Integer Array

Requirements

1. Mac, Windows or Linux PC
2. Java standard edition (SE) version 9 or higher.
3. MARS Java program

Introduction

In this lab, you will use *MARS* (**M**IPS **A**ssembler and **R**untime **S**imulator) to write and run a program that reads in a 4 x 4 matrix of numbers stored as an Integer array. You will take the program written for Lab 2 and expand it for this lab. The purpose of this lab is to learn to use "procedures."

As before, if you have any problems running MARS refer to the instructions in Lab 1.

There are two different ways to convert the ASCII into integers; arithmetic and logic.

- If your student ID ends in an even number, then you use arithmetic.
- If your student ID ends in an odd number, then you use logic.

  ***Note**: this is the opposite from Lab 2.*

Exercise 1

1. Make a new version of lab 2 called "Lab 3"
2. Convert your vector read routine in Lab 2 to a procedure that reads in a vector of length 4.
3. Call the vector read routine 4 times to read in the matrix "M", use the "**`jal`**" instruction.
4. Between each call, store the results into the 16-entry integer array "M".

   Note: This will be storing the matrix into the array in *row major* order.

Same as Lab2, use only whole numbers (0 – 9). The input on the screen should look like a matrix just as below:

```
Row 1: 3 4 2 9
Row 2: 4 6 0 1
Row 3: 5 5 3 3
Row 4: 1 2 3 4
```

(with the matrix values input by the user on the console)

After reading all 4 rows of the matrix into the array:

- Write a loop routine that sums all the elements of the array. The loop routine should have only one branch at the end of the loop. (Loop practice)
- Use an "if then else" construct to print "0" or "1" if the sum is even or odd. (If then else practice)

```
Sum: 1
```

Upload your finished program to Canvas.