

* Data Structure Lab (DSL):- Practical Number - 182 (Group - E)

Name:- Kaustubh Shrikant Khadra

Class:- Second Year Engineering.

Div:- A

Roll Number:-

Batch:-

Department:- Computer Department.

College:- AISSMS's IOIT.

Title:- To simulate deque with function to add and delete elements.

Aim:- Write a C++ program to simulate deque with function to add and delete elements.

Objective:-

- To study the concepts of deque.
- To understand the operation to insert and delete element from front and rear end.

Theory:-

In doubly-ended queue we can make use of both the ends for insertion of elements as well as use both ends for deletion of the elements.

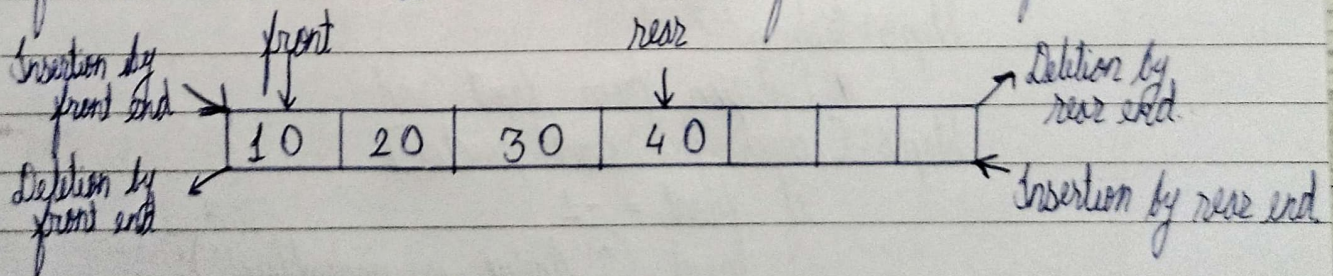


fig:- Doubly ended queue.

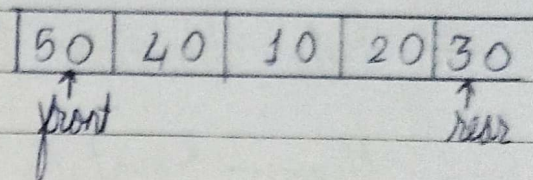
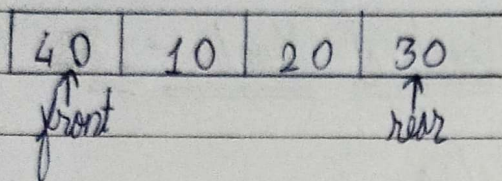


fig:- Insertion by front end.

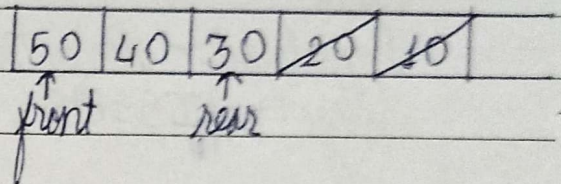
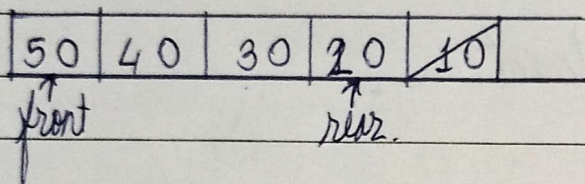


fig:- Deletion by rear end

Algorithm:-

Step 1: Start

Step 2: Initialize size of queue as Max=50.

Step 3: // Insertion at front end.

Step 3: [Check for the front position]

if (front == 1)

print ("cannot add item at front")

return;

Step 4: [Insert at front]

else:

front = front - 1;

q[front] = no.;

Step 5: Return

// Deletion from front end.

Step 6: [check for front pointer]

if front == -1.

print ("Queue is underflow");

return;

Step 7: [Perform deletion]

```
else  
    no = q[front]  
    print("Deletion element is", no);  
    if front == rear  
        front = 0;  
        rear = 0;
```

else:

```
front = front + 1;
```

Step 8: [Check for overflow]

```
if (rear == Max)
```

```
    print("Queue is overflow");
```

Step 9: - [Insert Element]

else :

```
rear = rear + 1;
```

```
q[rear] = no;
```

[Set rear and front pointer]

```
if rear == 0
```

```
    rear = 1;
```

```
if front == 0
```

```
    front = 1;
```

Step 10: [Check for the rear pointer]

```
if rear == 0
```

```
    print("Cannot delete value at rear");  
    return;
```

Step 11: [Check for and perform deletion]

else:

```
no = q[rear];
```

```
if front == rear
```

```
    front = rear = 0;
```


else rear = rear - 1;
print ("Deleted element is", no.);
Return.

Program: -

Output: -

Conclusion: -

Performed enqueue and dequeue operation on double-ended Queue.