WAP to simulate the working of a circular queue of 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

Code:

```
#include  <stdio.h>
#include
<stdlib.h>  #define
N 5

int q[N];
int front = -1, rear = -1;
void insert(int);
int deleteq();
void display();
int main()
{
   int n, choice;
   printf("\n1.Insert\n2.Delete\n3.Display\n4.Exit\n");
   do
   {

      printf("\nEnter your option : \n");
      scanf("%d", &choice);
      switch (choice)
      {
      case 1:
         printf("Enter the number to be inserted in the queue : \n");
         scanf("%d", &n);
         insert(n);
         break;
      case 2:
```

```
            n = deleteq();
            if (n != -1)
                printf("\n The number deleted is : %d\n", n);
            break;
        case 3:
            display();
            break;
        case 4:
            exit(0);
            break;
        default:
            printf("Invalid option\n");
            exit(0);
            break;
        }
    } while (choice != 4);
}
void insert(int num)
{
    if ((front == 0 && rear == N - 1) || (rear == (front - 1)))
        printf("\n OVERFLOW");
    else if (front == -1 && rear == -1)
    {
        front = rear = 0;
        q[rear] = num;
    }
    else if (rear == N - 1 && front != 0)
    {
        rear = 0;
        q[rear] = num;
    }
    else
    {
        rear++;
        q[rear] = num;
    }
}
int deleteq()
{
    int val;
    if (front == -1 && rear == -1)
    {
        printf("\n UNDERFLOW");
        return -1;
```

```
    }
    val = q[front];
    if (front == rear)
        front = rear = -1;
    else
    {
        if (front == N - 1)
            front = 0;
        else
            front++;
    }
    return val;
}
void display()
{
    int i;
    printf("\n");
    if (front == -1 && rear == -1)
        printf("\n QUEUE IS EMPTY");
    else
    {
        if (front < rear)
        {
            for (i = front; i <= rear; i++)
                printf("\t %d", q[i]);
        }
        else
        {
            for (i = front; i < N; i++)
                printf("\t %d", q[i]);
            for (i = 0; i <= rear; i++)
                printf("\t %d", q[i]);
        }
    }
}
```

Output:

```
1.Insert
2.Delete
3.Display
4.Exit

Enter your option :
1
Enter the number to be inserted in the queue :
1

Enter your option :
1
Enter the number to be inserted in the queue :
2

Enter your option :
1
Enter the number to be inserted in the queue :
3

Enter your option :
1
Enter the number to be inserted in the queue :
4

Enter your option :
1
Enter the number to be inserted in the queue :
5

Enter your option :
1
Enter the number to be inserted in the queue :
6

 OVERFLOW
```

```
Enter your option :
3

        1       2       3       4       5
Enter your option :
2

 The number deleted is : 1

Enter your option :
2

 The number deleted is : 2

Enter your option :
2

 The number deleted is : 3

Enter your option :
2

 The number deleted is : 4

Enter your option :
2

 The number deleted is : 5
```

```
Enter your option :
2

 UNDERFLOW
Enter your option :
3


 QUEUE IS EMPTY
Enter your option :
4
```