Description | My Submissions | Hints/Editorial | AC Submissions | My Notes (0)

Circular Queue

? Ask Doubt

Time-Limit: 1 sec Score: 0/100 Difficulty :

Memory: 256 MB Accepted Submissions: 100 Relevant For: AZ-201

Description

Design your implementation of the circular queue. The circular queue is a linear data structure in which the operations are performed based on FIFO (First In First Out) principle and the last position is connected back to the first position to make a circle. It is also called "Ring Buffer".

In a normal Queue, we can insert elements until the queue becomes full. But once the queue becomes full, we can not insert the next element even if there is a space in front of the queue.

Implement the following functions:

1. **MyCircularQueue(k)**: Initializes the object with the size of the queue to be k.
2. **int Front()**: Gets the front item from the queue. If the queue is empty, return -1.
3. **int Rear():** Gets the last item from the queue. If the queue is empty, return -1.
4. **bool enQueue(int value):** Inserts an element into the circular queue. Return true if the operation is successful.
5. **bool deQueue()**: Deletes an element from the circular queue. Return true if the operation is successful.
6. **bool isEmpty():** Checks whether the circular queue is empty or not.
7. **bool isFull()**: Checks whether the circular queue is full or not.

Initially, the queue is empty.

You must solve the problem without using the built-in queue data structure in your programming language.

Input Format

The first line of input contains *Q* - the number of queries.
The second line contains *k.*
Each of the next lines contains queries of one of the types mentioned in the problem statement.

Constraints

$1 \le k, Q \le 10^5$

Sample Input 1

Copy

```
9
3
enQueue 1
enQueue 2
enQueue 3
enQueue 4
Rear
isFull
```

C++14[GCC]

Submit

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  class MyCircularQueue
```