```c
#include<stdio.h>
#include<process.h>
#define QUE_SIZE 3
int item,front=0,rear=-1,q[QUE_SIZE],count=0;
void insertrear()
{
 if(count==QUE_SIZE)
 {
  printf("Queue Overflow\n");
  return;
 }
 rear=(rear+1)%QUE_SIZE;
 q[rear]=item;
 count++;
}
int deletefront()
{
 if(count==0)
   return -1;
 item=q[front];
 front=(front+1)%QUE_SIZE;
 count--;
 return item;
}
void display()
{
 int i,f,r;
```

circularqueue.c *Untitled2

```
28      if(count==0)
29      {
30       printf("Queue is empty\n");
31       return;
32      }
33      f=front;
34      r=rear;
35      printf("Contents of the queue are:\n");
36      for(i=1;i<=count;i++)
37      {
38        /*if(front<=rear)
39        {
40          printf("%d\n",q[f]);
41          f=(f+1)%QUE_SIZE;
42        }
43        else
44        {
45         printf("%d\n",q[r]);
46         r=(r+1)%QUE_SIZE;
47        }*/
48        printf("%d\n",q[f]);
49        f=(f+1)%QUE_SIZE;
50      }
51      }
52      int main()
53      {
54       int choice;
```

```
55  for(;;)
56  {
57   printf("1.Insert at rear\n2.Delete Front\n3.Display\n4.Exit\n");
58   printf("Enter choice:\n");
59   scanf("%d",&choice);
60
61   switch(choice)
62   {
63    case 1: printf("Enter the item to be inserted:\n");
64            scanf("%d",&item);
65            insertrear();
66            break;
67    case 2: item=deletefront();
68            if(item==-1)
69              printf("Queue is empty\n");
70            else
71              printf("Item deleted=%d\n",item);
72            break;
73    case 3: display();
74            break;
75    case 4: exit(0);
76   }
77  }
78 }
79
```

"C:\Users\Neha Chadaga\Desktop\circularqueue.exe"

```
1.Insert at rear
2.Delete Front
3.Display
4.Exit
Enter choice:
1
Enter the item to be inserted:
34
1.Insert at rear
2.Delete Front
3.Display
4.Exit
Enter choice:
1
Enter the item to be inserted:
76
1.Insert at rear
2.Delete Front
3.Display
4.Exit
Enter choice:
1
Enter the item to be inserted:
23
1.Insert at rear
2.Delete Front
3.Display
4.Exit
Enter choice:
1
Enter the item to be inserted:
89
Queue Overflow
1.Insert at rear
2.Delete Front
3.Display
4.Exit
Enter choice:
3
Contents of the queue are:
34
76
23
1.Insert at rear
2.Delete Front
3.Display
4.Exit
Enter choice:
2
Item deleted=34
```

```
1.Insert at rear
2.Delete Front
3.Display
4.Exit
Enter choice:
3
Contents of the queue are:
76
23
1.Insert at rear
2.Delete Front
3.Display
4.Exit
Enter choice:
1
Enter the item to be inserted:
80
1.Insert at rear
2.Delete Front
3.Display
4.Exit
Enter choice:
3
Contents of the queue are:
76
23
80
1.Insert at rear
2.Delete Front
3.Display
4.Exit
Enter choice:
2
Item deleted=76
1.Insert at rear
2.Delete Front
3.Display
4.Exit
Enter choice:
3
Contents of the queue are:
23
80
```

```
1.Insert at rear
2.Delete Front
3.Display
4.Exit
Enter choice:
2
Item deleted=76
1.Insert at rear
2.Delete Front
3.Display
4.Exit
Enter choice:
3
Contents of the queue are:
23
80
1.Insert at rear
2.Delete Front
3.Display
4.Exit
Enter choice:
4

Process returned 0 (0x0)   execution time : 33.369 s
Press any key to continue.
```