Google Developer Student Club
Meerut Institute of Engineering and Technology

DSA BOOTCAMP

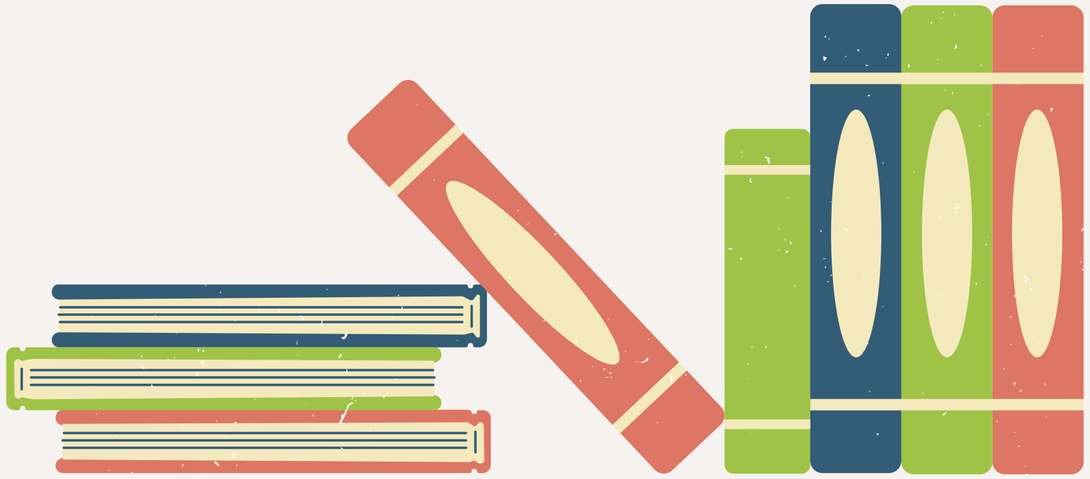
SESSION 5

24 FEB2023

Google Developer Student Club
Meerut Institute of Engineering and Technology

TODAY'S TOPIC :

RECURSION AND STACK



RECURSION

A Function is said to be recursively defined, if a function containing either a call statement to itself or a call statement to a second function that may eventually result in a call statement back to the original fuction.

A Recursive function must have the following properties :

- There must be certain criteria, called **BASE CRITERIA** , for which the function does not call itself.
- Each time the function does call itself the argument of the function must be closer to a base value.

Types of Recursion

Direct Recursion

In direct recursion, the function calls itself directly

```
void direct()
{
    //code
    direct();
}
```

Indirect Recursion

If a function calls itself indirectly from another function, then this type of recursion is called indirect recursion

```
void indirect()
{
    //code
    func();
}
void func()
{
    //code
    inderct();
}
```

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

int sum(int num)
{
    if (num != 0)
    {
        return num + sum(num - 1);
    }
    return 0;
}

int main()
{
    int num;
    cout << "Enter the number : ";
    cin >> num;
    cout << "Sum is : " << sum(num);

