

Data Structures & Algorithms (Course & Lab) – Fall 2015
(BS-SE-F14 Morning & Afternoon)

PRACTICE Assignment # 2

Submission Deadline: None!!

Instructions

- Although this is just a practice assignment, but still you need to work on it individually. Absolutely NO collaboration or discussion is allowed.
 - You are not required to submit this assignment, but I will assume in quizzes and exams that you have completed this assignment.
-

In Queue data structure elements are inserted at the back and removed from the front end of the queue. A **Double Ended Queue** (Deque) is a queue in which you can insert and remove elements **both at the front and at the back of the queue**. In this assignment, you are required to implement a class **Deque** (which will be used to store integers). The **Deque** class should have the following public member functions:

• Deque (int n) Constructor to create a Deque which can contain up to n elements.
• ~Deque () Destructor to destroy the Deque.
• bool isEmpty () returns true if the Deque is empty, and false otherwise.
• bool isFull () returns true if the Deque is full, and false otherwise.
• void display () to display the Deque from front to back.
• bool insertAtFront (int val) to insert val at the front of the queue.
• bool insertAtBack (int val) to insert val at the back of the queue.
• bool removeFromFront (int& val) to remove an integer from the front of the queue and store it in the variable val .
• bool removeFromBack (int& val) to remove an integer from the back of the queue and store it in the variable val .

The last four functions should have the **bool** return type. These functions should return true if the desired operation was performed successfully. Otherwise these functions should return false.

Note that:

- **The time complexity of last 4 member functions should be constant i.e. $O(1)$.**
- Your Deque class should refuse the insertion of a new element ONLY when there are no empty slots left in the Deque. So, make sure that all the boundary cases are properly handled.

You are also required to write a menu-driven program which allows the user to use all of the aforementioned functionalities of the **Deque** class. A sample menu is shown below:

1. Create an empty Deque of a specified size
2. Insert value at Front
3. Insert value at Back
4. Remove a value from Front
5. Remove a value from Back
6. Display the Deque
7. Exit

Note: Follow these *good programming practices* when writing your code:

- There should be no memory leaks, dangling pointers, or any other type of runtime error in your program.
- Comment your code intelligently.
- Use meaningful variable and function names.
- Indent your code properly.
- Do not use any global or static variables.

😊 GOOD LUCK! 😊