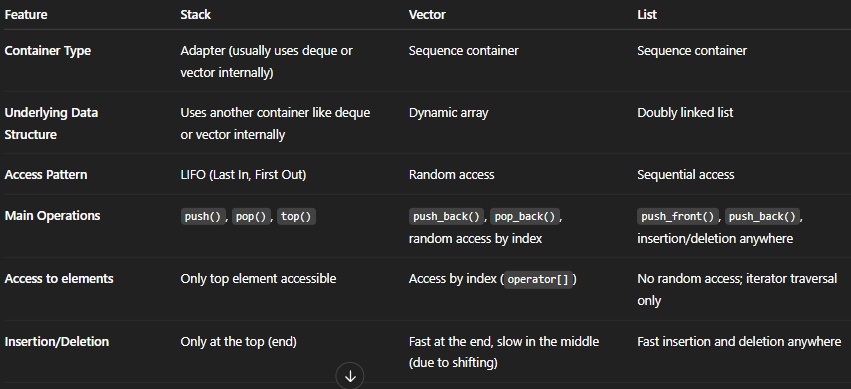
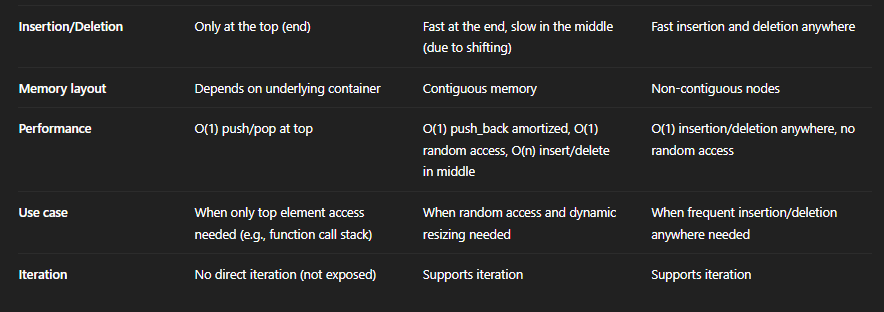
**LIFO / FILO / Last Come First Serve**

****

****

**CODE 🡪**

#include<iostream>

#include<stack>

using namespace std;

void printingstackelements(stack<int>& a) // remember to add & before name of variable and NOT its TYPE

{

// for (auto it = a.begin(); it != a.end(); it ++ ) ... no begin and end functions in stack

while (!a.empty()) {

cout << "element: " << a.top() << endl; // print the top element

a.pop(); // remove it and then print the next

}

}

int main() {

stack<int> mystack;

// empty(), size(), top(), push(), pop()

mystack.push(0);

mystack.push(1);

mystack.push(2);

mystack.push(3);

mystack.pop();

if (mystack.empty())

cout << "stack is empty" << endl;

else

cout << "stack is not empty" << endl;

cout << "stack size: " << mystack.size() << endl;

cout << "top most element: " << mystack.top() << endl;

cout << "\nall elements: " << endl;

printingstackelements(mystack);

return 0;

}

**DUDE REMEMBER :**

1. **You don’t print it using iterators.... you print the top and then push it till the stack isn’t empty using a While loop instead of a for loop.**
2. **When the elements are printed, they are printed in the reverse order in the manner they were pushed into the stack ,as LAST IN FIRST OUT LIFO rule.**