https://course.acciojob.com/idle?question=b47e7025-826e-48d5-ab1c-345bc0a1687b

- MEDIUM
- Max Score: 40 Points

Implement two Stacks in an Array

Create a data structure twoStacks that represent two stacks. Implementation of twoStacks should use only one array, i.e., both stacks should use the same array for storing elements.

```
Following functions must be supported by twoStacks.

    1. pop1() -> pops an element from first stack and print the popped element. Let this be function 1.
    2. push1(int x) -> pushes x to first stack. Let this be function 2.
    3. pop2() -> pops an element from second stack and print the popped element. Let this be function 3.
    4. push2(int x) -> pushes x to second stack. Let this be function 4.

Implementation of twoStack should be space efficient.
```

Input Format

The first line contains an integer `n` that specifies the number of operations.

In the next `n` lines, there are operations.

`1` signifies calling function pop1().

2 signifies calling function push1(). An integer is followed by 2 which signifies the number to be pushed in stack 1.

3 signifies calling function pop2().

4 signifies calling function push2(). An integer is followed by 4 which signifies the number to be pushed in stack 2.

Output Format

In new lines, print -1 in case of stack underflow and overflow.

Print the integer that is popped otherwise.

Example 1

Input

```
5
1
4 5156
2 989
3
1
```

Output

```
-1
5156
989
```

Explanation

```
Total number of operations is 5.

1 signifies calling pop1(). Since stack 1 is empty, -1 is printed.

4 signifies calling push2(). 5156 is the number pushed into stack 2 as it follows integer
4.

2 signifies calling push1(). 989 is the number pushed into stack 1 as it follows integer
2.
```

`3 signifies calling pop2(). Since stack 2 is not empty, popped element, i.e 5156 is printed.`

`1 signifies calling pop1(). Since stack 1 is not empty, popped element, i.e 989 is printed.`

Example 2

Input

```
6
1
3
4 5156
2 265
2 568
1
```

Output

```
-1
-1
568
```

Constraints

`1 <= n <= 500`

Topic Tags

- **Stacks**
- **Arrays**

My code

```
// n java
import java.util.*;
```

```
class twoStacks {
        int[] arr;
        int size;
        int top1, top2;

        // Constructor
        twoStacks(int n)
        {
                size = n;
                arr = new int[n];
                top1 =-1;
                top2 = n;//it will store reverse
        }

        // Method to push an element x to stack1
        void push1(int x)
        {
         // Your code here
                arr[++top1]=x;
        }

        // Method to push an element
        // x to stack2
        void push2(int x)
        {
       // Your code here
                arr[--top2]=x;
        }
```

```
        // Method to pop an element from first stack
        void pop1()
        {
                // Your code here
                if(top1==-1)
                        System.out.println("-1");
                        else
                 System.out.println(arr[top1--]);
        }

        // Method to pop an element
        // from second stack
        void pop2()
        {
                // Your code here
                if(top2==size)
                        System.out.println("-1");
                        else System.out.println( arr[top2++]);
        }
};
public class Main {

        /* Driver program to test twoStacks class */
        public static void main(String[] args)
        {
                twoStacks ts = new twoStacks(50);
        Scanner sc = new Scanner(System.in);
        int n;
        n = sc.nextInt();
        for(int i =0; i<n; i++){
```

```
            int temp;
            temp = sc.nextInt();
            if(temp == 1) ts.pop1();
            else if(temp==3) ts.pop2();
            else if(temp == 2) {
                int temp2;
                temp2 = sc.nextInt();
                ts.push1(temp2);
            }
            else{
                int temp2;
                temp2 = sc.nextInt();
                ts.push2(temp2);
            }
        }
        }
}
```