CSCP 2031: Data Structures and Algorithms
Lab 08
Topics: Linked List

Please keep in mind the following points while coding. Violating any of these will result in credit deduction.

- Indent your code.
- Comment your code.
- There should be no memory leakage in your class. There should be no dangling pointers.
- Make functions, objects, variables as constant wherever possible.
- Plan your code carefully on a piece of paper before you implement it.
- Submit only .CPP and .H files in a single ZIP folder.
- The name of each CPP file should be the question number. E.g., Q1.cpp
- The name of the ZIP folder should be your complete registration number. E.g., L1F20BSSE0000
- void main() is not allowed. Use int main()
- You must work in multiple files. i.e separate .h and .cpp files
- You must not use system("pause")
- You must not use any built-in functions.
- Use meaningful variable names.
- You are required to follow the naming conventions as follow:
    - Variables: firstName; (no underscores allowed)
    - Functions: getName(); (no underscores allowed)
    - ClassName: BankAccount (no underscores allowed)

Task 1

Implement doubly circular linked list with the following functions:

1. Constructor
2. Insert at tail
3. Insert at head
4. Insert At Position
5. Delete a node (cater all the cases)
6. Display
7. Sort values placed in the list in ascending order
8. Sort values placed in the list in descending order
9. Count
10. Destructor – delete all nodes in linked list and empty the list

Your program should be menu based, where user selects to insert at head, tail, at a specific position, insert in sorted order, delete a node from head, from tail, from a specific position, sort values in ascending, in descending order, count and display the total values in the linked list. Your program should exit when user presses zero.

Task 2

Using Link created above do following task.

- Insert integer data in link list and create 10 nodes
- Search a number in link list and return position of that number.

Task 3

Write the following Functions

- reverseList() // reverse the contents of the list.
- GetTotal() // tells the sum of the elements in the list
- swapHeadAndTail() // swaps the value present at the head and tail
- display() //displays the list