1BM19CS044
Deepak   Circular Queue

```
#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 choice;
   int item;
   do {
    Printf("Circular Queue\n");
    Printf(1.insert to Queue(Enqueue|");
printf("2.delete from the Queue (DeQueue)");
printf(" 3. Display the content");
printf(" 4. Exit\n");
printf("Enter your choice:");
 scanf("%d", &choice);
  switch(choice)
 {
  case 1 : printf("Enter the element to be
                          enqueued");
         scanf("%d", & item);
         Enque(item);
         break;
  case 2 : item=Deque();
     if( item = = -1)
        printf("Queue is empty");
      else
   printf("Dequeued element from the queue
                %d", item);
   break;
 case 3 : display();
        break;
 case 4 : exit(0);
 default : printf("entered wrong choice\n");
 }
```

```
} while(choice != 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(-1);
}
else
{
    item = queue[front];
    if (front == rear)
    {
        front = -1;
        rear = -1;
    }
    else
    {
        front = (front +1) % MAX;
    }
    return item;
}
}

Void display()
{
    int i;
```

```
if(((front==-1) && (rear==-1))
        || (front==rear))
{
  printf("Queue is empty\n");
    return;
}
else
{
 printf("Elements in the Queue are");
  for(i=front; i<=rear; i++)
    printf("%d\n", queue[i]);
}
}
```