

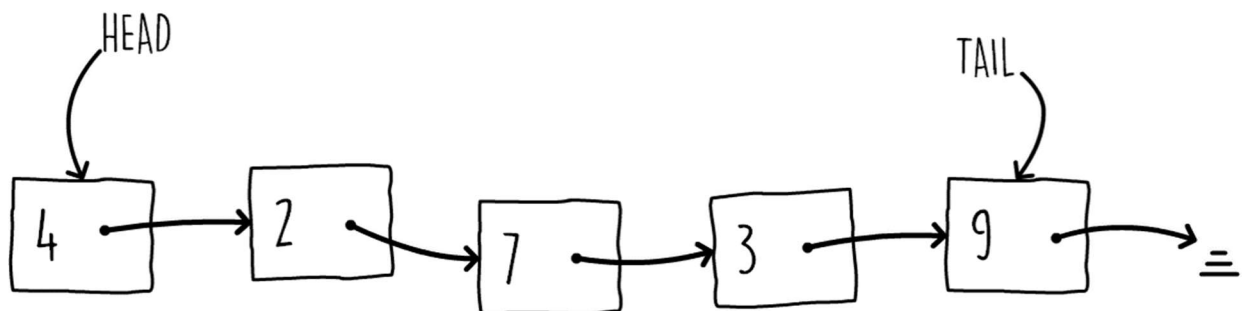
Data Structures

CS202

Assignment 1

Deadline: 21st September 2019
11:55 pm

In case of ambiguity, contact Anusheh Zohair Mustafeez
at 21100072@lums.edu.pk



Q1. Modify the LinkedList header and cpp files given to implement a doubly-linked linked list with the following functions:

- **Constructor:** Allocates memory for your LinkedList object
- **Copy Constructor:** Given a pointer to another LinkedList object, copies all elements from that object to your new list
- **Destructor:** Deallocates memory
- **Get_Head and Get_Tail:** Returns a pointer to the head or tail of the list respectively
- **Search_for_Element:** Given a node value input, returns the node that value corresponds to. If value not found in list, returns null pointer
- **Get_Min and Get_Max:** Returns a pointer to the node with the maximum or minimum value of the list respectively
- **Delete_Head, Delete_Tail and Delete_Element:** Deletes the head, tail or element whose value is given as input respectively
- **Insert_at_Head, Insert_at_Tail:** Inserts node with value given to the function at head or tail of the list respectively
- **Insert_After:** Takes two inputs: the value of node after which a new node is to be added and the value for the new node to be added. It then adds the new node in the correct position.
- **Get_Length:** Returns the number of nodes of the list
- **Reverse_List:** Reverses the direction of the LinkedList object such that the tail is now the head and vice versa with all intermediate pointers reversed (You are not allowed to instantiate another LinkedList to do so).

Feel free to add any helper functions you like.

Bonus Question: Modify the code to make a doubly linked circular linked list.

Q2. Import LinkedList.cpp and use the Linked List prototype you made in Q1 to make a program which computes results from polynomials where each polynomial is a linked list and each node represents a term in the polynomial. Your program should have functions which:

- Return the coefficient of a given exponent
- Compute and return roots of a quadratic polynomial
- Find and return the x and y coordinates of the vertex of a quadratic polynomial
- Modify a quadratic polynomial according to graph shifts in both across the x and y axes
- Given multiple polynomials, return the product and sum of those polynomials.

Q3. Modify the stack header and cpp files given to implement a stack data structure with the help of the LinkedList class you created in Q1. Make sure your stack object has the following basic implementations:

- **Constructor, Destructor and Copy Constructor**
- **Push:** Pushes item at the top of the stack
- **Pop:** Returns and deletes item at the top of the stack
- **Top:** Returns item at the top of the stack without deleting it
- **Get_Length:** Returns number of items in the stack
- **Is_Empty:** Returns true if there is no item in the stack otherwise returns false

Feel free to add any helper functions you like.

Q4. Modify the queue header and cpp files given to implement a queue data structure with the help of the LinkedList class you created in Q1. Make sure your queue object has the following basic implementations:

- **Constructor, Destructor and Copy Constructor**
- **Enqueue:** Adds item to the end of the queue
- **Dequeue:** Returns and deletes item at front of the queue
- **Get_Front:** Returns item at front of the queue without deleting it
- **Get_Length:** Returns number of items in the queue
- **Is_Empty:** Returns true if there is no item in the queue otherwise returns false

Feel free to add any helper functions you like.

Q5. Now that you understand how stacks and queues are made, create a program which uses two queues as made in Q4 to implement a stack.

Q6. Finally, write a program that reverses the contents of the file “text.txt” using the stack data structure you made in Q3. First make a function that reverses the contents character-wise and then make another function that does the same string-wise.

Submission Guidelines:

Submit a zip folder containing the following files on your lms Assignment 1 tab:

- LinkedList.h
- LinkedList.cpp
- Bonus.h
- Bonus.cpp
- Polynomials.cpp
- stack.h
- stack.cpp
- queue.h
- queue.cpp
- Two_Queue_Stack.cpp
- Reverse.cpp

Naming Convention: A1_**your roll number**.zip