Project Objective. The assigned project consists of four programs which you are required to write and execute using MARs Simulator. The programs worth 70 points of total project grade where program 2, program 3 and program 4 are worth 20 points each, while program 1 worths only 10 points. You are also required to submit a project report which worths 30 of total project grade.

Program 1 (10 points) Write a MIPS assembly code that sends the user a message *"Hello, may I have your name, please?"*. The MIPS code takes a name from the keyboard (the user enters a name). Then, MIPS code sends *"Welcome, name"* message. Check the output sample below as an example.

Output Sample
MIPS: "Hello, may I have your name, please?"
Users: "Rasha"
MIPS: "Welcome, Rasha"

Program 2 (20 points) Write a MIPS assembly code that reads four <u>positive integer</u> **a, b,c,** and **d** as input parameters. The program shall calculate the following equation:

$$F = (a+b) - (c+d) + (b+3)$$

The program should output **F** as an integer, using **syscall.** The code shall execute in MARS to prompt the user to enter four positive integer numbers where each is separated by the Enter Key. Below is a sample of the input and output data.

Sample of data

a	b	c	d	Expected F
2	3	5	2	4
10	20	30	10	13
5	5	25	25	-32

Program3 (20 points) Below is a code written in C language. You are required to translate that code

into MIPS assembly code. Assume i is in $s0, j is in $s1, and k is in $s2. Below is a sample of the

input and output data.

```
int j=3, k = 5;
printf ("Program starts\n");
for (i=0; i<5; i++){
        f = i+j-k;
        printf("f = %d\n",f);
}
printf ("Program ends\n");
```

Sample of data

i	f
0	-2
1	-1
2	0
3	1
4	2

Program4 (20 points) Convert the following code written in C into an assembly code. Then, execute

your code in MARs Simulator. You may use any static and temporary register in your code.

```
printf("Loop starts\n");
for (i=10; i>0; i-=2) array[i] = i+2;
printf("Loop ends\n");
```

Project Report (30%)

The project report will be graded out of 100, and the points will be distributed as following:

A. **Professional preparation (10 points):**

You are required to submit a <u>typed document</u> with text of the paragraphs in **TimesNew Roman 11 pt font**, clear and grammatically well-formed explanations, page numbering and document heading numbering (1.0, 2.0, 3.0, etc to identify the required sections listed below).

B. **Report Content (90 points):**

You are required to provide the answers for the following numbered section headings:

1.0 **Program Input/Output (20 points):** Describe (or list) the inputs and outputs of each program.

2.0 **Program Design (20 points):** Describe briefly how each code (program 1, program 2, program 3 and program 4) operates.

3.0 **Symbol Table (20 points):** Create a 2-column Table describing all Registers used and their specific Purpose in the code, where each register is listed on a separate row and identified by register name $t0, $s0, etc., aswell as any Labels used and their purpose on separate rows. You may takethe table below as sample.

Registers	Purpose & Labels
$a0	Argument of syscall to print string
:	:

4.0 **Learning Coverage (10 points):** Provide a list of at least 5 technical topicslearned from this project.

5.0 **Test Results (20 points):** Provide one screen shot of proper MIPS code executions in MARS for your Test Plan inputs. To illustrate, you are required to execute each program (1, 2, 3, 4) for one time. Take a screen shot of the execution of each program and include it in your report. Your screen shot must show the text segment, data segment and registers.

Project Submission

1. Create a **folder** and name it as **p1_username**, where *username* is youruniversity username.

2. In MARS, save the code of program 1 as **c1.asm**, save the code ofprogram 2 as **c2.asm**, program 3 as **c3.asm** and program 4 as **c4.asm**.

3. Save your **report** as a PDF file and name it as **p1_report.pdf**.

4. Place the three asm files and the report document inside p1_usernamefolder.

5. Compress (or zip) the folder.

6. Then, upload the compressed folder on Blackboard.