3) Write a program to simulate queue of integers using an array

```
#include <stdio.h>
#include <stdlib.h>

#define MAX 3
int qu[MAX];
int front = -1;
int rear = -1;

void insert();
int delete_q();
void display();

int main()
{
    while (1)
    {
        int choice,d;
        printf("\n1. insert \t 2.delete \t 3.display \t 4.exit\n");
        scanf("%d", &choice);
        switch (choice)
        {
        case 1:
            insert();
            break;
        case 2:
            d=delete_q();
             if (d!= -1)
                printf("\n The number deleted is : %d", d);
            break;
        case 3:
            display();
            break;
        case 4:
            exit(0);
        }
    }
}

void insert()
{
    if (rear == MAX - 1)
    {
        printf("Queue is Full\n");
        return;
    }
    printf("Enter the element to be inserted\n");
    int a;
    scanf("%d", &a);
    if ((front == -1) && (rear == -1))
    {
        front = rear = 0;
    }
    else
    {
        rear++;
    }
    qu[rear] = a;
}
```

```
int delete_q()
{
    int val;
    if(front==-1 ||rear<front)
    {
        printf("Underflow\n");
        return -1;
    }
    else{
        val=qu[front];
        front++;
        if(front>rear)
        {
            front=rear=-1;
        }
        return val;
    }
}
void display()
{
    printf("the elements are:\t");
    for (int i = front; i <= rear; i++)
    {
        printf("%d \t", qu[i]);
    }
}
```

output:

```
1. insert □ 2.delete □ 3.display □ 4.exit
1
Enter the element to be inserted
12

1. insert □ 2.delete □ 3.display □ 4.exit
1
Enter the element to be inserted
13

1. insert □ 2.delete □ 3.display □ 4.exit
1
Queue is Full
1. insert □ 2.delete □ 3.display □ 4.exit
3
The elements are :
12
13
1. insert □ 2.delete □ 3.display □ 4.exit
2
12 is deleted from the queue

1. insert □ 2.delete □ 3.display □ 4.exit
2
Queue is empty
```