

Stack Overflow

Stack is a basic data structure. Where 3 operation can be done-

1. Push: You can push object to the stack
2. Pop: You can pop the object to the stack
3. Top: You can check the value of the top object.

For further details you can get idea here (if you really don't know) :

https://en.wikibooks.org/wiki/Data_Structures/Stacks_and_Queues

Now we have a problem here, there are N stack in front of you. they are numbered from 1 to N. Each of them are initially empty. Now you will have Q operations. Each operation can be one the below 4 types:

1. push i x, Push a value of x to stack numbered i
2. pop i, Pop value from the stack numbered i, if stack is empty discard the operation
3. put i j, Put the **j'th** stack on top of the **i'th** stack. So there will be no element left on the j'th stack.
4. top i, Print the value of the top element of ith stack. If stack is empty print "Empty!"

Check the Sample IO for further understanding...

Input:

Input starts with an integer **T** (≤ 5), denoting the number of test cases.

The first line of a case is a blank line. The next line contains two integers **N** ($1 \leq N \leq 10^4$), **Q** ($1 \leq Q \leq 5 \cdot 10^4$).

The next **Q** lines will contain a operation like above mentioned.

($1 \leq i, j \leq N$), ($1 \leq x \leq 10^5$)

Output

For each test case, print the case number in a single line. Then for each 4th type operation you should print the value or "Empty!" if the stack is empty.

| Input | Output |
|----------|---------|
| 1 | Case 1: |
| 3 18 | 1 |
| push 1 1 | 2 |

| | |
|----------|--------|
| push 2 2 | 4 |
| push 3 3 | 4 |
| push 3 4 | Empty! |
| top 1 | Empty! |
| top 2 | 3 |
| top 3 | 1 |
| put 1 3 | Empty! |
| pop 2 | |
| top 1 | |
| top 2 | |
| top 3 | |
| pop 1 | |
| top 1 | |
| pop 1 | |
| top 1 | |
| pop 1 | |
| top 1 | |

Judge data is huge so use faster IO like scanf/printf

Problem-setter: Ahmad Faiyaz