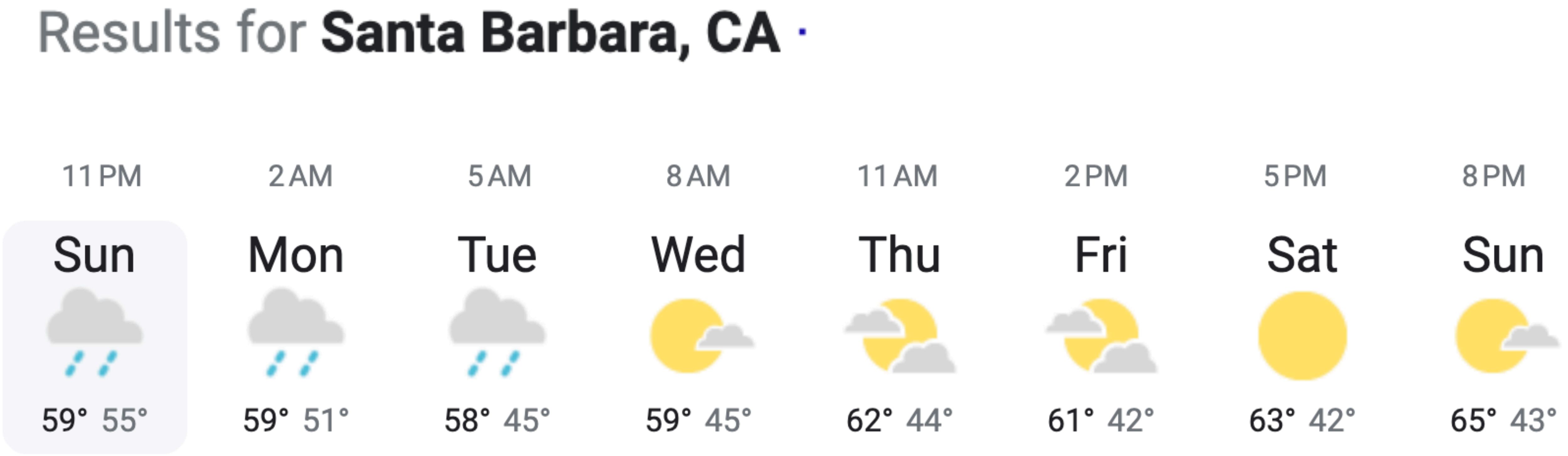
STACKS

Problem Solving with Computers-II

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

int main(){
    cout<<"Hola Facebook\n";
    return 0;

}
```



Results for **Santa Barbara, CA** ·

11 PM	2 AM	5 AM	8 AM	11 AM	2 PM	5 PM	8 PM
Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun
59° 55°	59° 51°	58° 45°	59° 45°	62° 44°	61° 42°	63° 42°	65° 43°

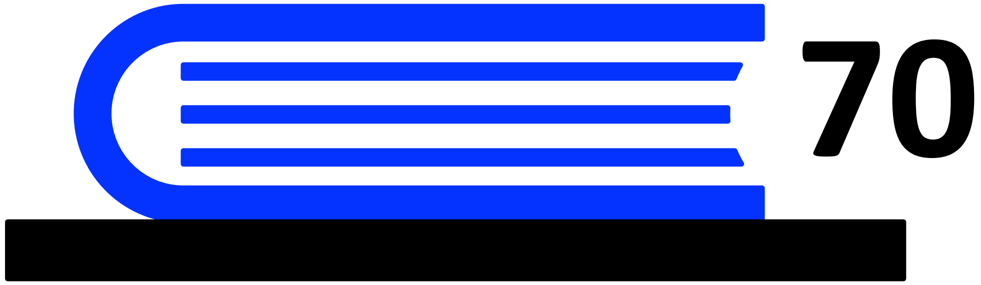
https://leetcode.com/problems/daily-temperatures/

