Assignment 5

We are going to apply our knowledge of MIPS procedures and arrays on this assignment. Write a program that uses procedures to compute values based on an array. The MIPS program should include a procedure for each of the following:

- min – return the minimum value in the array
- max – return the maximum value in the array
- sum – return the sum of the values in the array
- avg – return the average value of the array

Make sure you call each procedure separately and DO NOT combine the logic into one procedure or the mainline code.

- For min, max, and sum, pass the memory location of A and the size of A as arguments. Use the values passed and do not depend on any other values from the calling program. Return the calculated value in $v0.
- For avg, you may pass the sum as calculate by the sum procedure in $a0 and the size of the array in $a1.
- After each procedure call, save the value returned from each procedure to the appropriate memory location and print the results to the terminal.

Use the following as the start of your data segment.

```
.data
A:          .word 11, 250, 20, 70, 60, 140,150, 80, 90,100, 1, 30, 40,
120,130, 5
size:       .word 16 # length of array A
min:        .word  0
max:        .word  0
sum:        .word  0
average:    .word  0
largestInt: .word  2147483647 # You may want to use this for min procedure
```

Example output:

```
The minimum is 1
The maximum is 250
The sum is 1297
The average is 81
```

Challenge/Bonus question (Worth 10 extra points):

- Create another array called B in the data section using the .space directive. Make the array the same length as A.
- Write a procedure called **reverseArray**, which accepts the memory location of array 1 (A), the memory location of array 2(B), and the length of the arrays. This procedure will fill array 2 in reverse order of array 1. If you call **reverseArray** with A as array 1 and B as array 2, then 5

should be the first element in B and 11 should be the last. This procedure does not have any return values.

- From the mainline, call **reverseArray** and **AFTER** calling the procedure, print array B. (You may use a procedure or print it from the mainline code.

The following are required for all assignments and are included in the rubric for grading:

- You need to name your file as "LastName-Name-Assign.asm" (Example: Talley-Michelle-Assign5.asm)
- Your program will need to have the exact output unless otherwise stated. ***Make sure to use spaces and newlines as required.***
- You need pseudo code to describe your planned implementation.
- Your source needs to have comments that explain your implementation.
- Your procedures need to have comments
- You need to make sure you exit your program and avoid calling procedures unnecessarily.
- You need to include the following set of comments at the top of your source code for all assignments.
  #Your Name
  #Assignment # (Example: Assignment #5)
- You need to submit your source code on blackboard.
- Please submit your files in a zip file named LastName-FirstName-Assign5.zip) and make sure you include any files that are used as includes in the zip file (Example: utils.asm).