CSCI-21 Lab 5, due 4/4/22

Please work in groups of two for this.

Exercise 1: Write a MIPS program that declares a string (an `.asciiz`) containing a fairly short string, such as "The quick brown fox jumps over a lazy dog".

```
      .data
str:  .asciiz    "The quick brown fox jumps over a lazy dog\n"
```

The program will reverse the string in place; that is, without copying it to another place in RAM. The basic algorithm is as follows:

```
Create 2 pointers, here called s and t (use registers)
Set pointer s to the start of the string.
Set pointer t to the end of the string, minus the newline (scan for the
newline ('\n'), then back up one place).
while( s < t )
   swap *s and *t
   increment s
   decrement t
end loop
```

A swap routine in a language like C needs a temporary location `c`, e.g.

```
 c = *t
*t = *s
*s = c
```

In C, `*s` means the thing pointed to by `s`. In assembler the registers you `lb` the bytes into can serve as the temporary location(s).

Exercise 2: Write a MIPS program that processes an array by applying an averaging filter to it. An averaging filter works like this: create a new array where each element at index J is the average of the three elements from the old array at indexes J-1, J, and J+1

```
      .data
size: .word 12
array:
      .word 50,53,52,49,48,51,99,45,53,47,47,50
result:
      .word 0,0,0,0,0,0,0,0,0,0,0,0
```

In the above, the second element of the result is the average of 50, 53, and 52. The first and last elements of the new array are copies of the corresponding first and last elements of the old array. Use all integer arithmetic.

Remember to set the load and branch delays ON in the QtSpim simulator. Additional exercise: try to calculate the number of clock cycles each of your programs take to run if all instructions (including `nop` for delay slots) except for multiply and divide take 1 clock cycle; multiply takes 32 clock cycles; divide takes 38 clock cycles. You can put these values (and show how you arrived at them) into the body of the email you send me for this lab.