DARSHAN NINGTHOUJAM
1BM19CS041

Circular Queue

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 4

int front = -1;
int rear = -1;
int queue[MAX];

void Enque(int);
int Deque();
void display();
int main()
{
  int option;
  int item;
  do{
    printf("\n CIRCULAR QUEUE \n");
    printf("\n 1. Insert to QUEUE");
    printf("\n 2. delete from QUEUE");
    printf("\n 3. display QUEUE ");
    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 == -999)
         printf("Queue is empty\n");
         else
         printf("Removed element from the queue
                  is %d \n", 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");
 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("\n Queue contents:\n");
  for (i=front; i<=rear; i++)
    printf("%d\t", queue[i]);
  }
}
```