    return 0;
}
```

Example 1 :

Enter the number : 5

Sum is : 15

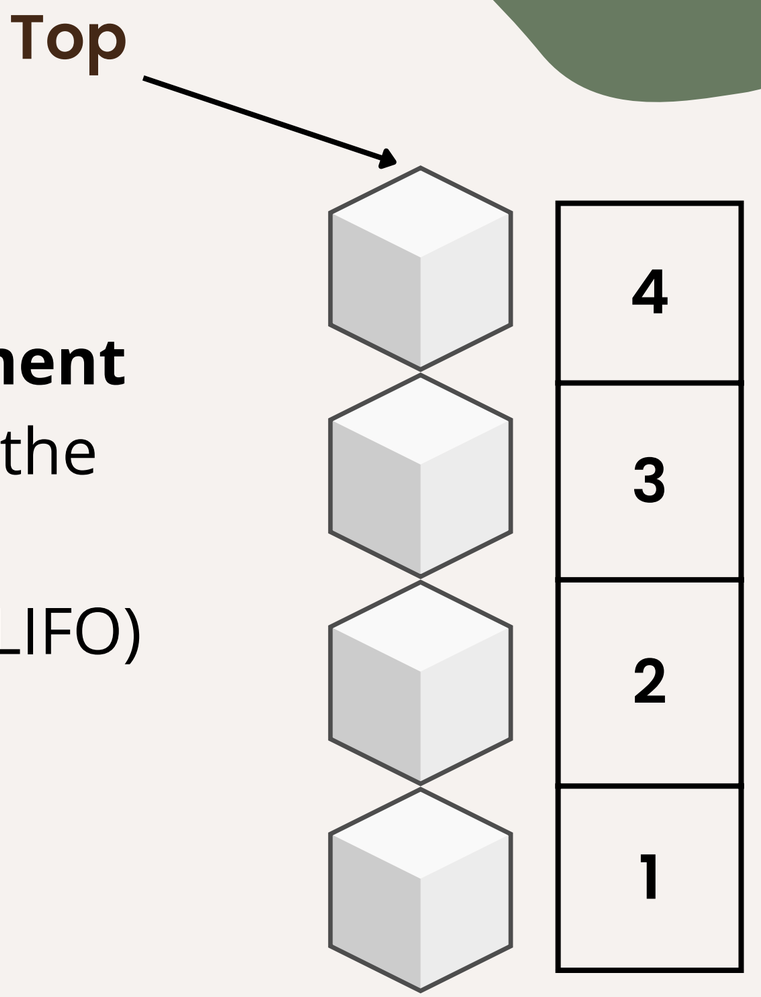
Example 2:

Enter the number : 2

Sum is : 3

STACK

- A stack is a linear data structure in which all insertions and deletion are made at one end called the **TOP OF STACK.**
- A stack is a Ordered collection of **homogeneous** data elements**.**



