Miko Farin (704789591)

Project 3

1. Some notable obstacles I faced and overcame was figuring out how to set parameters and making sure that it is configured to the specification. Also another type of obstacle I faced was to use case and switch statements so I didn’t have to write most of the alphabet and numbers to fit the specifications. Another was to change string into integers that you are able to manipulate as numbers. This allowed for comparison for strings and for the functions you needed to write. Lastly, one obstacle I faced and I could not overcome was have switch statements and nested loops. This overwhelmed my code and I could not fit the specification.

2. The design of my code was to check the code or string if it was true/correct or if it was false/incorrect. This was set by using the bool i**sValidEbayListingString(string auctionstring)**, however, I was not able to finish this portion of my code due to time constraints and I was not able to figure out the code I needed to fit the specification. The next step of my code, I wrote a function named **bool listingSold(string auctionstring).** This function was to add up all the Bids and compare it to the listing amount. If the amount of dollars in bids were above the listing price we set the bool to true indication it was sold off. Contrary, if the listing amount was more that the or equal to the amount in dollars to the bids, it was considered false, indication it was not sold off. Then after we add we use the function of **int howMuch(string auctionstring),** to show the amount of the bids all together. We were to convert the string into an integer amount. Lastly, we used the last funtion named **int watchers(string auctionstring),** this function was to give the amount of watcher in the and after the bid.You needed to add all of W’s then subtract the amount of U’s within the string to get the amount of watcher.

3. Test cases:

* assert(isValidEbayListingString("") == false);

This test the function is to test if the string if there is a listing or anything writing.

* assert(isValidEbayListingString(" ") == false);

This was to test if the string was an empty string, if it was we concluded it was false.

* assert(isValidEbayListingString("a") == false);

To test if the beginning letter had to be a L or l character. This is false.

* assert(isValidEbayListingString("z") == false);

To test if the beginning letter had to be a L or l character. This is false.

* assert(isValidEbayListingString("0") == false);

To test if the beginning letter had to be a L or l character. This is false.

* assert(isValidEbayListingString("l50") == true);

This was to test if the string was valid if it was just the listing string.

* assert(isValidEbayListingString("L50") == true);

This was to test if the string was valid if it was just the listing string. Capital letters are accepte.

* assert(isValidEbayListingString("L00050") == false);

The string is not valid if the letter L is followed by a zero character.

* assert(isValidEbayListingString("L-1") == false);

The trailing character after the L had to be a digit character.

* assert(isValidEbayListingString("Lambasdff ") == false);

The trailing character after the L had to be a digit character.

* assert(isValidEbayListingString("L50L50L50l50") == false);

To test if there that repeating L or listing price was invalid

* assert(isValidEbayListingString("L100WU") == true);

You are able to watch after a listing price or a bid.

* assert(isValidEbayListingString("L100W") == true);