3/11/25 13/10/25

Q) WAP to simulate the working of a circular queue of integers using an array. Provide the following operations. Insert, Delete & Display. The program should print appropriate messages for queue empty & queue overflow conditions

Pseudo code

```
void enqueue(int x){
    if(front==-1 && rear==-1){
        front=rear=0;
        queue[rear]=x;
    }
else if((front=rear+1 || front=0 && rear=N-1){
        Printf("queue is full");
}
```

```
    else{
        rear=(rear+1)%N;
        queue[rear]=x;
    }
void dequeue(){
    if (front==-1 && rear==-1)
    {
        Printf("queue is empty!");
    }
    else if (front==rear){
        printf("deleted element is: %d", queue[front]);
        front=rear=-1;
    }
    else{
        printf("deleted element is: %d", queue[front]);
        front=(front+1)%N;
    }
}

Void display(){
    i=front;
    while(i!=rear){
        printf("%d", queue[front]);
        i++;
    }
    Printf("%d", queue[rear]);
}
```

3/11/25

code:-

```
#include<stdio.h>
# define N 5
int queue[N]
int front=-1, rear=-1;
void enqueue(int x){
    if((front==0 && rear==N-1) || (front==rear+1)){
        Printf("The Queue is Full\n");
    }
    else if(front==-1 && rear==-1){
        front=rear=0;
        queue[rear]=x;
    }
    else{
        rear=(rear+1)%N;
        queue[rear]=x;
    }
}
void dueueue(){
    if(front==-1 && rear==-1){
        Printf("The Queue is Empty!\n");
    }
    else if(front==rear){
        Printf("The deleted element is %d\n",
                queue[front]);
        front=-1;
        rear=-1;
    } else {
        Printf("The deleted element is %d\n",
        queue[front]);
        front=(front+1)%N;
} }
```

```
Void display(){
    if (front==-1 && rear==-1){
        Printf("The Queue is Empty!\n");
    }else{
        int i=front;
        Printf("The Queue is :\n");
        while(i!=rear){
            PrintF("%d\n", queue[i]);
            i=(i+1)%N;
        } Printf("%d\n", queue[rear]);
    }
}
Void main(){
    int ch,y;
    do{
    Printf("Enter element from 1 to 4 to
        insert, delete, display & stop");
    scanf("%d", &ch);
    switch(ch){
        case 1:
            Printf("\nEnter element to
                insert:");
            Scanf("%d", &y);
            enqueue(y);
            break;
        case 2:
            dequeue();
            break;
        Case 3:
            display();
            break;
```

```
            case 4:
                break;
            default:
                printf("Your choice is Invalid!");
        }
    } while(ch!=4);
}
```

Olp:-

Enter elemnt from 1 to 4 to insert, delete, display & stop: 1

Enter element to insert: 22

Enter element from 1 to 4 to insert, delete display & stop: 1

Enter element to insert: 23

Enter element from 1 to 4 to insert, delete display & stop: 1

Enter element to insert: 24

Enter element from 1 to 4 to insert, delete display & stop: 1

Enter element to insert: 25

Enter element from 1 to 4 to insert, delete display & stop: 2

deleted element is 2

Enter element from 1 to 4 to insert, delete display & stop: 3

The queue is:

23
24
25