

CIRCULAR QUEUE

Solution:

```
#include<stdio.h>

# define MAX 100

int cqueue_arr[MAX];

int front = -1;

int rear = -1;

void insert(int item)
{
    if((front == 0 && rear == MAX-1) || (front == rear+1))
    {
        printf("Queue Overflow \n");
        return;
    }
    if(front == -1)
    {
        front = 0;
        rear = 0;
    }
    else
    {
        if(rear == MAX-1)
            rear = 0;
        else
            rear = rear+1;
    }
    cqueue_arr[rear] = item ;
}
```

```

void deletion()
{
    if(front == -1)
    {
        printf("Queue Underflown\n");
        return ;
    }
    printf("Element deleted from queue is : %d\n",cqueue_arr[front]);
    if(front == rear)
    {
        front = -1;
        rear=-1;
    }
    else
    {
        if(front == MAX-1)
            front = 0;
        else
            front = front+1;
    }
}

void display()
{
    int front_pos = front,rear_pos = rear;
    if(front == -1)
    {
        printf("Queue is empty\n");
        return;
    }
}

```

```

    }

    printf("Queue elements :\n");
    if( front_pos <= rear_pos )
        while(front_pos <= rear_pos)
        {
            printf("%d ",cqueue_arr[front_pos]);
            front_pos++;
        }
    else
    {
        while(front_pos <= MAX-1)
        {
            printf("%d \n",cqueue_arr[front_pos]);
            front_pos++;
        }
    }
    front_pos = 0;
    while(front_pos <= rear_pos)
    {
        printf("%d\n ",cqueue_arr[front_pos]);
        front_pos++;
    }
}

printf("\n");
}

int main()
{
    int choice,item;

    printf("-----START-----\n");

    do

```

```

{

printf("1.Insert\n");
printf("2.Delete\n");
printf("3.Display\n");
printf("4.Quit\n");
printf("Enter your choice : ");
scanf("%d",&choice);

switch(choice)
{
case 1 :
    printf("ENTER THE ELEMENT TO BE INSERTED IN QUEUE:\t ");
    scanf("%d", &item);
    insert(item);
    break;
case 2 :
    deletion();
    break;
case 3:
    display();
    break;
case 4:
    break;
default:
    printf("CHOOSED OPTION IS WRONG\n");
}
}while(choice!=4);
printf("-----ENDS-----\n");

```

```
return 0;
```

```
}
```

```
C:\WINDOWS\SYSTEM32\cmd.exe
-----START-----
1.Insert
2.Delete
3.Display
4.Quit
Enter your choice : 1
ENTER THE ELEMENT TO BE INSERTED IN QUEUE:      10
1.Insert
2.Delete
3.Display
4.Quit
Enter your choice : 1
ENTER THE ELEMENT TO BE INSERTED IN QUEUE:      20
1.Insert
2.Delete
3.Display
4.Quit
Enter your choice : 1
ENTER THE ELEMENT TO BE INSERTED IN QUEUE:      60
1.Insert
2.Delete
3.Display
4.Quit
Enter your choice : 1
ENTER THE ELEMENT TO BE INSERTED IN QUEUE:      66
1.Insert
2.Delete
3.Display
4.Quit
Enter your choice : 1
ENTER THE ELEMENT TO BE INSERTED IN QUEUE:      25
1.Insert
2.Delete
3.Display
4.Quit
Enter your choice : 2
Element deleted from queue is : 10
1.Insert
2.Delete
3.Display
4.Quit
Enter your choice : 3
Queue elements :
20 60 66 25
1.Insert
2.Delete
3.Display
4.Quit
```

C:\WINDOWS\SYSTEM32\cmd.exe

Enter your choice : 5

CHOOSSED OPTION IS WRONG

1.Insert

2.Delete

3.Display

4.Quit

Enter your choice : 4

-----ENDS-----

(program exited with code: 0)

Press any key to continue . . .