

## Problem Challenge 3

We'll cover the following ^

- Frequency Stack (hard)
- Try it yourself

### Frequency Stack (hard) #

Design a class that simulates a Stack data structure, implementing the following two operations:

1. `push(int num)` : Pushes the number 'num' on the stack.
2. `pop()` : Returns the most frequent number in the stack. If there is a tie, return the number which was pushed later.


**Example:**


After following push operations: `push(1)`, `push(2)`, `push(3)`, `push(2)`, `push(1)`, `push(2)`, `push(5)`


1. `pop()` should return 2, as it is the most frequent number
2. Next `pop()` should return 1
3. Next `pop()` should return 2


### Try it yourself #

Try solving this question here:

 Java

 Python3

 JS


 C++

```
1 class FrequencyStack:
2
3     def push(self, num):
4         # TODO: Write your code here
5         return 0
6
7     def pop(self):
8         return -1
9
10
11 def main():
12     frequencyStack = FrequencyStack()
13     frequencyStack.push(1)
14     frequencyStack.push(2)
15     frequencyStack.push(3)
16     frequencyStack.push(2)
17     frequencyStack.push(1)
18     frequencyStack.push(2)
19     frequencyStack.push(5)
20     print(frequencyStack.pop())
21     print(frequencyStack.pop())
22     print(frequencyStack.pop())
23
24
25 main()
26
27
28
29
30
31
```

Run

Save

Reset



 Completed

---

 Report an Issue  Ask a Question