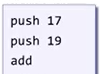
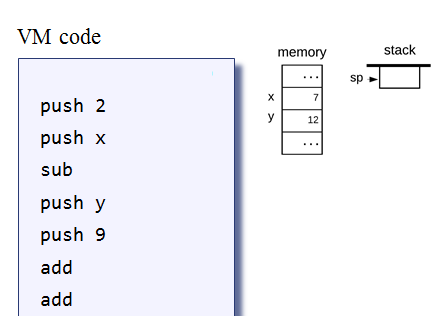
Name: Date:

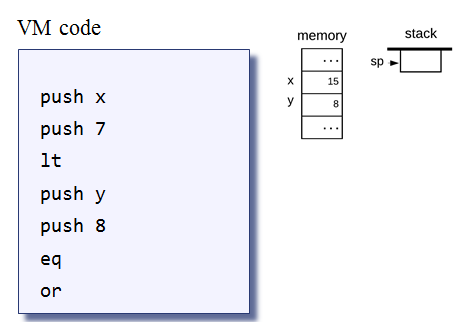
1. Stack Arithmetic Commands: What the state of the stack and the memory after the following VM code is executed. Where will the stack pointer (sp) end up, if it originally begins at address 256? Please illustrate the stack after every VM command has been executed.



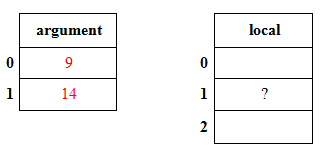
1. Stack Arithmetic Commands: What the state of the stack and the memory after the following VM code is executed. Where will the stack pointer (sp) end up, if it originally begins at address 256? Please illustrate the stack after every VM command has been executed.



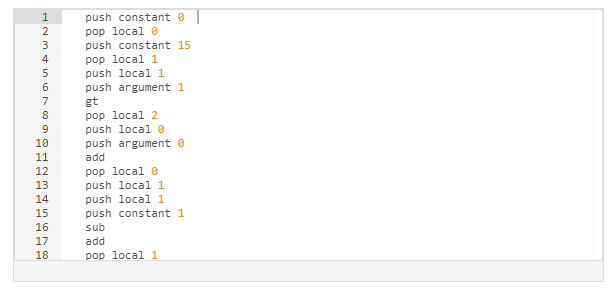
1. Stack Logical Commands: Again, what the state of the stack and memory after the following VM code is executed. Where will the stack pointer (sp) end up, if it originally begins at address 256? Please illustrate the stack after every VM command has been executed.



1. Suppose the state of the argument and local memory segments are as follows:

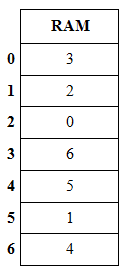


Now consider the following VM code:



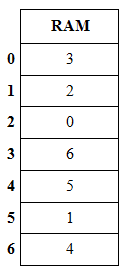
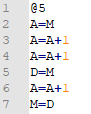
**What will be the value of local 1 after the VM code has executed?\_\_\_\_\_\_\_\_\_\_\_\_**

1. Suppose the state of the RAM is as follows and the adjacent assembly code will execute:

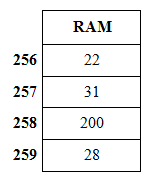
**What will be the value of the RAM[4] following the assembly code execution?\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. Suppose the state of the RAM is as follows and the adjacent assembly code will execute:

**What will be the value of the RAM[4] following the assembly code execution?\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

1. Suppose the state of the RAM is as follows and the adjacent pseudocode (like C++) will execute:

**What will be the value of the RAM[258] following the assembly code execution?\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Translate the following VM commands to Assembly instructions:**

* push constant 1
* push constant 5
* add
* pop static 7 //suppose inside of a file named **Add**
* pop local 2
* eq