Quiz - Lecture 10

Due 28 Aug at 23:59 **Points** 5 **Questions** 5 **Available** 27 Aug at 9:10 - 28 Aug at 23:59 1 day
Time limit None **Allowed attempts** Unlimited

Take the quiz again

Attempt history

	Attempt	Time	Score
KEPT	Attempt 2	1 minute	5 out of 5
LATEST	Attempt 2	1 minute	5 out of 5
	Attempt 1	less than 1 minute	0 out of 5

Score for this attempt: **5** out of 5
Submitted 27 Aug at 14:46
This attempt took 1 minute.

Question 1

1 / 1 pts

Briefly describe what the following Hack assembler code does

```
@KBD
D=M;
@48
```

```
D=D-A;
@num
M=D
(END)
@END
0;JMP
```

- Nothing, there is no way that this code is able to read the keyboard unless the user is able to press the number '48' which doesn't exist on the keyboard. If it could work it would put the value -48 in "num".
- Nothing, there is a syntax error and the code doesn't assemble.

Correct!

- [✓ It reads the keyboard code at the time of the first D=M command and subtracts 48 from it and puts the result in "num"]
- It reads the keboard register at the time of the @kbd command and exits if the scan code is more than 48.

Question 2

1 / 1 pts

What does the following Hack assembler code do?

```
(LOOP)
@KBD
D=M
@48
D=D-A
@num
M=D
@LOOP
D;JLT
@10
D=D-A
@LOOP
D;JGE
(END)
@END
0;JMP
```

- It loops, putting the scancode into "num", until the key pressed is between scancode 48 and 57.
- It loops until the user presses the space bar and then puts the scan code for the space bar minus 48 into num.
- It doesn't work because it goes into an infinite loop at the end of the code.
- It loops, putting the scan code into "num", until the keyboard scan code is less than 48 (which translates to the char '0').

Correct!

- [✓ It loops, putting the scan code minus 48 into "num", until the key pressed is between scancode 48 and 57.]

Question 3

1 / 1 pts

What does the following Hack assembler code always do to the current value in register D?

```
D=!D
D=D+1
```

○ Sets D to be 0

Correct!

● Sets D to be -D

○ Sets D to be 1 - D

○ Sets D to be 1

This flips all the bits in D and then adds 1. This is how you negate a number in 2s complement.

Question 4

1 / 1 pts

What does the following Hack assembler code do?

```
@pix
M=1
D=M
M=M+D
M=M+1
D=M
M=M+D
M=M+1
D=M
@SCREEN
M=D
(END)
@END
0;JMP
```

- ○ It draws the following pixels black, pixel 0, pixel 16 and pixel 32.
- ○ It draws one pixel black.
- ○ It draws no pixels black because of overflow.
- Correct! [✓ It draws the topmost leftmost three pixels on the screen black.]

Question 5

1 / 1 pts

What does the following Hack assembler code do?

```
@15
D=A
```

```
@count
M=D
@pix
M=1
(LOOP)
@pix
D=M
M=M+D
M=M+1
@count
M=M-1;
D=M
@LOOP
D;JGT
@pix
D=M
@SCREEN
M=D
(END)
@END
0;JMP
```

○ Nothing, it looks like it tries to draw to the screen memory but because it gets the offsets wrong it fails to write to the screen but instead writes to locations before the screen.

Correct!

● It draws 16 black pixels in a row in the top left of the screen

○ It draws 14 black pixels in the top left of the screen.

○ It draws nothing to the screen because of overflow in the pix variable.

Quiz score: **5** out of 5