Implementation of stack using queue

Brute Force :

2 queue use krenge - orig stack and another temp

- Pehle temp me insert krenge fir original wale me jitne bhi elements honge unko q2 me daldenge
- Swap krenge so that all elements are in orig stack and then repeat
- Basically isme dusri queue ka use kr re h for reversing so that last inserted element queue k front se miljaye
- q2 queue basically acts as temporary queue for storing the inserted element each time i.e the last in element.

```
class MyStack {
public:
    queue<int> q1;
    queue<int> q2;
    void push(int x) {
        q2.push(x);
        while(!q1.empty())
        {
            q2.push(q1.front());
            q1.pop();
        }
        swap(q2,q1);
    }

    int pop() {
        if(q1.size()!=0)
        {
            int ans=q1.front();
            q1.pop();
            return ans;
        }
        return -1;
    }

    int top() {
        return q1.front();
    }

    bool empty() {
        if(q1.size()==0)
            return 1;
        return 0;
    }
};

/**
 * Your MyStack object will be instantiated and called as such:
 * MyStack* obj = new MyStack();
 * obj->push(x);
 * int param_2 = obj->pop();
 * int param_3 = obj->top();
 * bool param_4 = obj->empty();
 */
```

Using a single queue : (Optimisation in space)

Ek single queue me sbse last me jo insert hua tha usko chhodke baki sare elements ko push krdenge firse. Kyunki koi bhi oper perfo

Ek loop chlayenge queue k size-1 tk so that jo last me insert hua h element wo front pe hi rahe and baki sare elements ko pop krke d

```
class MyStack {
public:
    queue<int> q;
    int size;
    MyStack() {
        size=0;
    }

    void push(int x) {
        size++;
        q.push(x);
        for(int i=0;i<q.size()-1;i++)
        {
            q.push(q.front());
            q.pop();
        }
    }

    int pop() {
        int ans;
        if(q.size()!=0)
        {
            ans=q.front();
            q.pop();
            size--;
            return ans;
        }
        return -1;
    }

    int top() {
        return q.front();
    }

    bool empty() {
        if(size==0)
            return true;
        return false;
    }
};

/**
 * Your MyStack object will be instantiated and called as such:
 * MyStack* obj = new MyStack();
 * obj->push(x);
 * int param_2 = obj->pop();
 * int param_3 = obj->top();
 * bool param_4 = obj->empty();
 */
```