DSA Lab 2-A

Group 4, 5, 6

19th Jan, 2015

Question 4

Using any of the stack implementations of previous questions, write a program to convert an infix expression to postfix expression.
The input infix expression can contain these characters (of course comma is NOT a valid operator): digit[**0-9**], **+**, *****, **-**, **/**, **(**, **{**, **}**, **)**, **)**

The output postfix string shall not contain any braces or parentheses. It can consist of only digits, **+**, *, **-**, **/**

Also note that the input string will not contain any spaces. However, the output shall have a space after each operator or operand.

Input:
The first line of the input contains T, the number of test cases (i.e. number of infix expressions to be input).
Each of the next T lines contain a string S, corresponding to an infix expression.

Output:
For each test case, output the corresponding postfix expression (exclude all the braces). Don't forget to add space after each character.

Constraints:
1 ≤ T ≤ 100
1 ≤ |S| ≤ 1000

Sample Input:

```
2
3+4*5/6
(300+23)*(43-21)/(84+7)
```

Sample Output:

```
3 4 5 * 6 / +
300 23 + 43 21 - * 84 7 + /
```