1BM19CS053
Divyanshu Dhanwan

```
#define size 5
int front = -1;
int rear = -1;
int queue[rear];

Void enque (int n)
{
    if (front == 0 && rear == size-1)
    {
        printf("Queue is full");
    }
    else if ( front == rear+1)
    {
        printf("Queue is full");
    }
    else if (front == -1 && rear == -1)
    {
        front ++;
        rear ++;
        queue[rear] = n;
    }
    else
    {
        rear = (rear+1) % size;
        queue[rear] = n;
    }
}
```

```
int dequeue()
{
    if(front == -1 && rear == -1)
    {
        return -1;
    }
    else
    {
        int ele;
        ele = queue[front];
        if(front == rear)
        {
            front = -1;
            rear = -1;
        }
        else
        {
            front = (front+1) % size;
        }
        return ele;
}

Void display()
{
    if(front == -1 && rear = -1
    {
        printf("Queue is empty");
    }
```

```
else
{
    printf("Queue Contents : \n");
    if(front <= rear)
    {
        for(int i = front ; i<=rear ; i++)
        {
            printf("%d \n", queue[i]);
        }
    }
    else
    {
        for(i=front ; i<=size-1 ; i++)
        {
            printf("%d \n");
        }
        for(i=0; i<=rear ; i++)
        {
            printf("%d \n");
        }
    }
}
}
```