Project 7 – Bubble Sort for Linked List

EGRE246 – Fall 2017

Assigned 11/22/2017 – Due 12/6/2017

Introduction:

So far we have programmed some data structure and algorithm concepts such as linked lists and state machines. This assignment focusses on the programming of an entry-level sorting method: Bubble sort. So far, we have seen how bubble sort can work in the context of an array, but this method can be applied to other linearly organized data structures. In this program, you write a bubble sort for the linked class you wrote in Project 6.

linked.h and linked.cpp

- The class maintains all functionality from the class written in program 6.
- Method called Bubble sort, returns void and has no parameters. Remember the steps
    1. Start at the beginning of the list
    2. Set a swap flag to false
    3. Iterate from the beginning of the list to the next to last item in the list
        i. If the data in the current item is greater than the data in the very next item, swap the values. Set the swap flag to true.
    4. If the swap flag is true, repeat steps 1-4, otherwise, the method is over.

main.cpp

In main.cpp you are going to implement 2 linked lists: 1) integer list, 2) Character list. Your program will also accept 2 command line arguments. The first argument is the number of elements in the list, the second argument is a seed for srand().

1. If argc < 3, throw a runtime_error with a message of

```
Too few arguments
Usage: main.exe NUMBER_OF_ITEMS SEED
```

2. Seed srand() with the integer value of the 3rd argument passed.
3. Create a list for integers and a list for characters.
4. Push the value of argv[1] number random integers (between 0-99 inclusively) into the integer list
5. Push the value of argv[1] numbers of characters (use the formula ‘a’ + rand()%26) into the character list
6. Display the integer list (unsorted)
7. Sort the integer list using bubble sort method
8. Display the integer list (sorted)
9. Display the character list (unsorted)
10. Sort the character list using bubble sort method
11. Display the character list (sorted)
12. In your catch block, accept a runtime_error object and display the .what() method

Submission:

Submit main.cpp, linked.h, linked.cpp, and makefile to blackboard.

Example Output: