Question 1

1 / 1 pts

Which of the following operators are provided by the Hack Virtual Machine?

- [x] and
- [x] lt
- [ ] ge
- [ ] multiply
- [ ] divide
- [x] gt
- [ ] xor
- [x] neg
- [x] sub
- [ ] le
- [x] add

Arithmetic/Boolean commands:

- add
- sub
- neg
- eq
- gt
- lt
- and
- or
- not

Question 2

1 / 1 pts

What are each of the following memory segments used for in the Hack Virtual machine?

static	holds values of class static va
argument	holds the arguments for the c
local	holds the local variables for th
constant	represents all the constants in
pointer	used to change the start addr
temp	fixed 8-entry segment that ho
field	this is not a segment
var	this is not a segment

result

this is not a segment

Question 3

1 / 1 pts

Consider the following **Jack** class:

```
class bob
{
    function int foo(int a)
    {
            var int x;
            let x = a + x ;
            return x ;
    }
}
```

What are the first three virtual machine commands that implement function foo?

```
function bob.foo 1
push constant 0
push argument 0
```

```
function bob.foo 1
push argument 0
push local 0
```

```
function bob 1
push argument 0
push local 0
```

```
push argument 0
push local 0
add
```

```
function bob.foo 1
push argument 0
pop pointer 0
```

Question 4

1 / 1 pts

Consider the following **Jack** function:

```
function int foo()
{
    var a,b,sum ;

    let b = 10 ;
    while ( a < b )
    {
        let sum = sum + a ;
        let a = a + 1 ;
    }
    return sum ;
}
```

What sequence of virtual machine commands will implement the two let statements in the body of the while loop?

```
push sum
push a
add
push a
push 1
add
```

```
push local 2
push local 0
add
push local 0
push constant 1
add
```

```
push local 2
push local 0
add
pop local 2
push local 0
push constant 1
add
pop local 0
```

```
push local 3
push local 1
add
pop local 3
push local 1
push constant 1
add
pop local 1
```

```
push local 3
push local 1
add
push local 1
push constant 1
add
```

```
push sum
push a
add
pop sum
push a
push 1
add
pop a
```

local :
a b sum
0 1 2

let sum = sum + a
push local 2 (sum)
push local 0 (a)
add
pop local 2 (sum)

let a = a+1
push local 0 (a)
push constant 1
add
pop local 0 (a)

Question 5

1 / 1 pts

Consider the following **Jack** class:

```
class bob
{
    function int foo(int a)
    {
            var int x;
            let x = a + x ;
            return x ;
    }
}
```

What sequence of virtual machine commands is used to initialise the local variable x?

- (●) `function bob.foo 1`
- ( )
```
push constant 0
pop  x
```
- ( )
```
push constant 0
pop local 0
```
- ( ) `function foo 1`
- ( ) The programmer has forgotten to initialise x, no code is generated.