
FACE
Prep

Stack Permutation

Stack permutations

You have been given two arrays having an equal number of elements. You have to find whether one array is the valid stack permutation of the other. An array is said to be a valid stack permutation of the other if and only if after applying some push and pop operations onto the sequence of elements in that array, will result in the other array.

Stack permutations

Example:

Input:
arr1[] = [1, 2, 3]
arr2[] = [2, 1, 3]

Output:
YES

```
import java.util.*;

class Main {

	static Boolean check(int ip[], int op[], int n){
		Stack<Integer> s = new Stack<Integer>();
		int j = 0;
		for (int i = 0; i < n; i++) {
			s.push(ip[i]);
			while (!s.isEmpty() && s.peek() == op[j]) {
				s.pop();
				j++;
			}
		}
		if (s.isEmpty())
			return true;
		return false;
	}
```

```java
	public static void main(String args[]){
		Scanner sc=new Scanner (System.in);
		int n=sc.nextInt();
		int input[]=new int[n];
		int output[]=new int[n];
		for(int i=0;i<n;i++)
			input[i]=sc.nextInt();
		for(int i=0;i<n;i++)
			output[i]=sc.nextInt();
		if (check(input, output, n))
			System.out.println("Yes");
		else
			System.out.println("Not Possible");

	}

}
```

THANK YOU

FACE Prep