

Topic

Assignment : 1

Q. 1) write a program implementing insert, delete & di-
 rect operation of circular queue
 in C++



```
#include <bits/stdc++.h>
using namespace std;
struct queue
```

```
{ int size;
  int *arr;
  queue (int s)
{}
```

```
front = rear = -1;
size = S;
arr = new int [S];
}
```

```
void enqueue (int value);
int dequeue();
void displayqueue();
}
```

```
void queue :: enqueue (int value)
{
```

```
if ((front == 0) & (rear == size - 1))
```



```

        {
            if (rear == (front - 1) % (size - 1))
                return;
            else if (front == -1)
                Front = rear = 0;
            arr[rear] = value;
        }
        else if (rear == size - 1 && front != 0)
        {
            rear = 0;
            arr[rear] = value;
        }
        else
        {
            rear++;
            arr[rear] = value;
        }
    }
    if (queue : deQueue())
    {
        if (front == -1)
        {
            printf("The queue is empty");
            return INT_MIN;
        }
        else
        {
            int dData = arr[front];
            arr[front] = arr[front + 1];
            front = front + 1;
        }
    }
}

```

```

    }
    else if (front == size - 1)
        front = 0;
    else
        front++;
    return data;
}

void queue :: displayqueue()
{
    if (front == -1)
    {
        printf("THE QUEUE IS EMPTY");
        return;
    }
    printf("THE ELEMENTS IN CIRCULAR QUEUE ARE : ");
    if (rear >= front)
    {
        for (int i = front; i < rear; i++)
            printf("%d ", arr[i]);
        printf("\n");
    }
    else
    {
        for (int i = front; i < size; i++)
            printf("%d ", arr[i]);
        printf("\n");
    }
}

```

```
For (int i=0; i < = rear; i++)
    printf("%d", arr[i]);
}
int main()
{
    queue q(5);
    q.enqueue(14);
    q.enqueue(22);
    q.enqueue(13);
    q.enqueue(-6);
    q.displayqueue();
    printf("Deleted value = %d", q.dequeue());
    printf("Deleted value = %d", q.dequeue());
    q.displayqueue();
    q.enqueue(7);
    q.enqueue(9);
    q.enqueue(10);
    q.enqueue(5);
    q.enqueue(14);
    q.dequeue();
    q.enqueue(20);
    q.dequeue();
    return 0;
}
```

→ QP: Element in circular queue are : 14 22 13 -6
Deleted value = 14
Deleted value = 22
Element in circular queue are = 13 -6
Element in circular queue are = 13 -6 9 20 5
Queue is full.