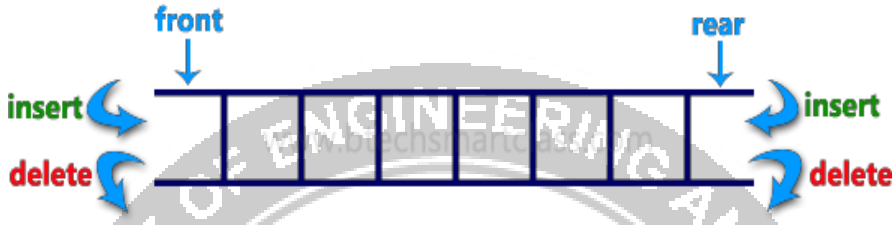
Double Ended Queue (Dequeue)

Double Ended Queue is also a Queue data structure in which the insertion and deletion operations are performed at both the ends (front and rear). That means, we can insert at both front and rear positions and can delete from both front and rear positions.



Double Ended Queue can be represented in TWO ways, those are as follows... Input Restricted Double Ended Queue

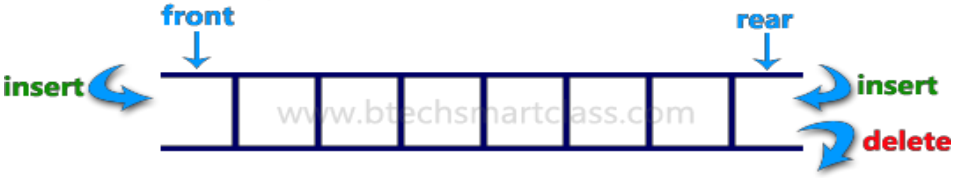
- Output Restricted Double Ended Queue
- Input Restricted Double Ended Queue

Input restricted double ended queue

In input restricted double ended queue, the insertion operation is performed at only one end and deletion operation is performed at both the ends.

Output Restricted Double Ended Queue

In output restricted double ended queue, the deletion operation is performed at only one end and insertion operation is performed at both the ends.



Program

```
#include<stdio.h>

#include<conio.h>

#define SIZE 100

void enQueue(int);

int deQueueFront();

int deQueueRear();

void enQueueRear(int);

void enQueueFront(int);

void display();

int queue[SIZE];

int rear = 0, front = 0;

int main()

{

        char ch;

        int choice1, choice2, value;

        printf("\n******* Type of Double Ended Queue *******\n");

        do

        {

        printf("\n1.Input-restricted deque \n");

        printf("2.output-restricted deque \n");
```

```
printf("\nEnter your choice of Queue Type : "); scanf("%d",&choice1);

switch(choice1)

{

case 1:

printf("\nSelect the Operation\n");

printf("1.Insert\n2.Delete from Rear\n3.Delete from Front\n4. Display");

do

{

printf("\nEnter your choice for the operation in c deque: ");

scanf("%d",&choice2);

switch(choice2)

{

        case 1:

                enQueueRear(value);

                display();

                break;

        case 2:

                value = deQueueRear();

                printf("\nThe value deleted is %d",value);

                display();

                break;
```

```
            case 3:

            value=deQueueFront();

            printf("\nThe value deleted is %d",value);

            display();

            break;

            case 4: display();

            break;

            default:printf("Wrong choice");
}

printf("\nDo you want to perform another operation (Y/N): ");

ch=getch();

getch();

 }while(ch=='y'||ch=='Y');

break;

case 2 :

printf("\n---- Select the Operation---- \n");

printf("1. Insert at Rear\n2. Insert at Front\n3. Delete\n4. Display");

do

{

            printf("\nEnter your choice for the operation: ");

            scanf("%d",&choice2);
```

```
        switch(choice2)
        {
        case 1:
                enQueueRear(value);
                display();
                break;
        case 2:
                enQueueFront(value);
                display();
                break;
        case 3:
                value = deQueueFront();
                printf("\nThe value deleted is %d",value);
                display();
                break;
        case 4: display();
                break;
        default:printf("Wrong choice");
        }
printf("\nDo you want to perform another operation (Y/N): ");
ch=getch();
```

```
        getch();

        }

         } while(ch=='y'||ch=='Y');

        break ;

        printf("\nDo you want to continue(y/n):");

        ch=getch();

        }while(ch=='y'||ch=='Y');

}

void enQueueRear(int value)

{

        char ch;

        if(front == SIZE/2)

        {

        printf("\nQueue is full!!! Insertion is not possible!!! ");

        return;

        }

        do

        {

        printf("\nEnter the value to be inserted:");

        scanf("%d",&value); queue[front] = value;
```

```
        front++;
        printf("Do you want to continue insertion Y/N");
        ch=getch();
        }while(ch=='y');
}
void enQueueFront(int value)
{
        char ch;
        if(front==SIZE/2)
        {
        printf("\nQueue is full!!! Insertion is not possible!!!");
        return;
        }
        do
        {
        printf("\nEnter the value to be inserted:");
        scanf("%d",&value);
        rear--;
        queue[rear] = value;
        printf("Do you want to continue insertion Y/N");
        ch = getch();
```

```
}

while(ch == 'y');

}

int deQueueRear()

{

        int deleted;

        if(front == rear)

        {

        printf("\nQueue is Empty!!! Deletion is not possible!!!");

        return 0;

        }

        front--;

        deleted = queue[front+1];

        return deleted;

}

int deQueueFront()

{

        int deleted;

        if(front == rear)

        {

        printf("\nQueue is Empty!!! Deletion is not possible!!!");
```

```
        return 0;

        }

        rear++;

        deleted = queue[rear-1];

        return deleted;

}



void display()

{

        inti;

        if(front == rear)

        printf("\nQueue is Empty!!! Deletion is not possible!!!")

        else{

        printf("\nThe Queue elements are:");

        for(i=rear; i< front; i++)

        {

        printf("%d\t ",queue[i]);

        }

        }

}
```