java -jar Mars.jar

**TPS (Think-Pair-Share) activity 1 Paired with the classmate sitting next to you**

**and do the following tasks (25 minutes):**

1. Before we use a new tool, we will need to find out how to use it. MARS DOES

NOT mean that MIPS is an alien language (well, sort of). Work with your

partner and find out what MARS stands for.

**MIPS Assembler and Runtime Simulator**

2. Since you have found out what MARS stands for, you probably have found out

the webpage of MARS as well. Visit the download page and download MARS in

your computer. To run MARS, just double click the downloaded jar file. You will

need Java to run it. (MARS is pre-installed in the lab computers, so no need to

download it if you are using a lab computer.)

**Done installed**

3. From the Tutorial materials page (you can find the link to it from the home

page), save both tutorial materials (MARS feature map and MARS tutorial)

as well as Fibonacci.asm in your Lab\_5 folder.

4. Follow Part 1 (Basic MARS Use) of the tutorial using Fibonacci.asm and

discuss the following questions:

a. How do you load an assembly file?

**File Open and it loads the file**

b. How do you assemble (compile) the program?

**You go to run and click assemble or F3**

c. How do you run the assembled program?

**Yo use the green play button, or F5**

d. Can you run a program before assembling it?

**No**

e. If you want to run the assembled program line by line, how to do it?

**You have use the play button with a 1**

f. How do you run the program again after it has finished running?

**You have to assemble again and run it. Or you can use the rewind button.**

TPS (Think-Pair-Share) activity 2 Paired with the same classmate and answer

the following questions (25 minutes):

1. What do the .data, .word, .text directives mean (i.e., what do you put in each

section)?

They are assembler directives.

With .data ( consecutive memory allocation )The following data items should be stored in the data segment. If the optional argument *addr* is present, the items are stored beginning at address *addr*.

With .word ( defines the size of the member )we Store the *n* 32-bit quantities in successive memory words. SPIM does not distinguish various parts of the data segment (.data, .rdataand .sdata).

With .text, ( anything after the the text the code starts)The next items are put in the user text segment. In SPIM, these items may only be instructions or words (see the .word directive below). If the optional argument *addr* is present, the items are stored beginning at address *addr*.

2. What does line 10 do?

fib: beq $t3, $0, finish # if branching equal (a == 0){go to finish}

it is a comparison, comparing $t3 to Zero, if it is equal then it goes to finish, so basically a While loop but for MIPS

3. What does line 15 do?

**j fib** it is jumping to the beginning of fib

4. How do you set a breakpoint in MARS? Set breakpoint on line 15 and list the

steps of achieving this.

You have to assemble and then mark the checkbox in the **Bkpt** in the **text segment** table

5. After your program stops because of a breakpoint, how do you continue to

execute your code? How do you step through your code?

You run it line by line with the Green play button with the 1

6. How can you find out the content of a register? How do you modify the value of

a register manually while running the program?

The content of your register is displayed in the register section in the rightmost part of the screen. You have to click on the value section and enter a hexadecimal value.

7. At what address is n stored in memory? Calculate the 13th fib number by

modifying this memory location.

It is the first address of of our data segment first two digits in this case.

8. Line 19 and 21 use the syscall instruction. What is it and how do you use it?

Syscall is used to print in this case. You load a value in the $v0 putting AND IT DOES SOMETHING**. 1 prints and 10 terminates the program**