NAME: ANSHUL H
USN: 1BM19CS020

LAB-4

Q-4) WAP to stimulate the working of a circular queue of ~~integers~~ integers using an array. Provide the following operations-
a) Insert b) Delete c) Display.
The program should print appropriate messages for queue empty and queue overflow conditions

```
# include <stdio.h>
# include <stdlib.h>
# define size 3
int front = -1;
int rear = -1;
void insert (int, int[]);
int delete (int[]);
void display (int[]);
int main()
{   int ch, queue[size];
    int item;
    do{
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", &ch);
switch(ch)
{ case 1: printf("Enter the element\n");
          &scanf("%d", &item);
```

NAME: ANSHUL H. SURANA
USN : 1BM19CS020

```
            insert(item, queue);
            break;
case 2:     item = delete(queue);
            if(item == -9999)
            printf("Queue Underflow\n");
            else
            printf("Removed element: %d", item);
            break;
            else
case 3:     display(queue);
            break;
case 4:     exit(0);
        }
}while(ch!=4);
    return 0;
}
void insert(int ele, int queue[])
{   if((rear+1)%size == front)
        printf("Queue Overflow\n");
    else
{   rear = (rear+1)%size;
    queue[rear] = ele;
    if((rear+1)%size == front)
    printf("Queue is full\n");
    if(front == -1)
        front = 0;
}
}
int delete(int queue[])
{   int item;
    if(front == -1)
    return -9999;
```

Anshul H. Surana

```
else
{ item = queue[front];
    if (front == rear)
{ front = -1;
    rear=-1;
} else {
{front = (front+1) %size; }
    return item;
  }
}
void display (int queue[])
{ int i;
 if (front==-1) && (rear==-1))
 printf("Queue is empty\n");
 else
{ if (front<=rear)
  for(i=front; i<=rear; i++)
  printf("%d", queue[i]);
else
{
  for(i=front; i<size; i++)
  printf("%d", queue[i]);
  for(i=0; i<=rear; i++)
  printf("%d", queue[i]);
  }
 }
}
```


Anshul H. Surana