main.c

```c
#include <stdio.h>
#include <stdlib.h>
#define MAX 3

int front=-1;
int rear=-1;

int queue[MAX];

void Enque(int);
int Deque();
void display();
int main(int argc, char **argv)
{
  int option;
    int item;
    do{
        printf("Circular Queue\n");
        printf("\n 1. Insert to Queue (EnQueue)");
        printf("\n 2. delete from the Queue (DeQueue)");
        printf("\n 3. Display the content ");
        printf("\n 4. Exit\n");
        printf("Enter the option :");
        scanf("%d",&option);
        switch(option)
        {
            case 1:  printf("Enter the element\n");
                     scanf("%d",&item);
                     Enque(item);
                     break;
            case 2: item=Deque();
                    if(item==-1)
```

eccableV... C Stop ■

main.c

```
34                          else
35                          printf("Removed element from the queue %d\n",item)
                            ;
36                          break;
37                  case 3: display();
38                          break;
39                  case 4: exit(0);
40              }
41      } while (option!=4);
42    return 0;
43  }
44
45  void Enque(int ele)
46  {
47      if(((front == 0 && rear == MAX - 1))|| (front == rear + 1) )
48      {
49          printf("Queue is full\n");return;
50
51      }
52      else
53      {
54        rear=(rear+1)%MAX;
55        queue[rear]=ele;
56        if(front ==-1)
57            front=0;
58
59
60      }
61  }
62  int Deque()
63  {
64      int item;
```

Stop ■

```
    if((front == -1)&&(rear == -1))
    {

        return(-1);
    }
    else
    {
        item=queue[front];

        if(front==rear)
        {
            front=-1;
            rear=-1;
        }
        else
        {
            front=(front+1)%MAX;
        }
        return item;
    }

}

void display()
{
  if (front == -1 && rear == -1)
  {
    printf("Queue is empty\n");
  }
  else
  {
    printf("Queue contents : \n");
    if (front <= rear)
```

```
85
86  }
87
88  void display()
89  {
90    if (front == -1 && rear == -1)
91    {
92      printf("Queue is empty\n");
93    }
94    else
95    {
96      printf("Queue contents : \n");
97      if (front <= rear)
98      {
99        for (int i = front; i <= rear; i++)
100       {
101         printf("%d\t", queue[i]);
102       }
103     }
104     else
105     {
106       for (int i = front; i <= MAX-1; i++)
107       {
108         printf("%d\t", queue[i]);
109       }
110       for (int i = 0; i <= rear; i++)
111       {
112         printf("%d\t", queue[i]);
113       }
114     }
115   }
116 }
```