EXPERIMENT NO. 5

IMPLEMENTATION OF CIRCULAR QUEUE USING ARRAY

Aim: Write A Program To Implement Circular Queue Using Array.

Theory: Queue is a particular kind of **Abstract Data Type** or Collection in which the entities in the collection are kept in order and the principal operations on the collection are the addition of entities to the **REAR** terminal position, known as **Enqueues**, and removal of entities from the **FRONT** terminal position, known as **Dequeues**. This makes the queue a ***First-In-First-Out (FIFO) Data Structure***.

Circular Queue is a Data Structure that uses a single, fixed-size Queue as if it were connected **end-to-end**. This structure lends itself easily to Queueing Data Streams.

Use:

The Useful Property of a Circular Queue is that it does not need to have its elements shuffled around when one is consumed. *(If a Non-Circular Queue were used then it would be necessary to shift all elements when one is consumed.)* In other words, the **Circular Queue** is well-suited as a **FIFO Queue** while a Standard, **Non-Circular Queue** is well suited as a **LIFO Queue.**

Algorithm:

FOR INSERT OPERATION:

1. Start

2. If (FRONT = = 0 and REAR = = N-1) or (FRONT = = REAR + 1) Then

3. Print: Circular Queue Overflow.

4. Else If (FRONT and REAR= = -1) Then [Check if QUEUE is empty]

(a) Set FRONT = 0

```
(b) Set REAR = 0
(c) Set QUEUE [REAR] = ITEM
5.    Else If (REAR = = N-1 and FRONT! =0) Then [If REAR reaches end if QUEUE]
6.    Set REAR = 0
(a) Set QUEUE [REAR] = ITEM
7.  Else
8.  Set REAR = REAR + 1 [Increment REAR by 1] [End of Step 5 If]
9.  Set QUEUE [REAR] = ITEM
10. Print: ITEM inserted [End of Step 2 If]
11. Exit
```

FOR POP OPERATION:

```
1. Start
2. If (FRONT = = -1) Then [Check for Underflow]
3. Print: Circular Queue Underflow.
4. Else
5. If (FRONT = = REAR) Then [If only element is left]
(a) Set FRONT = -1
(b) Set REAR = -1
6.  Else If (FRONT == N-1) Then [If FRONT reaches end if QUEUE]
7.  Set FRONT = 0
8.  Else
9.  Set FRONT = FRONT + 1 [Increment FRONT by 1] [End of Step 6 If]
```

10. Print: ITEM deleted [End of Step 2 If]

11. Exit.

Program:

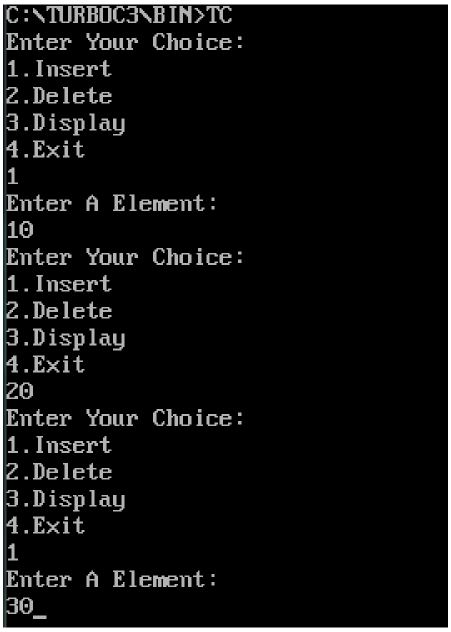
```
#include<stdio.h>
# define size 5
int queue [size];
int f=-1;
int r=-1;
int empty()
{
if(f==-1)
return 1;
else
return 0;
}
int full()
{
if((f==0)&&(r==(size-1))||(f==r+1))
return 1;
else
return 0;
}
void insert(int x)
{
if(full()==1)
printf("Circular Queue Overflow\n");
else if((f==-1)&&(r==-1))
{
f=r=0;
queue[r]=x;
}
else if((r==(size-1))&&(f!=0))
{
```

```
r=0;
queue[r]=x;
}
else
{
r++;
queue[r]=x;
}
}
void delet()
{
if(empty()==1)
printf("Circular
Queue Empty\n");
else if(f==r)
f=r=-1;
else if(f==(size-1))
f=0;
else
f++;
}
void display()
{
int i;
if(empty()==1)
printf("Circular
Queue Empty\n");
else if(f<=r)
{
printf("Contents:\n"
);
for(i=f;i<=r;i++)
{
printf("%d\n",queue
[i]);
}
}
else if(f>r)
{
printf("Contents:\n"
);
```

