Dequeue

Send Feedback

You need to implement a class for Dequeue i.e. for double ended queue. In this queue, elements can be inserted and deleted from both the ends.

You don't need to double the capacity.

You need to implement the following functions -

1. constructor

You need to create the appropriate constructor. Size for the queue passed is 10.

2. insertFront -

This function takes an element as input and insert the element at the front of queue. Insert the element only if queue is not full. And if queue is full, print -1 and return.

3. insertRear -

This function takes an element as input and insert the element at the end of queue. Insert the element only if queue is not full. And if queue is full, print -1 and return.

4. deleteFront -

This function removes an element from the front of queue. Print -1 if queue is empty.

5. deleteRear -

This function removes an element from the end of queue. Print -1 if queue is empty.

6. getFront -

Returns the element which is at front of the queue. Return -1 if queue is empty.

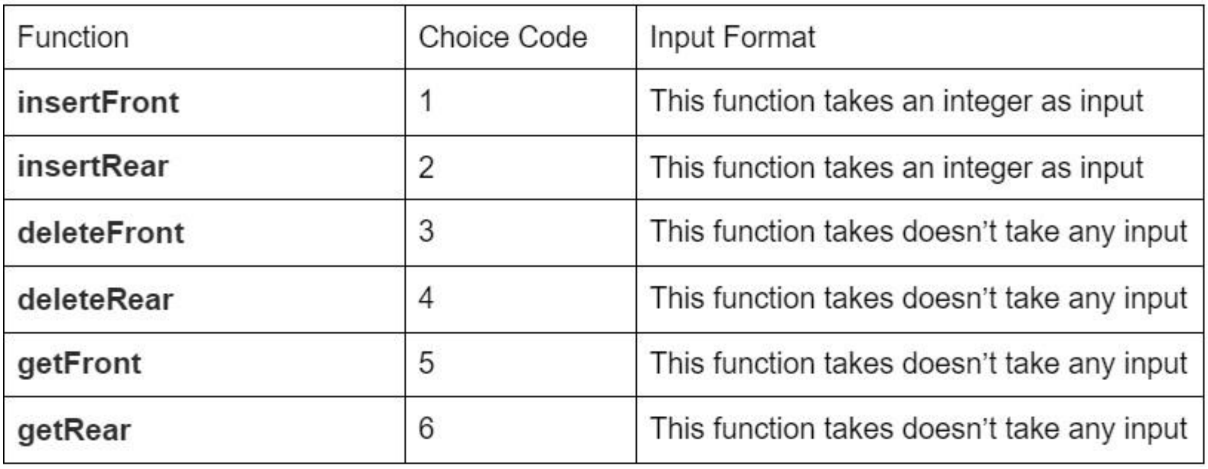
7. getRear -

Returns the element which is at end of the queue. Return -1 if queue is empty.

Input Format:

For C++ and Java, input is already managed for you. You just have to implement given functions.

For Python:
The choice codes and corresponding input for each choice are given in a new line. The input is terminated by integer -1. The following table elaborately describes the function, their choice codes and their corresponding input -



Function	Choice Code	Input Format
insertFront	1	This function takes an integer as input
insertRear	2	This function takes an integer as input
deleteFront	3	This function takes doesn't take any input
deleteRear	4	This function takes doesn't take any input
getFront	5	This function takes doesn't take any input
getRear	6	This function takes doesn't take any input

Output Format:

For C++ and Java, output is already managed for you. You just have to implement given functions.

For Python:
The output format for each function has been mentioned in the problem description itself.

Sample Input 1:

```
5
1
49
1
```

```
64
2
99
5
6
-1
```

Sample Output 1:

```
-1
64
99
```

Explanation:

The first choice code corresponds to getFront. Since the queue is empty, hence the output is -1.
The following input adds 49 at the top and the resultant queue becomes: 49.
The following input adds 64 at the top and the resultant queue becomes: 64 -> 49
The following input add 99 at the end and the resultant queue becomes: 64 -> 49 -> 99
The following input corresponds to getFront. Hence the output is 64.
The following input corresponds to getRear. Hence the output is 99.