Homework 3 Report

The *ShoppingCart* class in this implementation, a linked list, is very similar to the previous one. The only big difference being that this class is using a linked list to story items rather than an array. This makes the implementation more complex and tedious.

A difference in the functions is that in the previous class we had a *clear* and *destroy* function. This linked list class uses the *clearCart* function as both a destroy and clear use case.

The *checkItems* function in the array class is different and more efficient in the new implementation here. The new function is called *checkCase* and uses the difference between ASCII values between characters.

```
for (int i = 0; i < inCart.size(); i++)
{
    int difference = r_str[i] - c_str[i];

    if (difference == 32 || difference == -32  || difference == 0)
    {
        sameCount++;
    }
}
```

This is the main logic behind the function. If the difference is +32, then the item we want to remove, *r_str*, is the lowercase version of the item in the cart, *c_str*. If the difference is -32, then the item we want to remove is the uppercase version of the item in the cart. If the difference is 0, the cases are the same.

This implementation of the case checking function is much more efficient than the previous class as it is more versatile to use and simpler to understand.

The Big O of this class is O(n) as we need to go through the entire list if we want to print or remove an item. Adding an item is O(1), with the array implementation, it was O(n) in its worst case scenario.

An advantage of this class is that it builds upon and is more efficient than the array implementation with being case-insensitive, unlike the array class, and it keeps most of the same functions. A disadvantage is that it is still O(n) just like the previous array *ShoppingCart* class.