Stack →

```
#include<iostream>
#include<stack>
stack<int> st; // Initializing a stack
```

Stack method

```
st.push(10); pushes an element in the stack
st.pop(); pops the top element of the stack
st.top(); access the top element;
st.empty(); checks if the stack is empty
st.size(); Get the stack size.
```

Stack Implementation →

```
#include <bits/stdc++.h>
using namespace std;

class stack{
public:
    vector<int> stack_;
    stack(){};

    void push(int n){
        stack_.pushback(n);
    }

    int pop(){

        int res = stack_[stack_.size-1];
```

```
        stack_.pop_back();
        return res;
    }
```

→ Practice (8th, April, 2025)

```
#include <bits/stdc++.h>
using namespace std;
#include <bits/stdc++.h>
using namespace std;

class stack {

public:
    vector<int> stack_;
    stack()
    {

    }
    void push(int n){
        stack_.push_back(n);
    }
    int pop(){
        int res = stack_[stack_.size()-1];
        stack_.pop_back();
        return res;
    }
```

pratice Sheet 12/4/2025

```
#include <bits/stdc++.h>
using namespace std;

class stack {
public:
    vector<int> stack;
    stack(){

    }
    void push(int n){
        stack.push_back(n);
    }
    int pop(){
        int res = stack[stack.size()-1];
        stack.pop_back();
        return res;
    }
};
```