stack<int> s

Empty stack

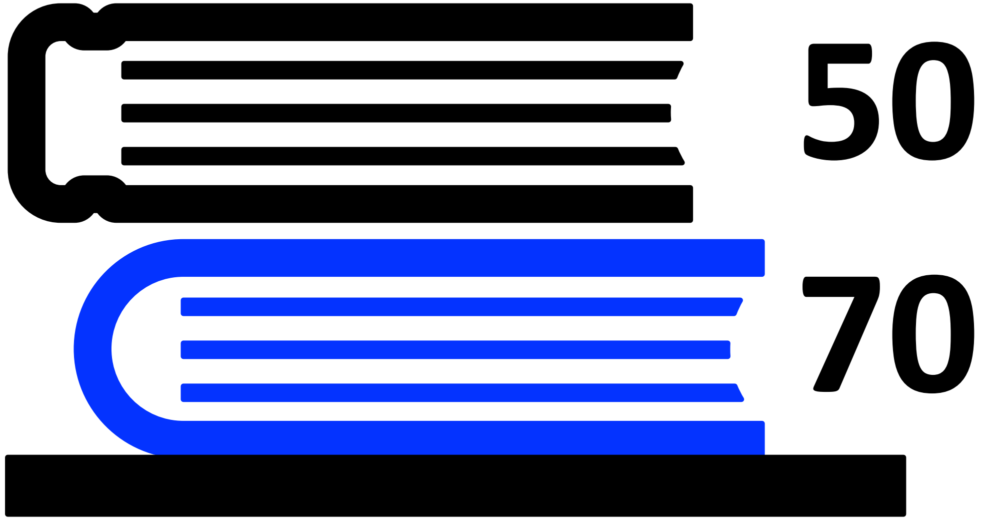
Operations: **push()** **pop()** **top()**

```
stack<int> s
s.push(70)
```



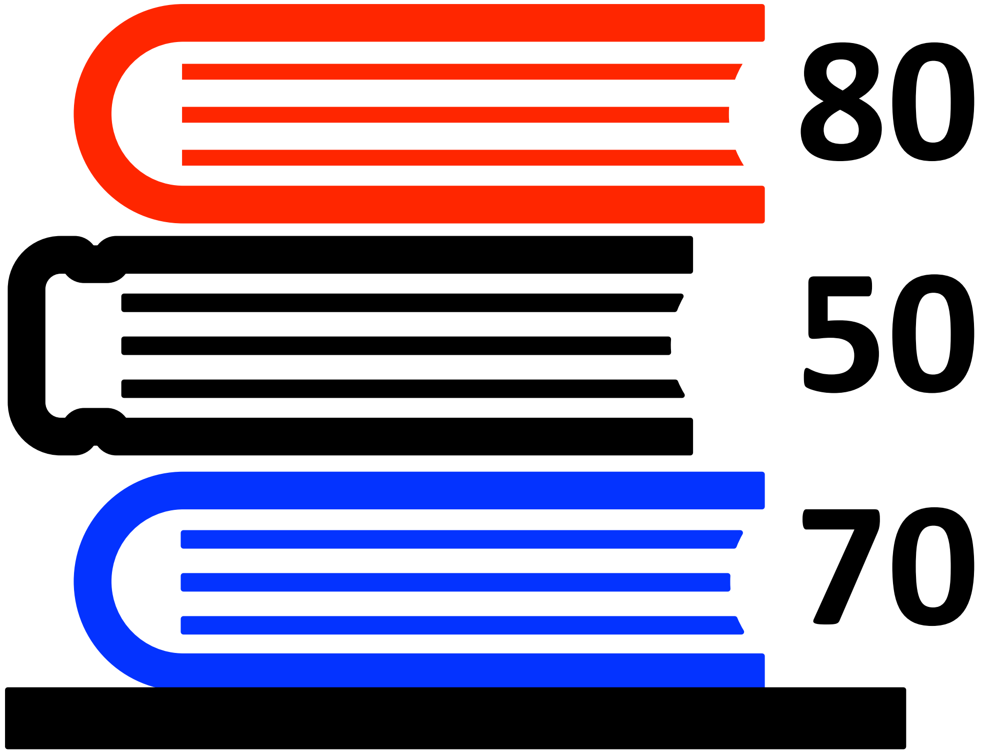
Operations: **push()** pop() top()

```
stack<int> s
s.push(70)
s.push(50)
```



Operations: **push()** pop() top()

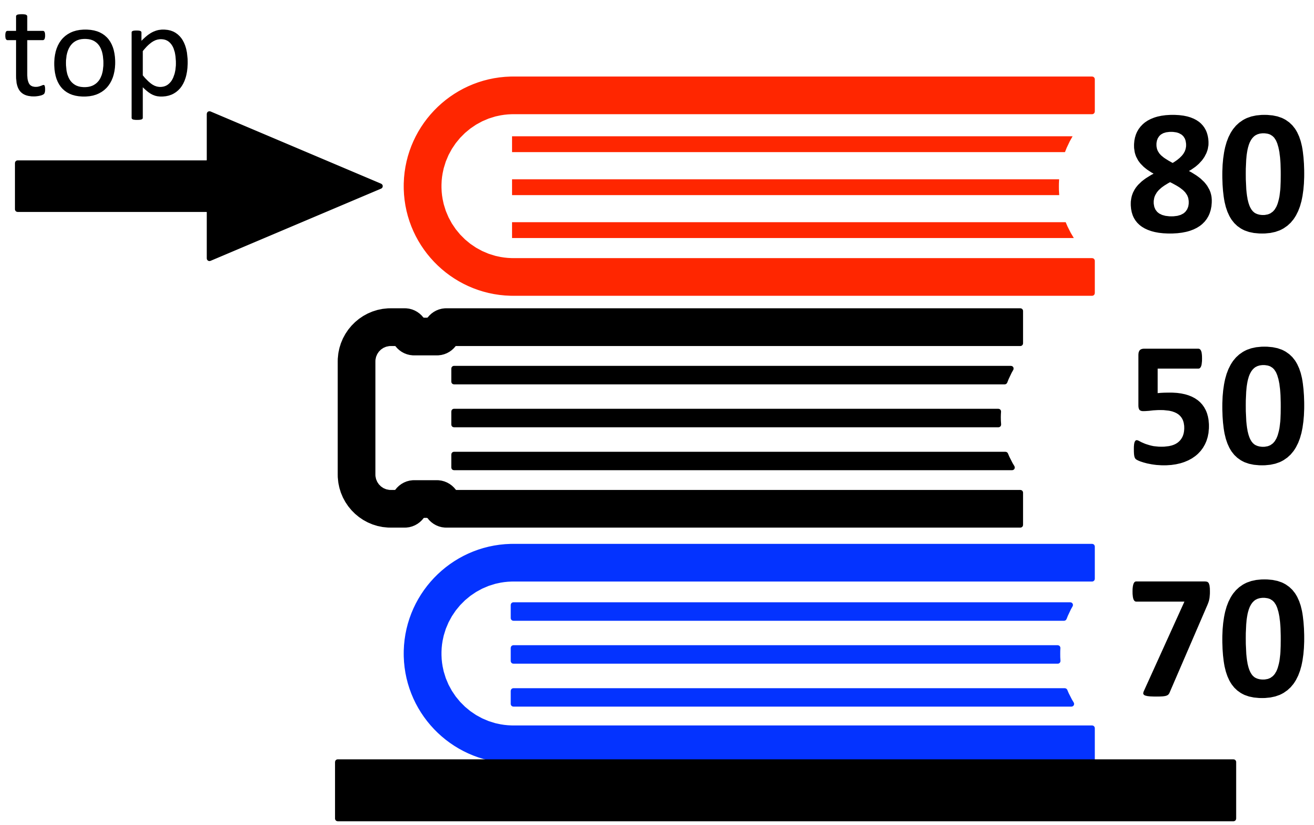
```
stack<int> s
s.push(70)
s.push(50)
s.push(80)
```



Operations: **push()** pop() top()

```
stack<int> s
s.push(70)
s.push(50)
s.push(80)
```

