Assignment No. 12.

Sumit Gulab Bhamare
SE Comp A08
Sub - DSL

Aim: To illustrate the concept of double-ended queue (deque).

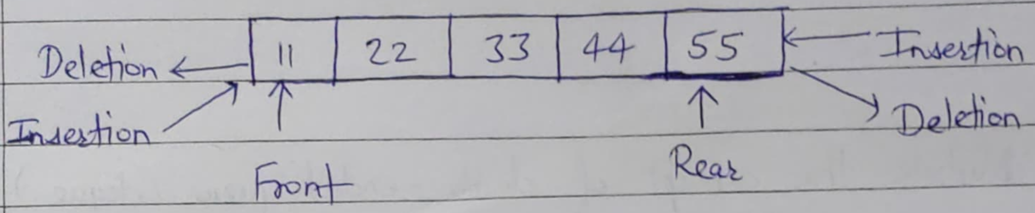
Problem Statement: A double-ended queue (deque) is a linear list in which additions & deletions may be made at either end. Obtain a data representation mapping a deque into a one-dimensional array. Write C++ program to simulate deque with functions to add & delete elements from either end of the deque.

Learning Objectives:
To understand concept of double-ended queue (deque)
To analyze the various functions of double-ended queue (deque)

Learning Outcome: Students will be able to implement stack & queue data structures & algorithms for solving different kinds of problems.

Theory:
~~Deque~~ DeQueue:
Dequeue is a data structure in which elements may be added to or deleted from the front or the rear. Like an ordinary queue, a double-ended queue is a data structure it supports the following operations: enq-front, enq-back, deq-front, deq-back, & empty. Dequeue can behave like a queue by using only enq-front & deq-front, and behaves like a stack by using only enq-front & deq-rear.

The DeQueue is represented as follows.



The output restricted Dequeue allows deletions from only one end & input restricted Dequeue allow insertions at only one end.

Input: Enter the elements in dequeue

Output: Add elements from either end, delete elements from both ends.

Algorithm:

Algorithm to add an element into DeQueue:

Assumptions: pointer f, r & initial values are -1, -1.
Q[] is an array
max represent the size of a queue

Algorithm to add an element from back:

```
Step 1: Start
Step 2: Check the queue is full or not as if (~~r=~~ r==max-1)
    if yes queue is full.
Step 3. If false update the pointer r as r=r+1
Step 4. Insert the element at pointer r as Q[r]= element
Step 5. Stop.
```

Algorithm to delete an element from the DeQueue

· Algorithm to delete an element from font:

Step 1. Start.
Step 2. Check the queue is empty or not as if (f==r) if yes queue is empty.
Step 3. If false update pointer f as f= f+1 & delete at position f as element = Q[f]
Step 4: If (f==r) reset pointer f & r as f=r=-1.
Step 5. Stop.

· Algorithm to delete an element from back:

Step 1. Start
Step 2: Check the queue is empty or not as if (f==r) if yes queue is empty
Step 3. If false delete element at position r as element =Q[r]
Step 4. Update pointer r as r=r-1
Step 5. If (f==r) reset pointer f & r as f=r=-1
Step 6. Stop.

Software required: g++ /gcc compiler.

Conclusion: Thus, we have studied the implementation of double-ended ~~deque~~ queue (deque).