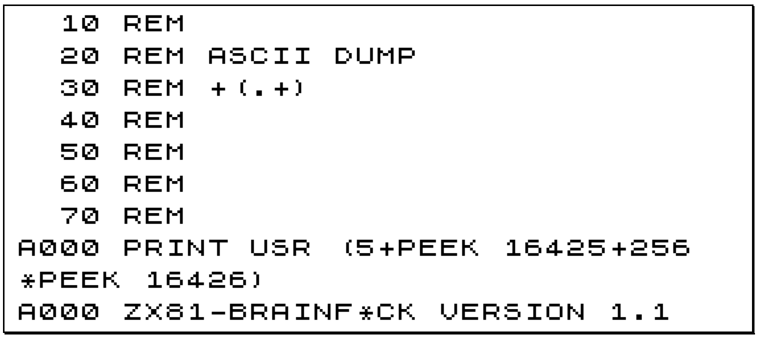
The actual Brainfuck-interpreter for ZX81 can be downloaded from the authors website:

http://www.swatosch.de/zx81/brainfuck.p

Programming

In this version for the ZX81 the program lines have to be written within Basic-REM-lines. Empty REM-lines will be ignored. The program can be split into several REM-lines. You can add or delete REM-lines just as needed.

Comments and program can be mixed as long as none of the eight commands are used in the comments.



```
  10 REM
  20 REM ASCII DUMP
  30 REM +(.+)
  40 REM
  50 REM
  60 REM
  70 REM
A000 PRINT USR (5+PEEK 16425+256
*PEEK 16426)
A000 ZX81-BRAINF*CK VERSION 1.1
```

At the end of the listing is one line containing the jump-in to the Brainfuck-interpreter and a following line holding the machinecode of the Brainfuck-interpreter.

Do not delete or edit these lines!

Run

To start the program simply enter RUN. A running program can be stopped with the BREAK-key at any time.

The program shows error-messages in the common way at the bottom line of the screen. Additionally it displays a readable errormessage in the same line.

If the program finishes without any error the 0-message folllowed by the last number of REM-line will be displayed.

Screen-output

On program-start the screen will be cleared. The text-output will start on the top left corner of the screen.

When printing a ASCII carriage-return-character the print position moves to the beginning of the next line.

As soon as the screen is completely filled the screen will be scrolled up a line automaticly.

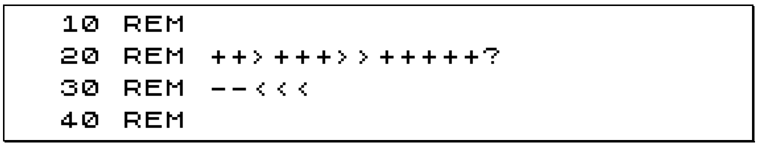
Debugging

There are two additional commands for debugging and studying of the language.

The `?` command initiates a cell-dump to the screen. The current cell will be marked blinking. The interpreter then waits for an input with the ENTER-key.

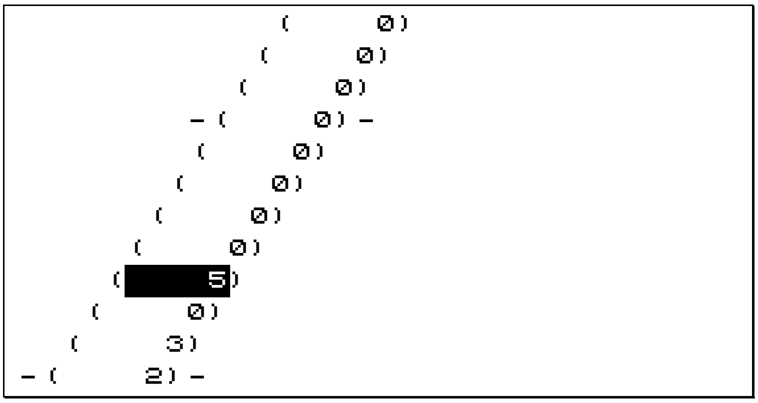
The `??` command starts a continuouse single step cell-dump. This is for investigating a program and the cells. This mode ends with the simple `?` command.

The following program:



```
  10 REM
  20 REM ++>+++>>+++++?
  30 REM --<<<
  40 REM
```

leads to the following cell-dump:



```
            (    0)
           (    0)
          (    0)
        -(    0)-
        (    0)
       (    0)
      (    0)
     (    0)
    (    5)
   (    0)
  (    3)
-(    2)-
```

Every eighth cell will be a mark to help counting.

Normally the bottom cell will be the first cell of the array. If the pointer moves up the displayed cells will shift to show all the cells around the current cell.

Example programs

The following listings show the Brainfuck-lines embedded in Basic-REM-lines.

ASCII-dump

The complete ASCII-character-set will print out…

```
  20 REM +(.+)
```

Echo-loop

Any character typed in will be printed to the screen.

```
  20 REM +(,.)
```

10-times-loop

This loop runs 10 times.

```
  20 REM ++++++++++
  30 REM (->,.<)
```

String-in-out

The typed in string will be stored and typed out at the end.

```
  20 REM >,(>,)
  30 REM <(<)
  40 REM >(.>)
```

Bubble-sort

All typed in characters will be sorted and printed out then.

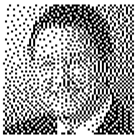
```
  20 REM >>>>>,.(>>>,.)
  30 REM <<<
  40 REM (<<<
  50 REM (>>>
  60 REM (-<<<-<+>(>)>>)
  70 REM <<<(<)>>
  80 REM (>>>+<<<-)<
  90 REM (>+>>>+<<<<-)
 100 REM <<)
 110 REM >>>(.(-))
 120 REM >>>(>>>)<<<)
```

Author

This document and the used program for the ZX81 can be downloaded at the authors website.

Status: April 2010

http://www.swatosch.de/zx81/brainfuck-ge.pdf
http://www.swatosch.de/zx81/brainfuck-en.pdf
http://www.swatosch.de/zx81/brainfuck.p



Matthias Swatosch
Grünewalder Berg 45
42105 Wuppertal
Germany

http://www.swatosch.de
malto:info@swatosch.de