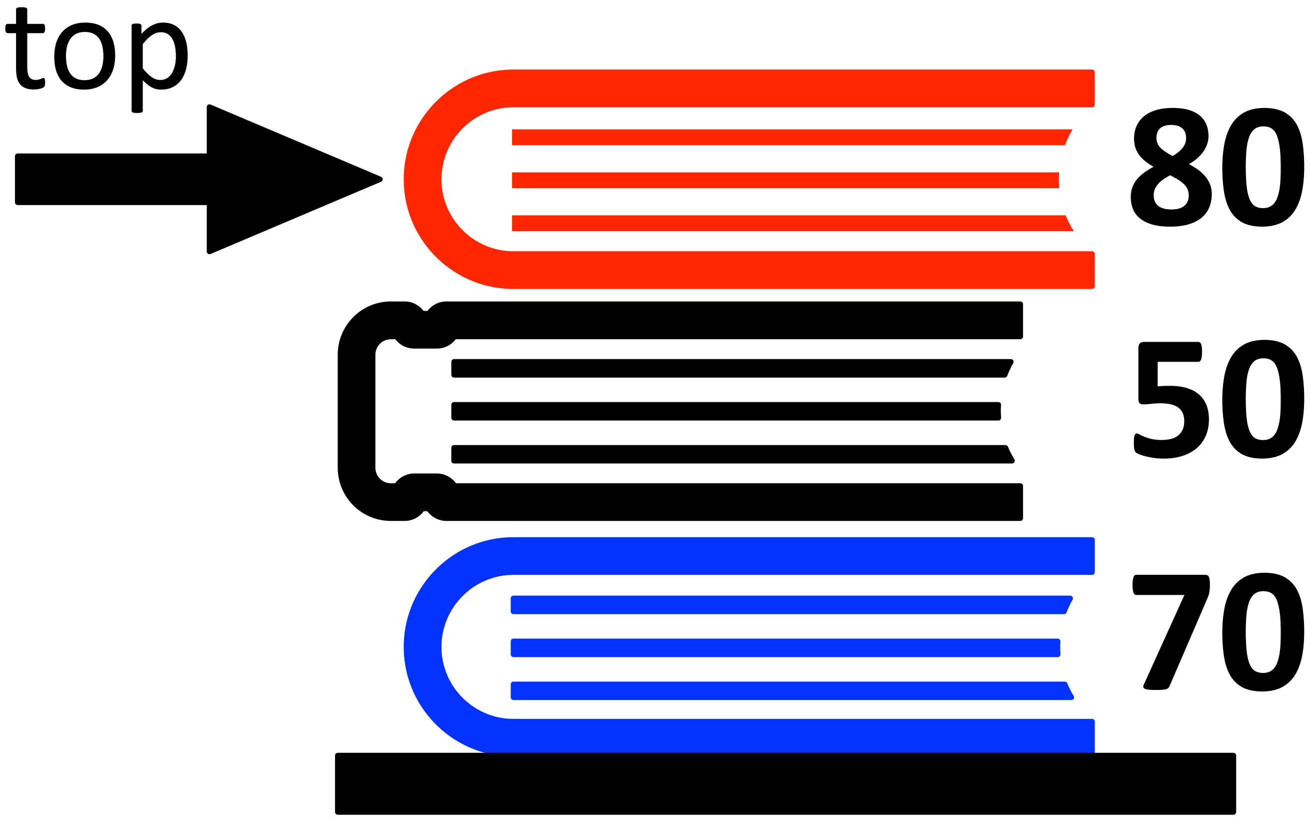
s.top() returns 80



Operations: push() pop() **top()**

```
stack<int> s
s.push(70)
s.push(50)
s.push(80)

s.top()
```



s.pop() removes value that was pushed in *last*

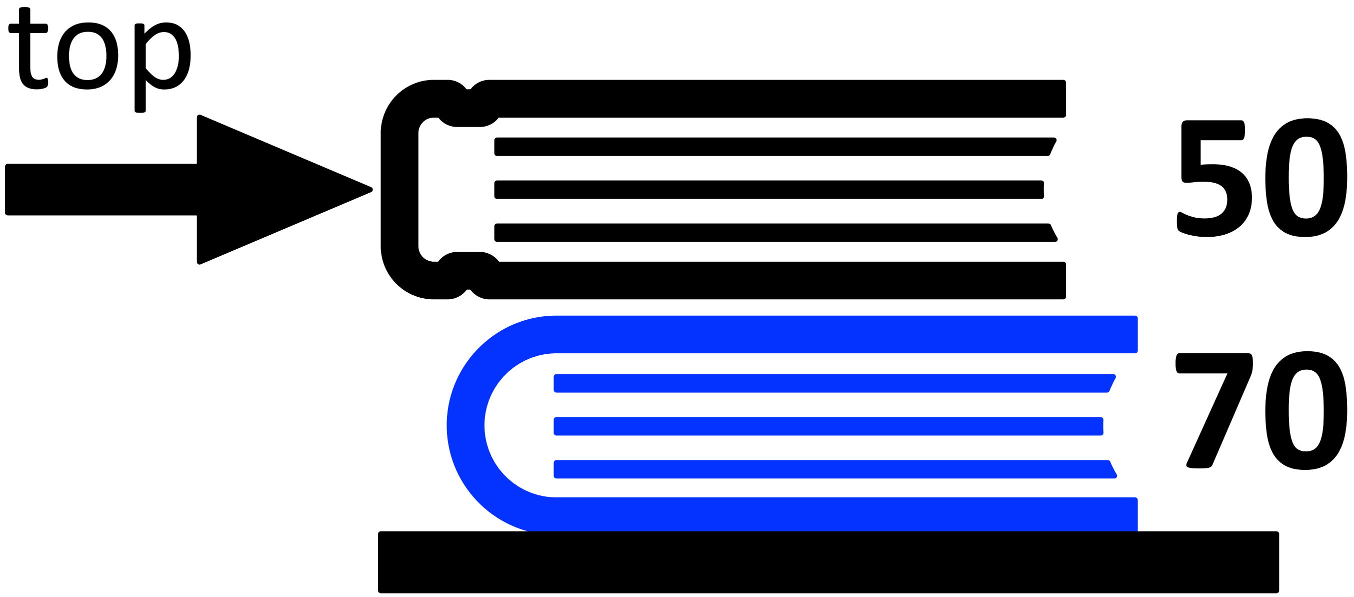
stack<int> s

s.push(70)

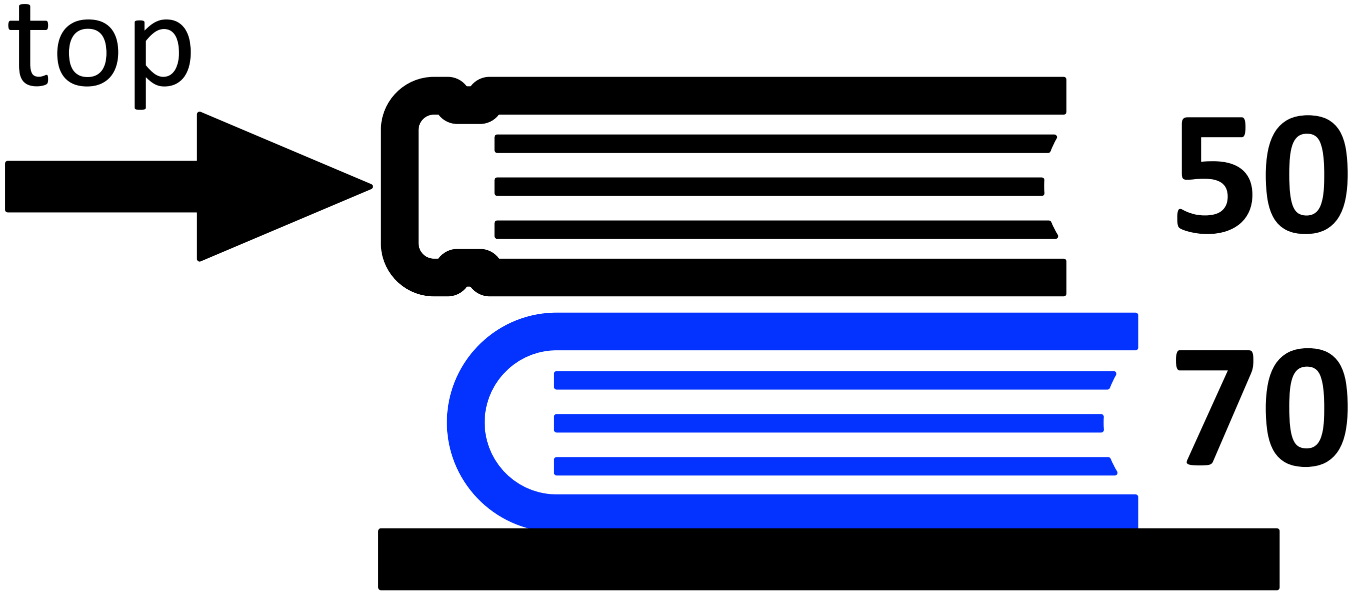
s.push(50)

s.push(80)

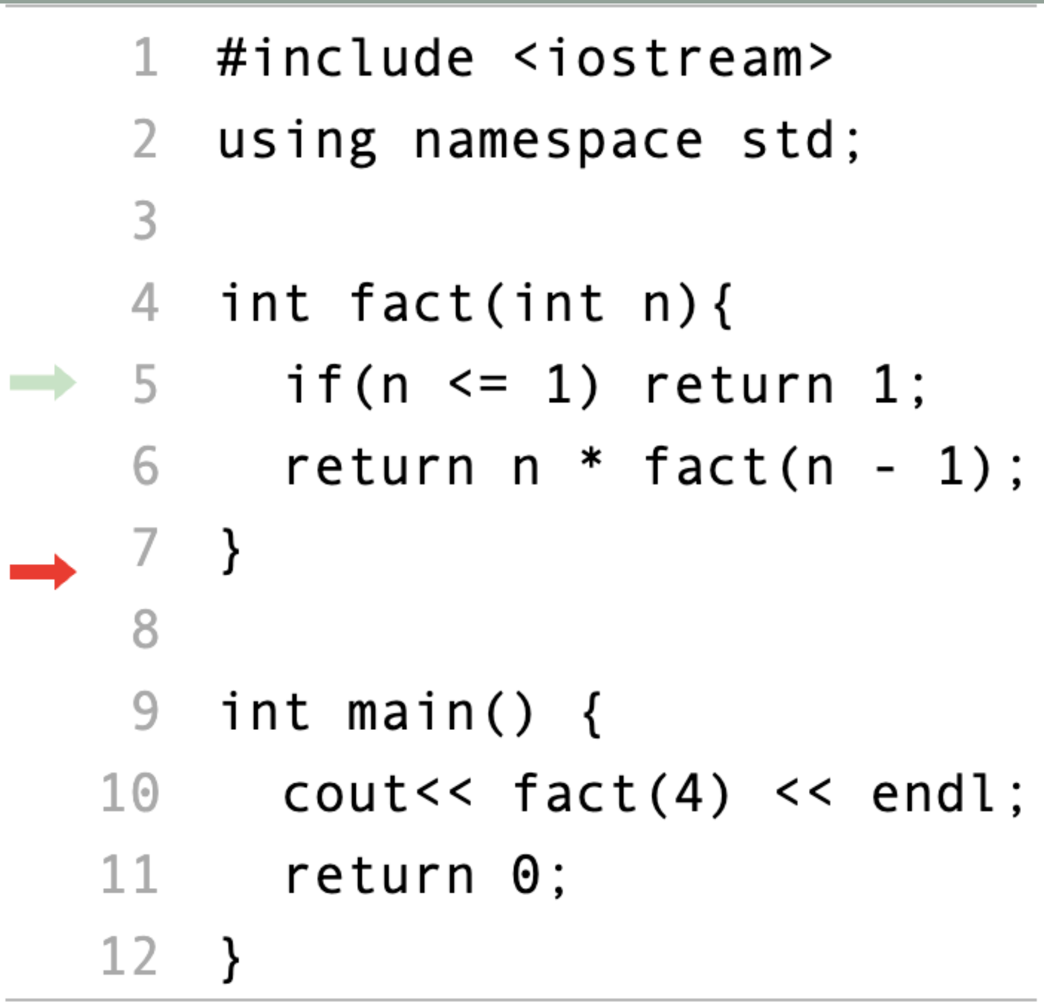
s.top()



s.pop() removes value that was pushed in *last*



The Last value In is the First value Out (LIFO)

