```
6..#include <stdio.h>
#include<stdlib.h>
#define SIZE 5
int CQ[SIZE];
int front=-1;
int rear=-1, ch;
int IsCQ_Full();
int IsCQ_Empty();
void CQ_Insert(int );
void CQ_Delet();
void CQ_Display();
void main()
{
printf("1.Insert\n2.Delete\n3.Display\n4.Exit\n");
while(1)
{
int ele;
printf("Enter your choice\n");
scanf("%d",&ch);
switch(ch)
{
case 1: if(IsCQ_Full())
printf("Circular Queu Overflow\n");
else
{
printf("Enter the element to be inserted\n");
scanf("%d",&ele);
CQ_Insert(ele);
}
break;
case 2: if(IsCQ_Empty())
printf("Circular Queue Underflow\n");
else
CQ_Delet();
break;
case 3: if(IsCQ_Empty())
printf("Circular Queue Underflow\n");
else
CQ_Display();
break;
case 4: exit(0);
}
}
}
```

```
void CQ_Insert(int item)
{
if(front==-1)
front++;
rear = (rear+1)%SIZE;
CQ[rear] =item;
}
void CQ_Delet()
{
int item;
item=CQ[front];
printf("Deleted element is: %d",item);
front = (front+1)%SIZE;
}
void CQ_Display()
{
int i;
if(front==-1)
printf("Circular Queue is Empty\n");
else
{
printf("Elements of the circular queue are..\n");
for(i=front;i!=rear;i=(i+1)%SIZE)
{
printf("%d\t",CQ[i]);
}
printf("%d\n",CQ[i]);
}
}
int IsCQ_Full()
{
if(front ==(rear+1)%SIZE)
return 1;
return 0;
}
int IsCQ_Empty()
{
if(front == -1)
return 1;
else if(front == rear)
{
printf("Deleted element is: %d",CQ[front]); front=-1;
return 1;
}
return 0;
}
/* output
1.Insert
2.Delete
3.Display
4.Exit
Enter your choice
1
Enter the element to be inserted
1
Enter your choice
1
Enter the element to be inserted
2
Enter your choice
1
Enter the element to be inserted
3
Enter your choice
3
Elements of the circular queue are..
1       2       3
Enter your choice
2
Deleted element is: 1Enter your choice
3
Elements of the circular queue are..
2       3
Enter your choice
```