

## jaz abstract machine instructions

### Stack Manipulation

<b>push c</b>	Pushes c onto the stack
<b>rvalue l</b>	Pushes contents of data location l onto the stack
<b>lvalue l</b>	Pushes address of data location l onto the stack
<b>pop</b>	Throws away value on top of the stack
<b>:=</b>	Stack top is placed by the lvalue below it and both are popped
<b>copy</b>	Pushes a copy of the top value on stack

### Control Flow

<b>label l</b>	Targets of jumps to l
<b>goto l</b>	Next instruction is taken from statement with label l
<b>gofalse l</b>	Pops the top value of the stack and jumps if it is zero
<b>gotrue l</b>	Pops the top value of the stack and jumps if it is nonzero
<b>halt</b>	Stops execution

### Arithmetic Operators

<b>+</b>	Adds top two values on stack and places result on stack
<b>-</b>	Similar to +, but subtraction is performed
<b>*</b>	Similar to +, but multiplication is performed
<b>/</b>	Similar to +, but integer division is performed
<b>div</b>	Similar to +, but remainder of division is performed

### Logical Operators

<b>&amp;</b>	Logical AND the top two values on stack and places result on stack
<b>!</b>	Negates the top of the stack
<b> </b>	Similar to &, but logical OR is performed

### Relational Operators

<b>&lt;&gt;</b>	Returns 0 if top two values on stack are equal otherwise returns 1
<b>&lt;=</b>	Similar to <>, but tests if top minus one is less or equal top
<b>&gt;=</b>	Similar to <>, but tests if top minus one is greater or equal top
<b>&lt;</b>	Similar to <=, but tests if top minus one is less than top
<b>&gt;</b>	Similar to >=, but tests if top minus one is greater than top
<b>=</b>	Similar to <>, but tests if top minus one is equal to top of stack

### Output

<b>print</b>	Writes top of the stack contents to output device
<b>show</b>	Writes a literal string to output device

## Subprogram Control

<b>begin</b>	Marks the beginning of parameter passing and subroutine call
<b>end</b>	Marks the end of parameter passing and subroutine call
<b>return</b>	Returns from subroutine
<b>call</b>	Subroutine call