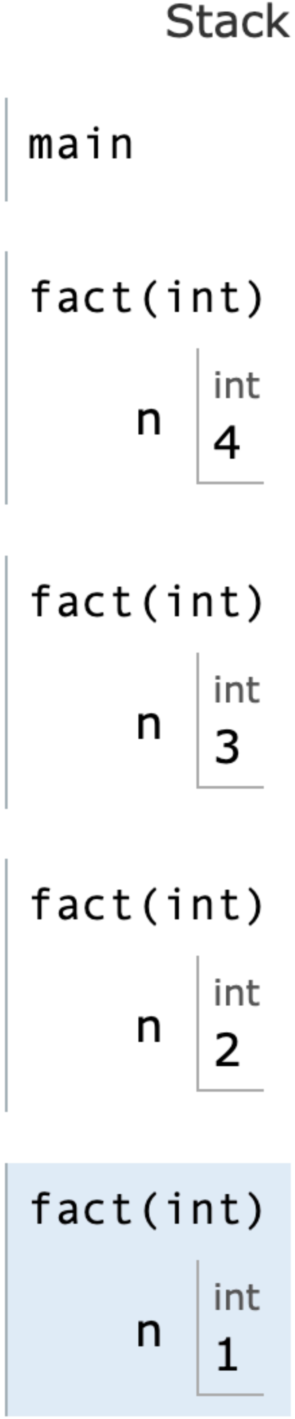


```
#include <iostream>
using namespace std;

int fact(int n){
  if(n <= 1) return 1;
  return n * fact(n - 1);
}

int main() {
  cout<< fact(4) << endl;
  return 0;
}
```

The call stack:



Stack

main

fact(int)
n int 4

fact(int)
n int 3

fact(int)
n int 2

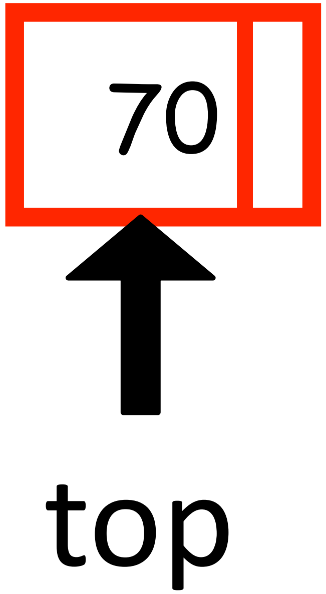
fact(int)
n int 1

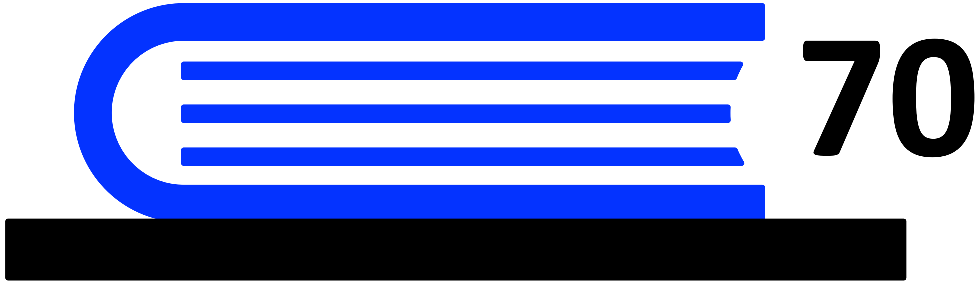
The Last value In is the First value Out (LIFO)

Implement using vector or linked list

Empty stack