Here, the **last element** inserted will be on the **top of the stack.** Since deletion is done from the same end, **last element** inserted will be the **first element** to be removed out from the stack and so on.

That's why the stack is also called **LAST - IN - FIRST - OUT** (LIFO)

Operations on Stack

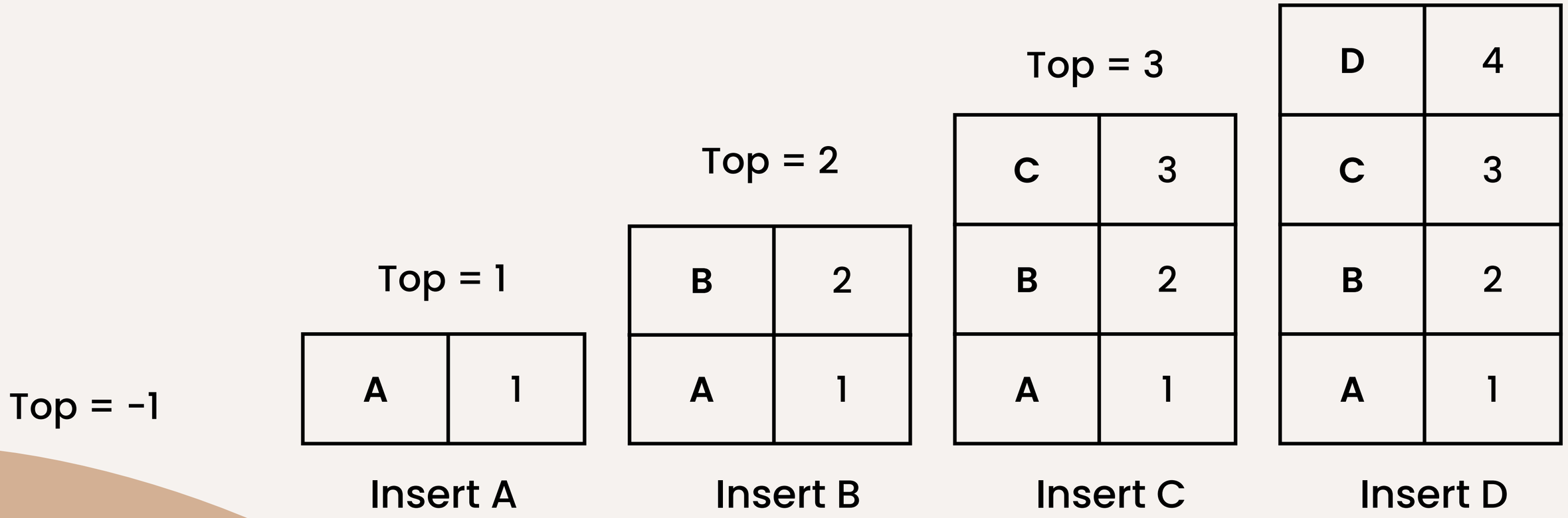
PUSH

