A-Instruction

value (v = 0 or 1)															
0	v	v	v	v	v	v	v	v	v	v	v	v	v	v	v

C-Instruction

			computation							destination			jump		
1	1	1	a	c1	c2	c3	c4	c5	c6	d1	d2	d3	j1	j2	j3

when a=0	c1	c2	c3	c4	c5	c6	when a=1
0	1	0	1	0	1	0	
1	1	1	1	1	1	1	
-1	1	1	1	0	1	0	
D	0	0	1	1	0	0	
A	1	1	0	0	0	0	M
!D	0	0	1	1	0	1	
!A	1	1	0	0	0	1	!M
-D	0	0	1	1	1	1	
-A	1	1	0	0	1	1	M+1
D+1	0	1	1	1	1	1	
A+1	1	1	0	1	1	1	
D-1	0	0	1	1	1	0	
A-1	1	1	0	0	1	0	M-1
D+A	0	0	0	0	1	0	D+M
D-A	0	1	0	0	1	1	D-M
A-D	0	0	0	1	1	1	M-D
D&A	0	0	0	0	0	0	D&M
D|A	0	1	0	1	0	1	D|M

d1	d2	d3	Mnemonic	Where to store the computed value
0	0	0	null	Not stored anywhere
0	0	1	M	Memory[A] (memory register addressed by A)
0	1	0	D	D register
0	1	1	MD	Memory[A] and D register
1	0	0	A	A register
1	0	1	AM	A register and Memory[A]
1	1	0	AD	A register and D register
1	1	1	AMD	A register, Memory[A], and D register

j1 (out < 0)	j2 (out = 0)	j3 (out > 0)	Mnemonic	Effect
0	0	0	null	No jump
0	0	1	JGT	if out > 0 jump
0	1	0	JEQ	if out = 0 jump
0	1	1	JGE	if out ≥ 0 jump
1	0	0	JLT	if out < 0 jump
1	0	1	JNE	if out ≠ 0 jump
1	1	0	JLE	if out ≤ 0 jump
1	1	1	JMP	jump

	Example Computation Instructions
D=A	Set D register to A register.
AD=A+1	Increment A register and set D register to incremented value.
M=~M	Invert RAM[A].
M=M-D	Subtract D register from RAM[A].
AMD=0	Set RAM[A] and A and D registers to zero.
0;JMP	Unconditional jump to ROM address in A register.
D;JNE	Jump to ROM address in A register if D register is non-zero.
D=D-1;JGT	Decrement D register, then jump to ROM address in A register if D register is greater than zero.
M=M-1;JGT	Not useful because the A register must be used for both the RAM address and the target jump address.

Hack Platform Keyboard Codes

Key Pressed	Code
newline	128
backspace	129
left arrow	130
up arrow	131
right arrow	132
down arrow	133
home	134
end	135
page up	136
page down	137
insert	138
delete	139
esc	140
f1-f12	141-152

Arithmetic Instructions

Operation	a=	a=0	a=1
Constants	0	1	-1
Echo	D	A	M
Negative	-D	-A	-M
Increment	D+1	A+1	M+1
Decrement	D-1	A-1	M-1
Addition		D+A	D+M
Subtraction		D-A	D-M
Inv. Subtraction		A-D	M-D
Not	!D	!A	!M
And		D&A	D&M
Or		D&M	D|M

Symbol	Address	Definition	Purpose
R0 - R15	0 - 15	Virtual Registers	Convenience
SP	0	Predefined Pointer	Stack Pointer
LCL	1	Predefined Pointer	
ARG	2	Predefined Pointer	Function Argument
THIS	3	Predefined Pointer	
THAT	4	Predefined Pointer	
SCREEN	16384 (0x4000)	I/O Pointer	Screen Buffer
KBD	24576 (0x6000)	I/O Pointer	Keyboard Buffer