3.b) WAP to simulate the working of a circular queue of integers using an array. Provide the following operations: Insert, Delete & Display The program should print appropriate messages for queue empty and queue overflow conditions

```
#include<stdio.h>
#define MAX 5 //maximum size of queue
int queue[MAX];
int front = -1,rear = -1;

//functions to insert an element in the circular queue
void insert(int value)
{
   if(front == (rear + 1) % MAX)
   {
      printf("Queue Overflow ! Cannot insert %d ",value);
   }
   else
   {
      if(front == -1)
      {
         //first insertion
         front = 0;
         rear = 0;
      }
      else
      {
         rear = (rear + 1) % MAX;
      }
      queue[rear] = value;
      printf("%d inserted into queue : \n",value);
   }
```

```
}

//function to delete an element from the queue
void delete()
{
   if(front == -1)
   {
      printf("Queue Underflow!Queue is empty\n");

   }
   else
   {
      printf("Deleted element :%d\n",queue[front]);
      if(front == rear)
      {
         //queue becomes empty
         front = -1;
         rear = -1;
      }
      else
      {
         front = (front +1 ) % MAX;
      }
   }
}

//functtion to display elements of the circular queue
void display()
{
   if(front == -1)
   {
```

```
        printf("Queue is empty.\n");
    }
    else
    {
        printf("Queue elements : ");
        int i = front;
        while(1)
        {
            printf("%d\n",queue[i]);
            if(rear == i)
            {
                break;

            }
            i = (i + 1) % MAX;
        }
    }
}

void displayFront()
{
    printf("The Element at front is %d",queue[front]);
}
void displayRear()
{
    printf("The Element at front is %d",queue[rear]);
}

//main function
int main()
{
```

```
    int choice,value;
    while(1)
    {
        printf("\n Circular Queue Operations : \n");
        printf("1.INSERT\n2.DELETE\n3.DISPLAY\n4.EXIT\n5.Display Front\n6.Display Rear\n");
        printf("Enter your choice : ");
        scanf("%d",&choice);

        switch(choice)
        {
            case 1 :
                printf("Enter value to insert : ");
                scanf("%d",&value);
                insert(value);
                break;
            case 2 :
                delete();
                break;
            case 3 :
                display();
                break;
            case 4:
                printf("Exiting program.\n");
                return 0;
            default:
                printf("Invalid choice! Please try again.\n");
        }
    }
    return 0;
}
```

OUPUT :

```
C:\Users\Mohammed Javeed\OneDrive\Desktop\Javeed\C Tutorial>cd "c:\Users\Mohammed Javeed\OneDrive\Desktop\Javeed\C Tutorial\" && gcc 03B.c -o 03B && "c:\Users\Mohammed Javeed\OneDrive\Desktop\Javeed\C Tutorial\"03B

 Circular Queue Operations :
1.INSERT
2.DELETE
3.DISPLAY
4.EXIT
5.Display Front
6.Display Rear
Enter your choice : 1
Enter value to insert : 10
10 inserted into queue :

 Circular Queue Operations :
1.INSERT
2.DELETE
3.DISPLAY
4.EXIT
5.Display Front
6.Display Rear
Enter your choice : 1
Enter value to insert : 20
20 inserted into queue :

 Circular Queue Operations :
1.INSERT
2.DELETE
3.DISPLAY
4.EXIT
5.Display Front
6.Display Rear
Enter your choice : 1
Enter value to insert : 30
30 inserted into queue :
```

```
 Circular Queue Operations :
1.INSERT
2.DELETE
3.DISPLAY
4.EXIT
5.Display Front
6.Display Rear
Enter your choice : 1
Enter value to insert : 40
40 inserted into queue :

 Circular Queue Operations :
1.INSERT
2.DELETE
3.DISPLAY
4.EXIT
5.Display Front
6.Display Rear
Enter your choice : 1
Enter value to insert : 50
50 inserted into queue :

 Circular Queue Operations :
1.INSERT
2.DELETE
3.DISPLAY
4.EXIT
5.Display Front
6.Display Rear
Enter your choice : 1
Enter value to insert : 60
Queue Overflow ! Cannot insert 60
```

```
 Circular Queue Operations :
1.INSERT
2.DELETE
3.DISPLAY
4.EXIT
5.Display Front
6.Display Rear
Enter your choice : 3
Queue elements : 10
20
30
40
50

 Circular Queue Operations :
1.INSERT
2.DELETE
3.DISPLAY
4.EXIT
5.Display Front
6.Display Rear
Enter your choice : 5
Invalid choice! Please try again.

 Circular Queue Operations :
1.INSERT
2.DELETE
3.DISPLAY
4.EXIT
5.Display Front
6.Display Rear
Enter your choice : 6
Invalid choice! Please try again.

 Circular Queue Operations :
1.INSERT
2.DELETE
3.DISPLAY
4.EXIT
5.Display Front
6.Display Rear
Enter your choice : 2
Deleted element :10
```

```
 Circular Queue Operations :
1.INSERT
2.DELETE
3.DISPLAY
4.EXIT
5.Display Front
6.Display Rear
Enter your choice : 2
Deleted element :20

 Circular Queue Operations :
1.INSERT
2.DELETE
3.DISPLAY
4.EXIT
5.Display Front
6.Display Rear
Enter your choice : 2
Deleted element :30

 Circular Queue Operations :
1.INSERT
2.DELETE
3.DISPLAY
4.EXIT
5.Display Front
6.Display Rear
Enter your choice : 2
Deleted element :40

 Circular Queue Operations :
1.INSERT
2.DELETE
3.DISPLAY
4.EXIT
5.Display Front
6.Display Rear
Enter your choice : 2
Deleted element :50
```

```
 Circular Queue Operations :
1.INSERT
2.DELETE
3.DISPLAY
4.EXIT
5.Display Front
6.Display Rear
Enter your choice : 2
Queue Underflow!Queue is empty

 Circular Queue Operations :
1.INSERT
2.DELETE
3.DISPLAY
4.EXIT
5.Display Front
6.Display Rear
Enter your choice :
```