

In-semester Test 2

- The test will be **1 hour** in duration
- No Internet, handbook or access to files
- The test will consist of **four** questions
- Do not change the names of any files.

Before the test

- Download the test files from Minerva
- Extract the .zip
- You will have a directory that contains all the required files.

During the test

- The only files you need to modify are **q1.asm, q2.asm, q3.asm, q4.asm**

After the test

- Create a **.zip** of the **directory** (right-click on the directory, click ‘Send to’ and select ‘Compressed (zipped) folder’).
- Upload to the submission area on Minerva.

Check your submission before leaving the examination venue. Once you have left the examination venue, there is no opportunity to re-submit.

Question 1

Consult the **q1.asm** file. In there, you will find the API for the code that you need to write.

[5 marks]

Question 2

Consult the **q2.asm** file. In there, you will find the API for the code that you need to write.

[10 marks]

Question 3

Consult the **q3.asm** file. In there, you will find the API for the code that you need to write.

[10 marks]

Question 4

Consult the **q4.hack** file. In there, you will find some lines of binary machine code.

Dis-assemble these binary instructions back into the Hack assembly language instructions and write them in the **q4.asm** file.

Place one instruction on each line in the file. Do not include any comments or spaces between characters i.e.

M=D

Rather than

M = D

[5 marks]

30 marks total

Assembly Language Reference

A-instruction format

0	v	v	v	v	v	v	v	v	v	v	v	v	v
----------	---	---	---	---	---	---	---	---	---	---	---	---	---

C-instruction format

I	I	I	a	c ₁	c ₂	c ₃	c ₄	c ₅	c ₆	d ₁	d ₂	d ₃	j ₁	j ₂	j ₃
---	---	---	---	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------	----------------

zx	nx	zy	ny	f	no	comp	mnemonic	alu
c ₁	c ₂	c ₃	c ₄	c ₅	c ₆	a=0	a=1	
1	0	1	0	1	0	0	0	0
1	1	1	1	1	1	1	1	1
1	1	1	0	1	0	-1	-1	-1
0	0	1	1	0	0	D		x
1	1	0	0	0	0	A	M	y
0	0	1	1	0	1	!D		!x
1	1	0	0	0	1	!A	!M	!y
0	0	1	1	1	1	-D		-x
1	1	0	0	1	1	-A	-M	-y
0	1	1	1	1	1	D+1		x+1
1	1	0	1	1	1	A+1	M+1	y+1
0	0	1	1	1	0	D-1		x-1
1	1	0	0	1	0	A-1	M-1	y-1
0	0	0	0	1	0	D+A	D+M	x+y
0	1	0	0	1	1	D-A	D-M	x-y
0	0	0	1	1	1	A-D	M-D	y-x
0	0	0	0	0	0	D&A	D&M	x&y
0	1	0	1	0	1	D A	D M	x y

d ₁	d ₂	d ₃	Mnemonic	Destination
0	0	0	null	Not stored anywhere
0	0	1	M	Memory[A]
0	1	0	D	D register
0	1	1	MD	Memory[A] and D
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

j ₁ out < 0	j ₂ out = 0	j ₃ out > 0	Mnemonic	Destination
0	0	0	null	No jump
0	0	1	JGT	If out > 0 then jump
0	1	0	JEQ	If out = 0 then jump
0	1	1	JGE	If out ≥ 0 then jump
1	0	0	JLT	If out < 0 then jump
1	0	1	JNE	If out ≠ 0 then jump
1	1	0	JLE	If out ≤ 0 then jump
1	1	1	JMP	Jump