The process of adding a new element at the top of stack is called **PUSH operation**

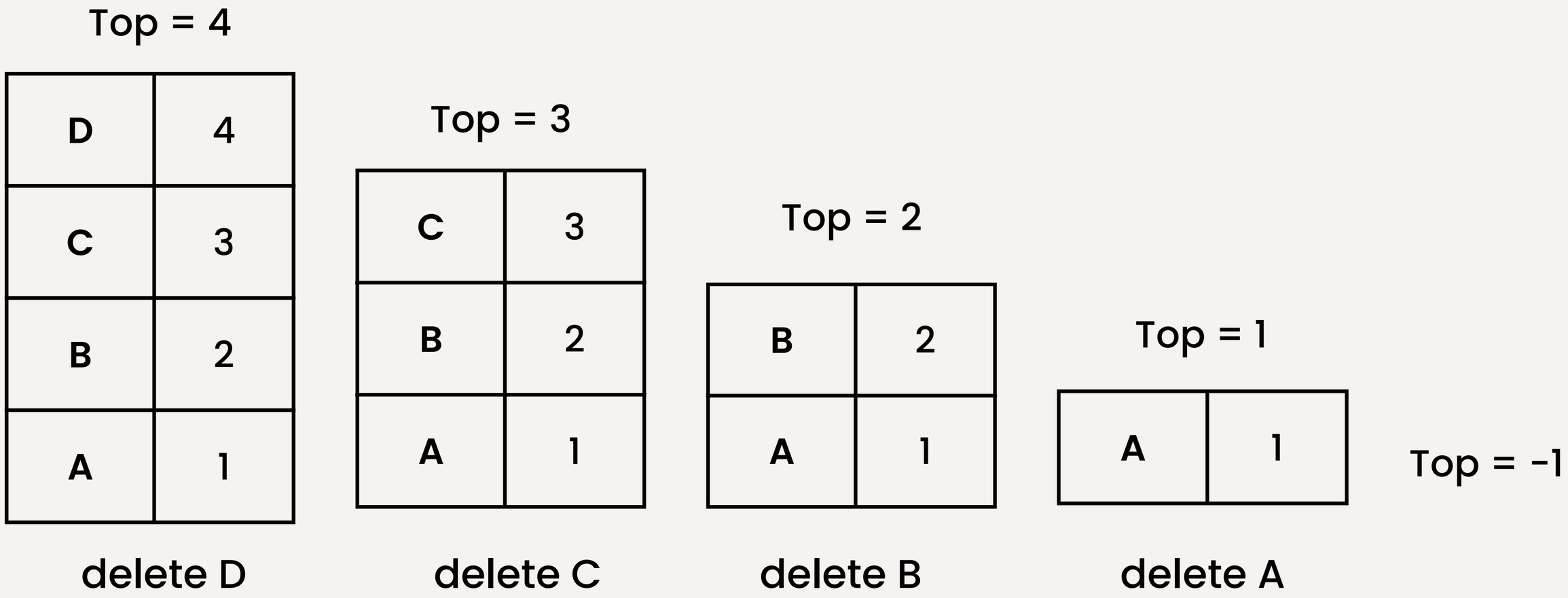
POP

The process of deleting an element from the top of stack is called **POP operation**

PUSH illustration:



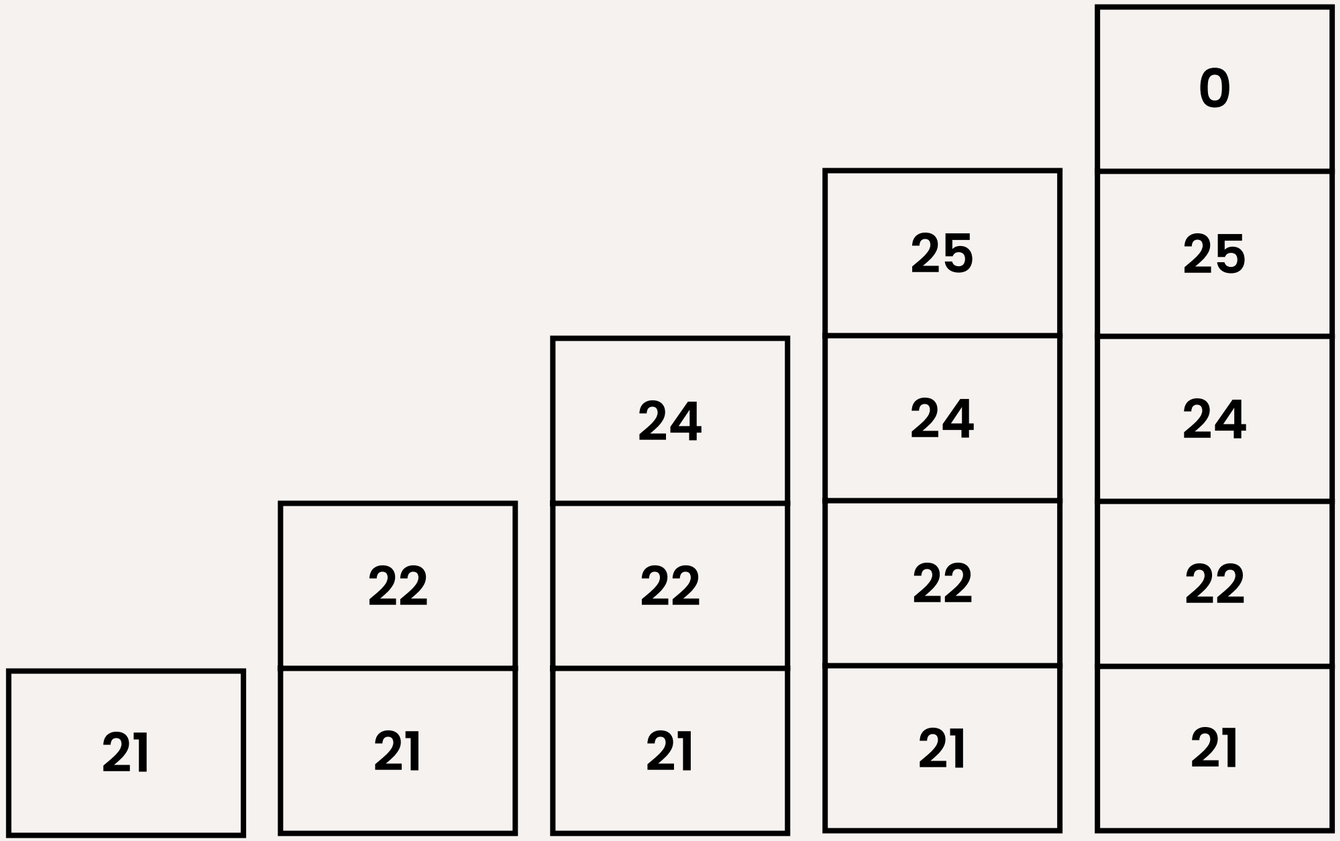
POP illustration:

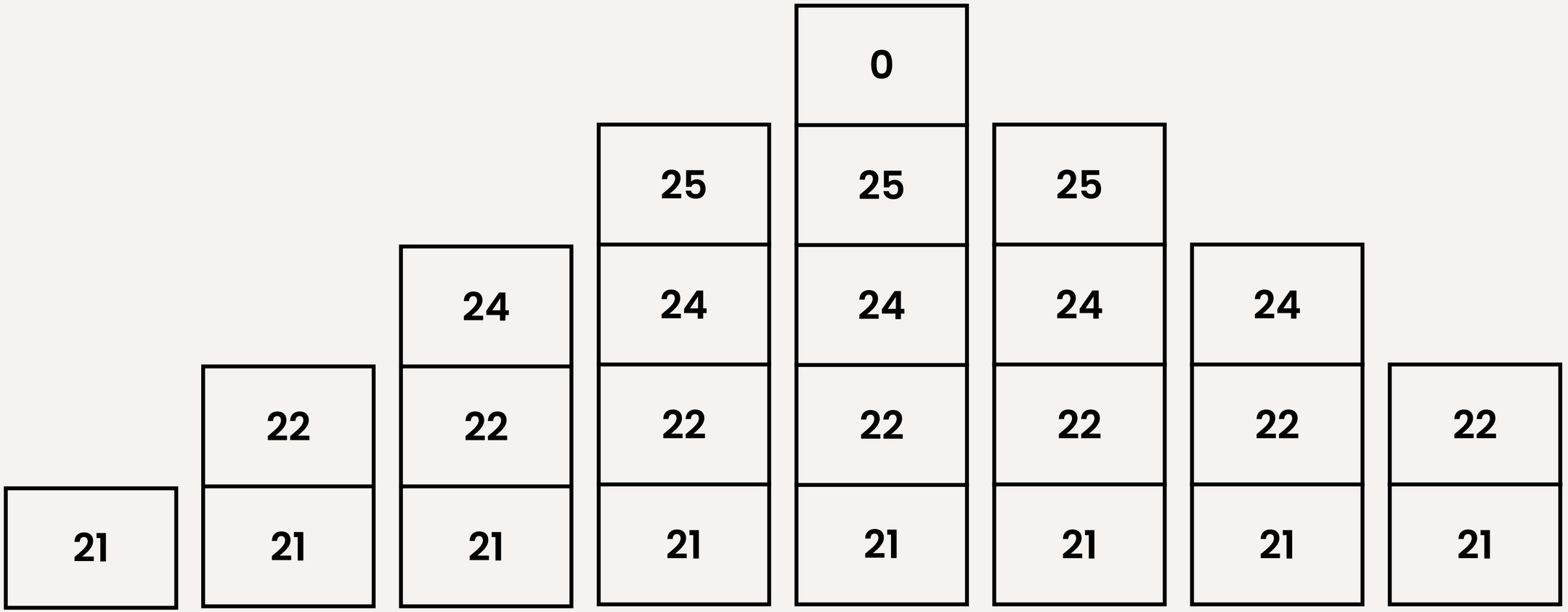


How to declare stack:

```
int main() {
    stack<int> s;
    s.push(21);
    s.push(22);
    s.push(24);
    s.push(25);
    int num=0;
    s.push(num);
    s.pop();
    s.pop();
    s.pop();

    while (!s.empty()) {
        cout << s.top() <<" ";
        s.pop();
    }
}
```




0
25
25
25
24
24
24
24
24
22
22
22
22
22
22
22
21
21
21
21
21
21
21

Google Developer Student Club
Meerut Institute of Engineering and Technology

Thank You

@cpTeamGDSC