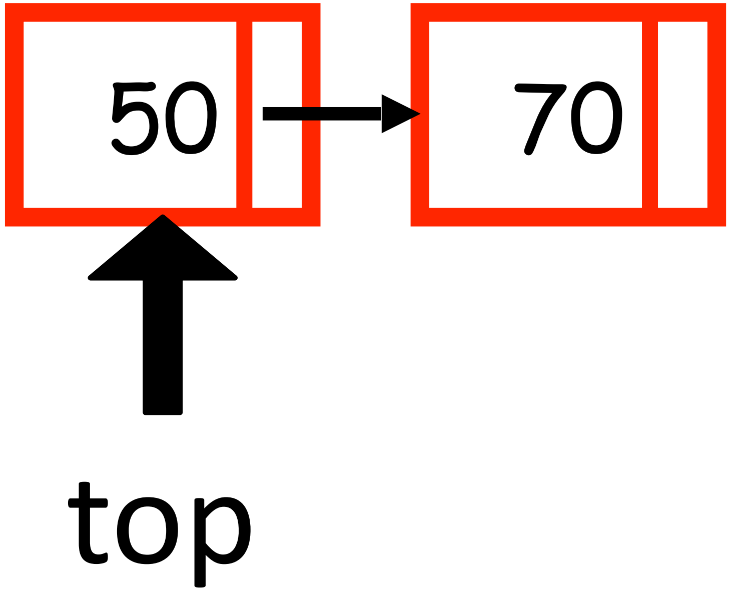
Stack Abstract Data Type





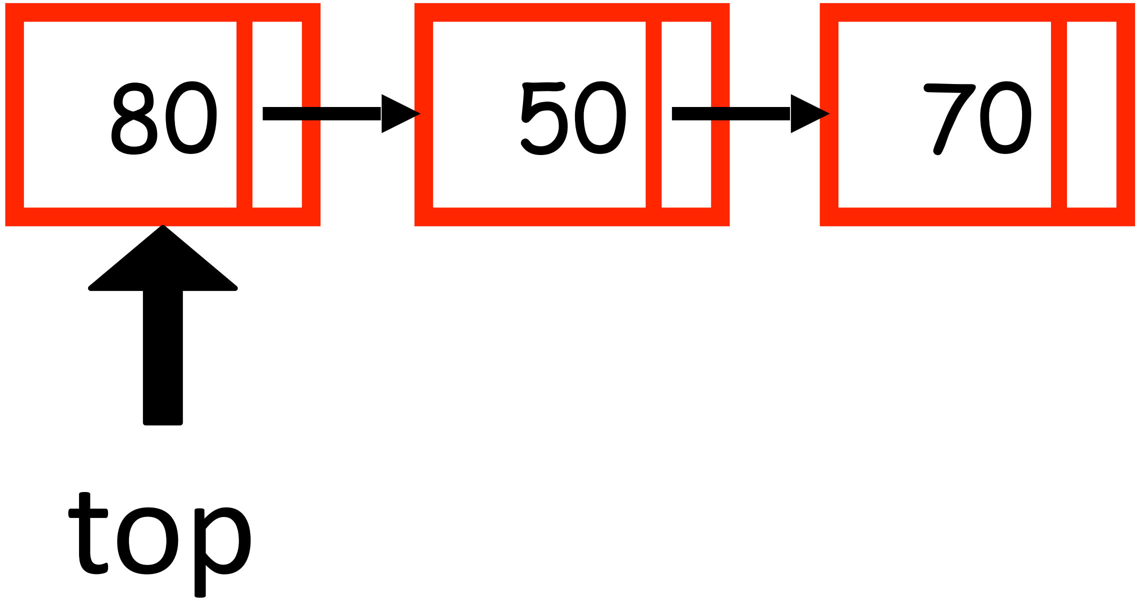
s.push(70)

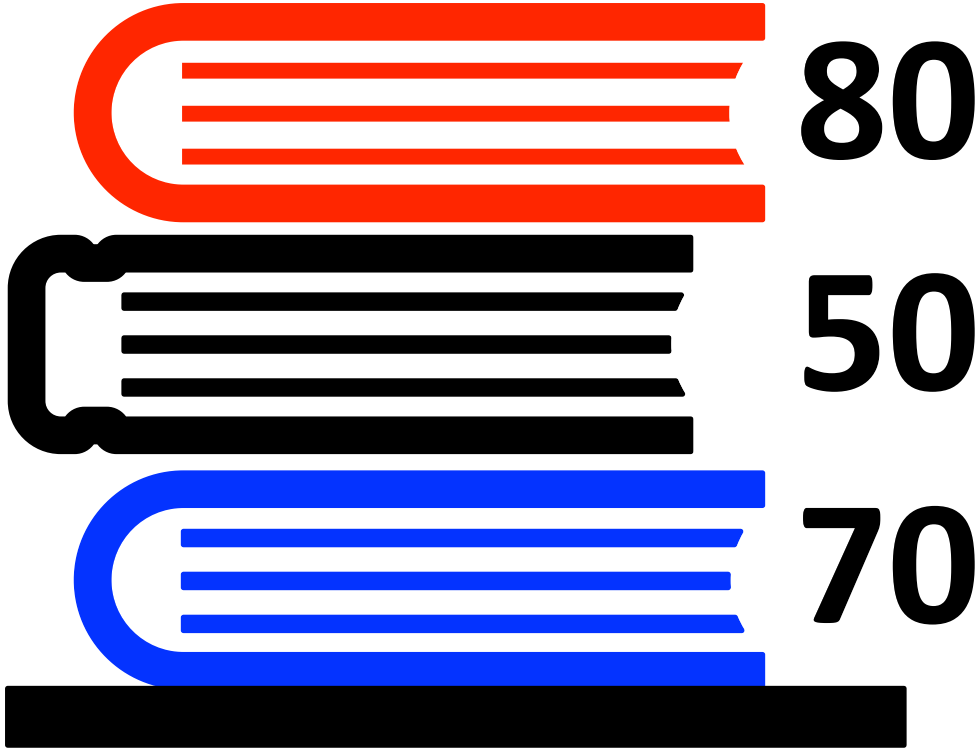
Stack Abstract Data Type



s.push(50)

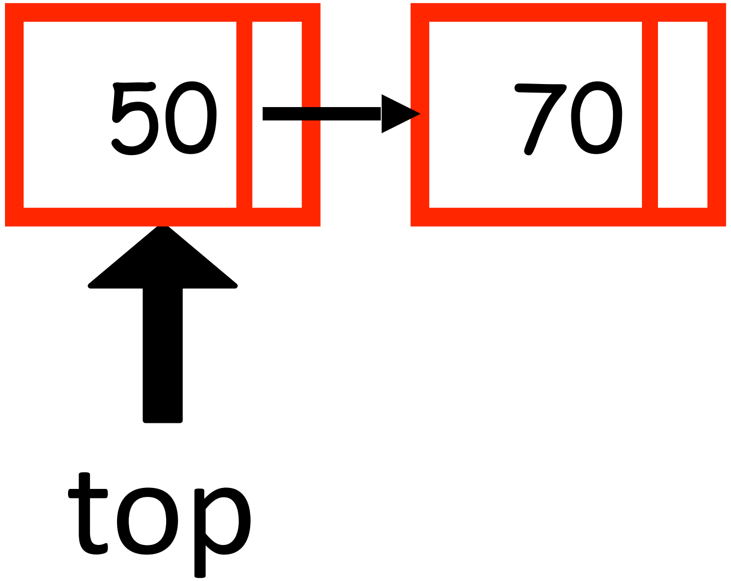
Stack Abstract Data Type





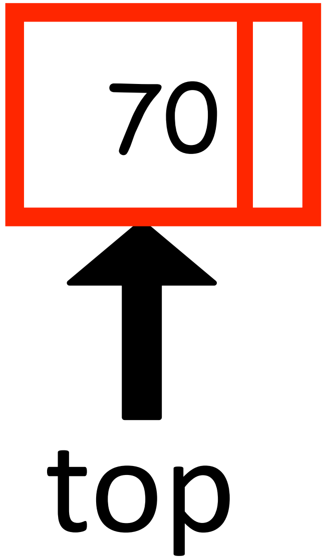
s.push(80)

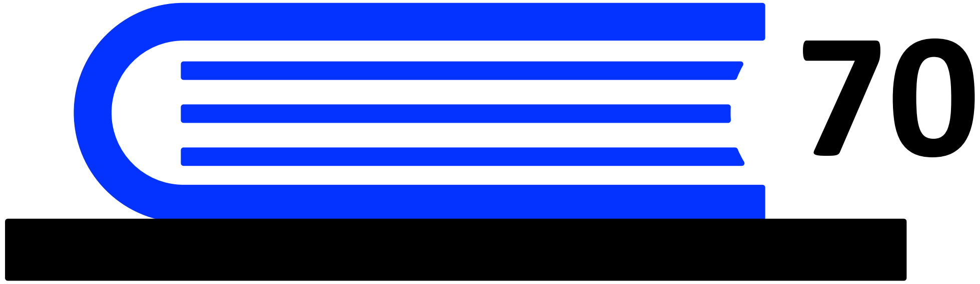
Stack Abstract Data Type



s.pop()

Stack Abstract Data Type





s.pop()

Stack Abstract Data Type

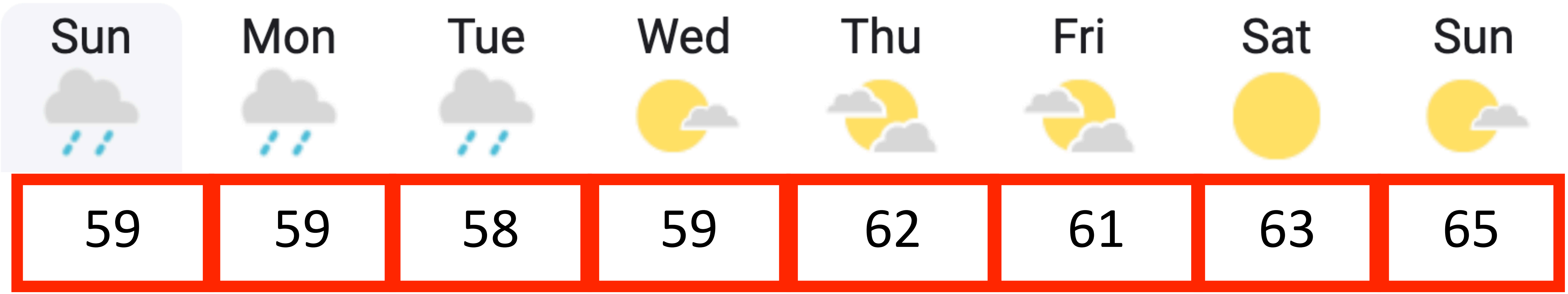
