Date: / /

DISHA - B
1BM19CS050

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

int front = -1
int rear = -1
int queue[MAX];
void Enqueue(int);
int Dequeue();
void display();
int main(int argc, char **argv)
{
    int option;
    int item;
    do{
        printf("Circular Queue\n");
        printf("\n1. Insert to Queue(Enqueue)");
        printf("\n2. delete from the Queue(Dequeue)");
        printf("\n3. Display the content");
        printf("\n4. 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 == -999)
                        printf("Queue is empty");
                    else
```

Date: / /

```
        printf("Removed element from the queue %d",
                                        item);
        break;
    case 3: display();
        break;
    case 4: exit(0);
    }
  }while(option!=4);
  return 0;
}
void enque(int ele)
{
  if(((front==0 && rear==MAX-1)||(front==rear+1))
  {
    printf("Queue is full\n");
    return;
  }
  else
  {
    rear=(rear+1)%MAX;
    queue[rear]=ele;
    if(front==-1)
      front=0;
  }
}
int Deque()
{
  int item;
  if((front==-1)&&(rear==-1))
  {
    return(-999);
  }
```

DISHA·B
1BM19CS050.

Date: / /

DISHA · B.
1BM19CS050

```
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");
  }
  else
  {
    printf("Queue contents:\n");
    if (front <= rear)
    {
      for(int i=front; i<=rear; i++)
      {
        printf("%d\n", queue[i]);
      }
    }
    else
    {
      for(int i=front; i<= MAX-1; i++)
      {
        printf("%d\n", queue[i]);
      }
      for (int i=0; i<=rear; i++)
      {
        printf("%d\n", queue[i]);
      }
    }
  }
}
```