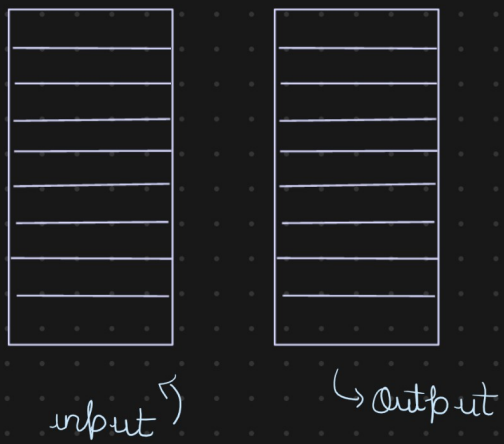
Queue Implementation using Stack,

a) push
b) pop
c) peek
d) top




```
#include <bits/stdc++.h>
using namespace std;
class Queue {
public:
    stack<int> input;
    stack<int> output;
    Queue() {

    }
    void helperfunc(){
        if(output.empty()){
            while(!input.empty()){
                output.push(input.top());
                input.pop();
            }
        }
    }
    void push(int n){
        input.push(n);
    }
    int pop(){
        helperfunc();
        int val = output.top();
        output.pop();
```

```
        return val;
    }
    int peek() {
        helperfunc();
        return output.top();
    }
    bool empty() {
        return input.empty() && output.empty();
    }
```

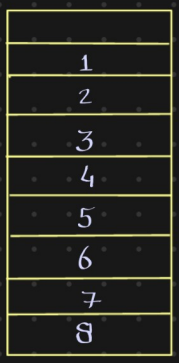
Queue method in cpp

```
#include<iostream>
#include<queue>
using namespace std;
queue<int> q;
q.push() // enqueue
q.pop() // dequeue
q.front()// get the front element
q.back()// get the back element
q.size()
q.empty()
```

practice Sheet (10th April, 2025)

a) push b) pop c) peek d) top

```
#include <iostream>
using namespace std;

class Queue {
public:
    stack <int> input;
    stack <int> output;

    Queue(){

    }

    void helperfunc() {
        if(output.empty()){
        while(!input.empty()){
            output.push(input.top());
            input.pop();
            }
        }
    }

    void push(int val){
        intput.push(val);
    }
```



↳ Input

↳ Output

```
int pop(){
    helperfunc();
    int res = output.top();
    output.pop();
    return res;
}

int peek(){
    helperfunc();
    int val = output.top();
    return val;
}

bool empty(){
    return output.empty() && input.empty;
}
}
```