Why implement a stack at all?
After all a stack is a vector or linked list with a **reduced set of operations**

Stack has only three operations: **push()** **pop()** **top()**

Why implement a stack at all?
After all a stack is a vector or linked list with a **reduced set of operations**

A stack is useful for keeping track of history information where computation only depends on the most recent information !!

Stack has only three operations: **push()** **pop()** **top()**



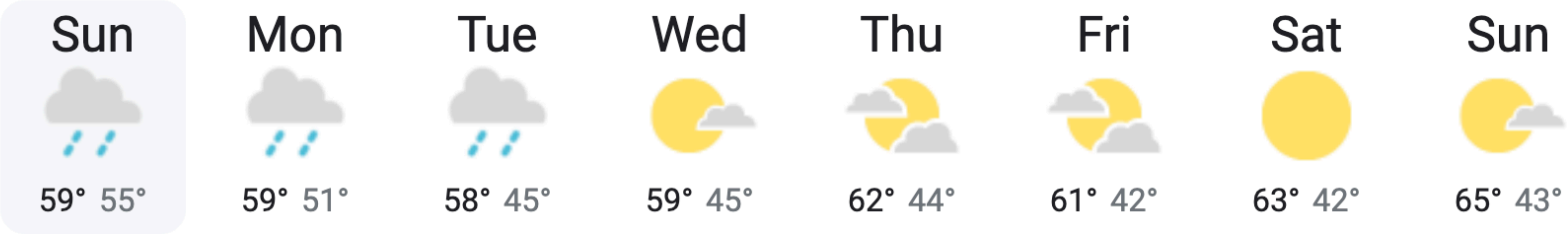
Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun
59	59	58	59	62	61	63	65

Given an array of integers temperatures represents the daily temperatures, return an array answer such that answer[i] is the number of days you have to wait after the ith day to get a warmer temperature. If there is no future day for which this is possible, keep answer[i] == 0 instead.

```
Input: temperatures = [59,59,58,59,62,61,63,65]
Output:  answer      = [4, 3, 1,  2, 2, 1, 1, 0]
```

https://leetcode.com/problems/daily-temperatures/

- **Attempt a different solution to this problem on leetcode**
- **Discuss your solutions with the course staff in office hours**



Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun
59° 55°	59° 51°	58° 45°	59° 45°	62° 44°	61° 42°	63° 42°	65° 43°

A stack is useful for keeping track of history information where computation only depends on the most recent information !!

https://leetcode.com/problems/daily-temperatures/