Ryan Hoffman
CS202
Project 11 Documentation
5/3/2019
Compile Instructions: make -> ./proj11

The purpose of this project was to introduce recursion by having us create two functions that make use of recursion. We also got to familiarize ourselves with vectors. Other things included with the project that we've learned in the past are templates, dynamic data structures, as well as manipulating dynamic memory, classes, pointers and iostream.

I made my design a little different from the recommended sorting algorithm for the vector_resort function. For the vector_resort function, I passed in the vector and the size of the vector into the function. My base case for the recursive function was when the size was equal to 1, the function would stop/return nothing. My sorting method was to go through each vector value using a for loop and compare it to the vector value ahead of it. Since I was sorting from least to greatest, if the vector value in front was greater, I swapped the values of the vectors using a temporary variable that help the value of the vector value. After going through the for loop, I recalled the function in order to keep the recursion going. For my vector_research function, I used the recommended algorithm from Wikipedia to design the function. The function returns the int value of the index that the value that we're searching for is at. The parameters are the vector, the value being searched for, and the left and right int values. In this case, the left value was 0 and the right value was the size of the vector minus 1. The middle value was equal to the left and right values added together and then divided by two. The base case for this function was if the left value was greater than the right value, the function returns -1. The recursive case has two parts. If the middle value of the vector was less than the value being searched for, the function re-calls itself except the left value is set to the mid value plus one. For the second part, if the mid value was greater than the value being searched for, the function re-calls itself except the right value was set to mid minus one. At the end the function returns the mid value which is the index that the value was found at, unless the value couldn't be found in which the function returns negative one.

In the proj11.cpp file, a vector of int values was created and the values were read in from a text file of int values. I created another vector of int values that was a copy of the previous vector created. I then called the sort function to sort the vector from least to greatest. I then called the search value with a random number in the text file and it returns the index of that value. Then I used the for loop that prints out the vector to show that it was sorted from least to greatest.

The only problem I had for this project was that when I was trying to swap the values, I created another vector that would hold the temporary value from the vector in the parameters of the function, when really I only needed a temporary variable of type T (or in this case int) that help whatever value type that the vector held.