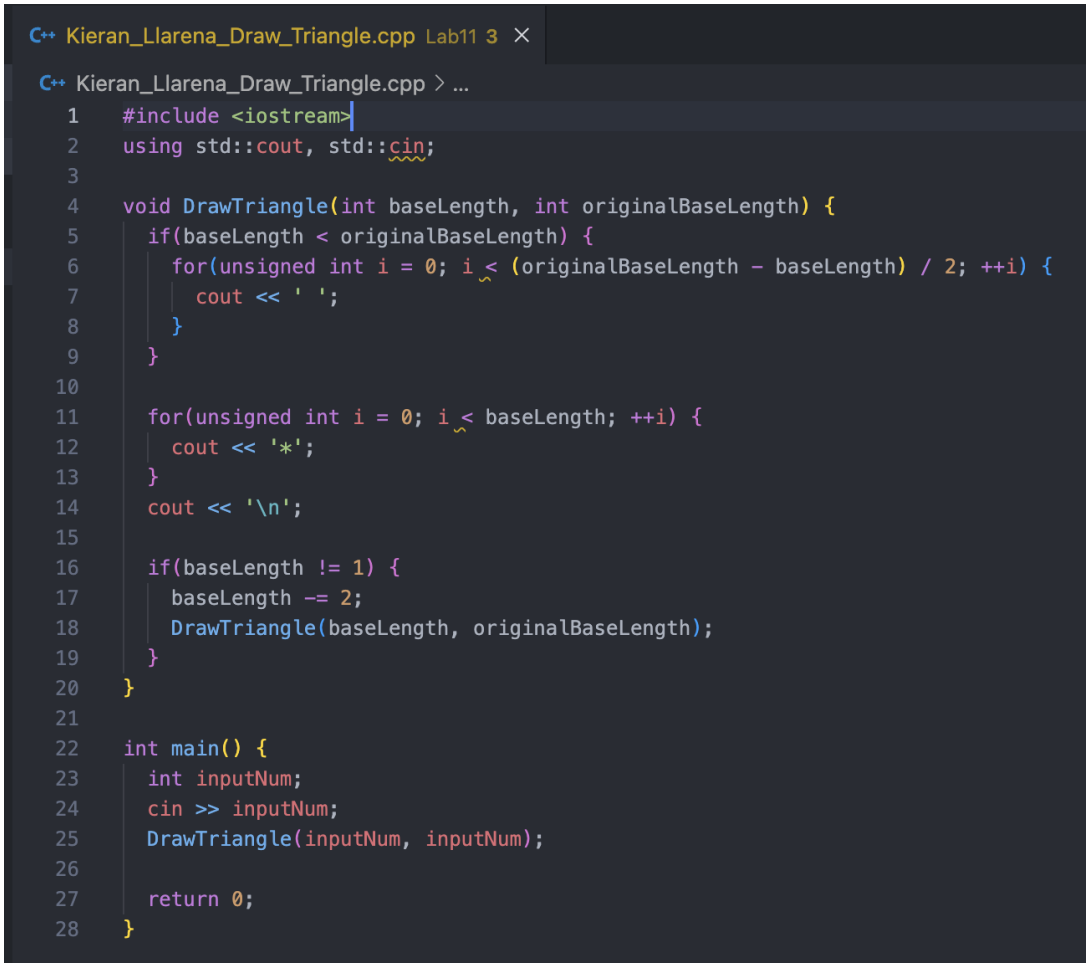
Assignment 1

Lab Explanation

The requirements for this lab were to create a program that would draw a triangle using a recursive function.

Code



I used for loops in my program to create the triangle. Additionally, I would preserve the original base length value that was passed on the first function call. I would compare the inputted base length and the original base length to center the output accordingly. I was going to use setfill() and setw() from iomanip initially, but I kept running into problems.

Test 1

```
→  output ./"Kieran_Llarena_Draw_Triangle"
3
***
 *
```

Test 2

```
→  output ./"Kieran_Llarena_Draw_Triangle"
19
*******************
 *****************
  ***************
   *************
    ***********
     *********
      *******
       *****
        ***
         *
```

Test 3

```
→  output ./"Kieran_Llarena_Draw_Triangle"
1
*
```

Test 4

```
→  output ./"Kieran_Llarena_Draw_Triangle"
9
*********
 *******
  *****
   ***
    *
```

Test 5

```
→  output ./"Kieran_Llarena_Draw_Triangle"
15
***************
 *************
  ***********
   *********
    *******
     *****
      ***
       *
```

Assignment 2

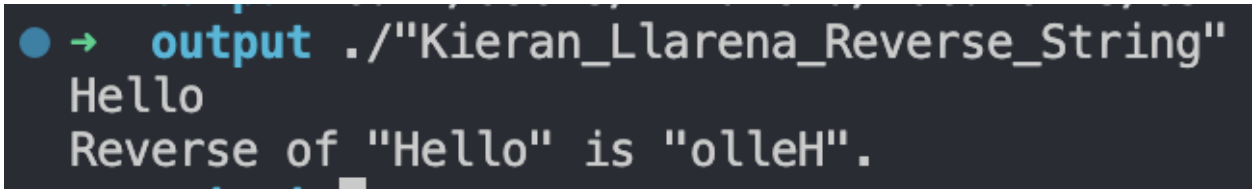
Lab Explanation

The requirements for this lab were to create a program that would get a user inputted string and output said string in reverse using a recursive function.

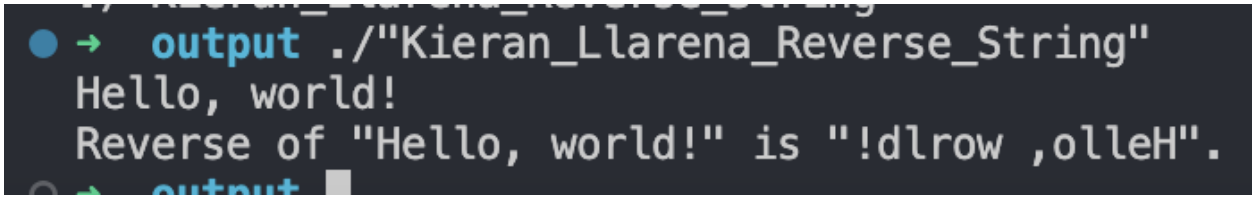
Code

My code is composed of two parts. The recursive function and an auxiliary function. The recursive function just recursively iterates through the inputted string starting from the end and appends the associated character at that index to a vector. This vector is declared in the auxiliary function, which converts this vector of characters into a string and returns it to be outputted in main

Test 1


→ output ./"Kieran_Llarena_Reverse_String"
Hello
Reverse of "Hello" is "olleH".

Test 2


→ output ./"Kieran_Llarena_Reverse_String"
Hello, world!
Reverse of "Hello, world!" is "!dlrow ,olleH".