Nikhil Mashettiwar

Alekhya Dulur

Project 2: Sorted List Description

Our main allows the user to enter three different options: int, string, and double. Based on the user input the main will execute and create a sorted list of integers, strings or doubles.

For example:

The main takes in the integers listed in the numbers array. A Sorted List is created and each item in the array is created into a node and inserted into the list. The insertion function uses the compare function given to determine what place the node will be inserted. Duplicates will not be inserted in the list. The program will take O(n) time to run because each item that has to be inserted has to compare its value to each item already in the list. Our list is created to be in descending order so the item to be inserted will be compared until a node which is smaller is found.

The iterator is created by passing the sorted list to be iterated as a parameter. The iterator stores the head value of the list as the current value. When SLNextItem is called, the function should return current and then change the current value of the iterator to the node after current. Getting the next node should take O(1) time because it is only getting one value which is already a member of the node. Inserting and deleting cannot happen during the iteration. The Iterator is used to print out each value of the list after it has been sorted.

After printing the list, the iterator has to be destroyed. The sorted list is then destroyed by freeing the memory allocated for the data in each node as well as the memory allocated to the node struct itself. After the while loop conducts this freeing of space, the destroy function will also have to delete the space allocated for the sorted list struct.