**CS 4400 - Problem Set 1**

**Rob Johansen**

**u0531837**



1. I subscribed using this email address: [rob.johansen@gmail.com](mailto:rob.johansen@gmail.com)
2. Here is my code for the string\_length() function:

**#include** <stdio.h>

**int** **string\_length**(**char** \*string)

{

**int** length = 0;

**for** (; \*string != '\0'; string++) {

length++;

}

**return** length;

}

**int** **main**()

{

**printf**("Length of 'Hello, world!' is %d\n", string\_length("Hello, world!"));

**printf**("Length '5' is %d\n", string\_length("5"));

**printf**("Length of empty string is %d\n", string\_length(""));

**printf**("Length of 'CS 4400 Student' is %d\n", string\_length("CS 4400 Student"));

**printf**("Length of 'Go Utes!' is %d\n", string\_length("Go Utes!"));

**return** 0;

}

1. Problem 2.61. The following expressions evaluate to 1 when their condition is true, and to 0 when false.  
   1. Any bit of x equals 1:  
        
       x && 1
   2. Any bit of x equals 0:  
        
       (x ^ INT\_MAX) && 1
   3. Any bit in the least significant byte of x equals 1:  
        
       (x & 0xff) && 1
   4. Any bit in the most significant byte of x equals 0:  
        
       (((x >> ((**sizeof**(**int**) - 1) << 3)) & 0xff) ^ 0xff) && 1
2. Problem 2.62. Here is my implementation of the int\_shifts\_are\_arithmetic() function, which yields 1 when run on a machine that uses arithmetic right shifts for ints, and 0 otherwise:

**int** **int\_shifts\_are\_arithmetic**()

{

**int** test = -1; // On a 2's complement machine, test will be all 1s

test >>= 1; // Shift right one bit

**return** test == -1; // If the shift was arithmetic, test will be unchanged

}