**Workshop 06**

[**Pracmarker Link**](https://cs.adelaide.edu.au/services/pracmarker/?post)

**Question 1**

Look at [thisfile](https://forums.cs.adelaide.edu.au/forums/pluginfile.php/58008/mod_page/content/5/workshop6_asm-3.pdf) giving the mapping of an assembler language program to its equivalent machine code on the Hack platform. For each of the highlighted "*(this line should be erased)*" lines say why the corresponding line in Prog.asm does not appear in the binary code: Prog.hack.

The lines for **(LOOP)**and **(END)** in Prog.asm do not appear in Prog.hack but information corresponding to these symbols does appear somewhere in Prog.hack. Highlight where in the Prog.hack information derived from **(LOOP)** and **(END)** occurs. Briefly describe what this information refers to.

**Question 2**

Briefly describe what a *virtual register* is in the context of  Hack machine and its assembler. Does the hack machine have an awareness of virtual registers? If so explain why and if not explain why not (you get most marks for your explanation). Does the Hack assembler have an awareness of virtual registers? Again, explain your answer.

**Question 3**

Draw the symbol table for the Prog.asm file in[this pdf.](https://forums.cs.adelaide.edu.au/forums/pluginfile.php/58008/mod_page/content/5/workshop6_asm-3.pdf) Put a tick on the entries that are populated in the second path.

**Question 4**

Hand assemble the following program:

@1023

D=A

@SCREEN

M=D

(END)

@END

0;JMP

To do this you will need to look at the figures on pages 109 and 110 in [chapter 6](http://www.nand2tetris.org/chapters/chapter%2006.pdf) of the nand2tetris book.

**Additional Questions**

**Question 5**

Hand assemble the following program on the CPU emulator:

@X

D=M

@Y

M=D+M

(END)

@END

0;JMP

Note when you translate to a .hack file you should write the binary code in text form (so you can read it). The emulator can read in these files.

**Question 6**

Split the following C instructions into their dest, comp and jmp parts.

D=M

AD=M

D=A+1;JGE

0=D;JMP

0;JMP

**Question 7**

Briefly describe the code you would use to map a destination string into a binary string when writing a Hack assembler.