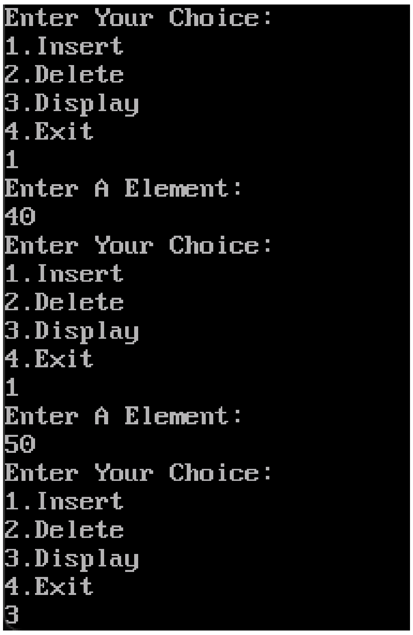
```
for(i=f;i<=(size-1);i++)
{
printf("%d\n",queue[i]);
}
for(i=0;i<=r;i++)
{
printf("%d\n",queue[i]);
}
}
}
int main()
{
int x,c;
do
{
printf("Enter Your Choice:\n");
printf("1.Insert\n");
printf("2.Delete\n");
printf("3.Display\n");
printf("4.Exit\n");
scanf("%d",&c);
switch(c)
{
case 1:
printf("Enter A Element:\n");
scanf("%d",&x);
insert(x);
break;
case 2: delet();
break;
case 3: display();
break;
}
}
while(c!=4);
```

```
return 0;
}
```

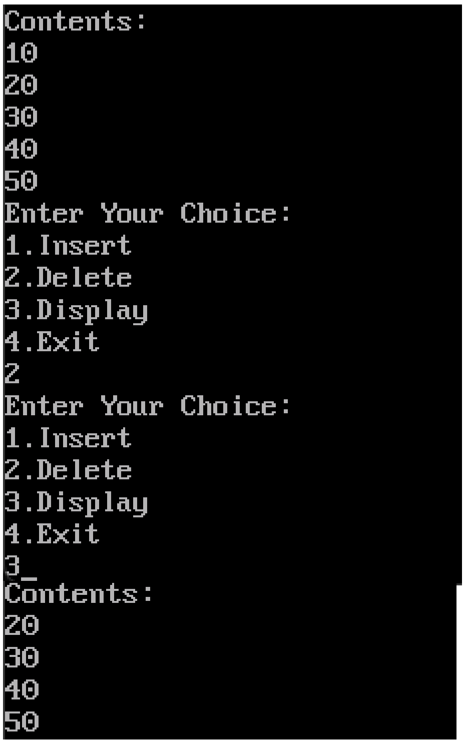
Output:



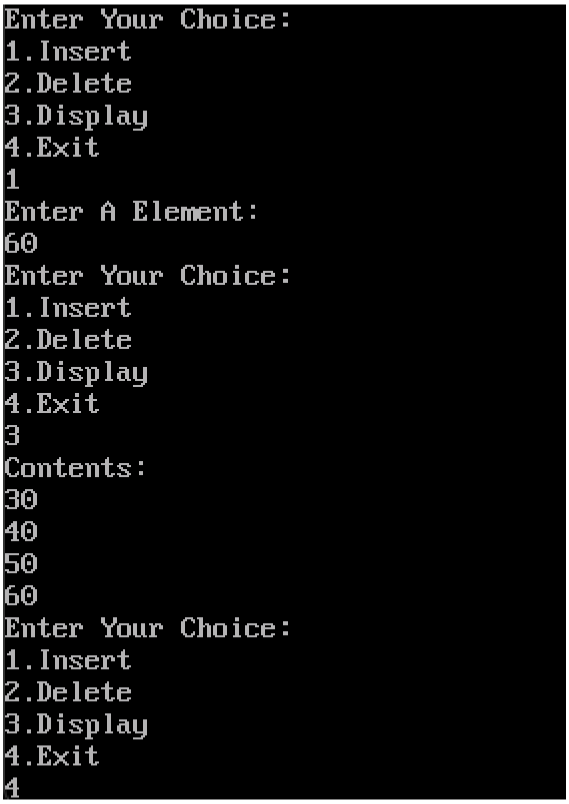
```
C:\TURBOC3\BIN>TC
Enter Your Choice:
1.Insert
2.Delete
3.Display
4.Exit
1
Enter A Element:
10
Enter Your Choice:
1.Insert
2.Delete
3.Display
4.Exit
20
Enter Your Choice:
1.Insert
2.Delete
3.Display
4.Exit
1
Enter A Element:
30_
```



```
Enter Your Choice:
1.Insert
2.Delete
3.Display
4.Exit
1
Enter A Element:
40
Enter Your Choice:
1.Insert
2.Delete
3.Display
4.Exit
1
Enter A Element:
50
Enter Your Choice:
1.Insert
2.Delete
3.Display
4.Exit
3
```



```
Contents:
10
20
30
40
50
Enter Your Choice:
1.Insert
2.Delete
3.Display
4.Exit
2
Enter Your Choice:
1.Insert
2.Delete
3.Display
4.Exit
3_
Contents:
20
30
40
50
```



```
Enter Your Choice:
1.Insert
2.Delete
3.Display
4.Exit
1
Enter A Element:
60
Enter Your Choice:
1.Insert
2.Delete
3.Display
4.Exit
3
Contents:
30
40
50
60
Enter Your Choice:
1.Insert
2.Delete
3.Display
4.Exit
4
```