

# BBM234 - MIPS Project

Assigned on: 30/03/2020 Due date: 11/04/2020 (23:59) – via the submit system.

## Introduction

In this project, you will learn how to write and simulate MIPS code.

You can either use **QtSpim** or **MARS** MIPS simulator for this project:

Download QtSpim from <http://spimsimulator.sourceforge.net/> and install it to your computer.

There are several tutorials for QtSpim, some of which are listed below. If you can find better ones, please let us know for future references.

<https://www.youtube.com/watch?v=r8WcV7AiLXs>  
<https://www.lri.fr/~de/QtSpim-Tutorial.pdf>

The second simulator you can use is MARS, which can be downloaded from the following link:

<http://courses.missouristate.edu/kenvollmar/mars/>

## How To Write and Simulate Your Code

- You should use a text editor such as notepad or wordpad and write your MIPS program first.
- Save your program with an extension .asm
- Open QtSpim and load your program.
- Run your program.
- Your results will be either in registers or in memory based on your program output.

## What to Turn In

In this assignment, you will write and simulate **two different programs**. You should write comments to explain the purpose of the instructions in your program. Please turn in each of the following items through the **submit system** (only .zip files are supported):

- array.asm
- function.asm
- report.pdf

Your report should include the explanation of the problem solutions and your **code with comments**. It should also include the screenshots of your tests and results clearly explained.

We encourage you to use the [LaTeX report template](#).

## Grading

Arrays Problem:	40 points
Function Call Problem:	50 points
Report:	10 points

## Problem 0: Arrays Using for Loops

Write a MIPS code for the following C code fragment. You are **not allowed** to use multiplication and division instructions (mult, mul, div) for this problem.

```
int Array[5];
for (int i = 0; i < 5; i++){
    if(Array[i]%2==0)
        Array[i] = Array[i]/2;
    else
        Array[i] = (Array[i]*3)+1;
}
```

Your array size will be 5. Test your code for the test cases given below.

Test 1: A={1,2,3,4,5} Test 2: A={2,4,10,0,50} Test 3: A={1,3,5,9,13}

For each test, save the screenshot of the memory and registers before and after running the code.

An array  $A$  can be defined as follows:

```
.data
A: .word 1, 2, 3, 4, 5    #Do not forget a space after a comma!
.text
main: la $t1, A
```

After **la** instruction, the address of the first element of array A will be stored in register **t1**. Then, you can write your code afterwards. You can also define the array size in the data segment as well.

## Problem 1: Function Calls

Write a MIPS code for the following C code fragment.

```
int main(){
    int a, b, result;
    if(a == b)
        result = a*b;
    else
        result = assess(a, b);
    return result;
}
int assess(int a, int b){
    if(b<a)
        return upgrade(a, b);
    else
        return demote(a, b);
}
int upgrade(int a, int b)
{return 4*(a+b);}
int demote(int a, int b)
{return 4*(b-a);}
```

Test your programs for the following input values:

Test 1: a=8, b=8 Test 2: a=3, b=5 Test 3: a=5, b=3

For each test, save the screenshot of the registers before and after running the code (**\$s0=a, \$s1=b, \$s3=result**). You should clearly show how you use stack in your code.

