Lab Program - 3

3) WAP to simulate the working of a 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.

```
#include <stdio.h>
#define SIZE 5
int front = -1;
int rear = -1;
int queue[SIZE];

int IsEmpty(){
    if ((front == -1 && rear == -1)){
        return 1;
    }
    else
    return 0;
}

int IsFull(){
    if (rear == SIZE-1){
        printf("The queue is full!");
        return 1;
    }
    else
    return 0;
}

void Enqueue(int x){
    if (IsFull()){
        printf("Element cannot be
                added.\n");
```

```
        return;
    }
    else if (IsEmpty()) {
        front = 0;
        rear = 0;
    }
    else {
        rear++;
    }
    queue[rear] = x;
}

void Dequeue(){
    if (IsEmpty()){
        printf("The queue is empty!");
        return;
    }
    else {
        if (front == rear){
            printf("The element dequeued
                    is :%d \n", queue[front]);
            front++;
        }
    }
}

void display(){
    if (IsEmpty()){
        printf("The queue is empty!");
        return;
    }
    else {
        printf("The elements of the
        queue are as follows: \n");
```

```
        for (int i=front; i<=rear; i++) {
            printf ("\t%d", queue[i]);
        }
    }
}

int main(){
    int n, i;
    while (i!=-1) {
        printf ("\n Enter your choice: \n
                1. Enqueue \n 2. Dequeue \n
                3. Display \n 4. Exit \n");
        scanf ("%d", &n);

        if (n==1) {
            int item;
            printf ("Enter the element to
                    be added to the queue:\n");
            scanf ("%d", &item);
            Enqueue (item);
        }
        else if (n==2) {
            Dequeue ();
        }
        else if (n==3) {
            display ();
        }
        else
        i=-1;
    }
}
```