CS1020E | Lab 6 | Exercise 1

Infix to Postfix

Objectives

One of the objectives of this exercise is to learn how to use the STL `stack` container adaptor.

Problem Description

An arithmetic expression can be expressed in three notations: *prefix*, *infix*, and *postfix*. The infix notation is the one most of us are familiar with, where each binary operator appears between its two operands. For example, "`2 * ( 3 + 4 )`" is an infix expression. On the other hand, in the postfix notation, each binary operator appears behind its two operands. For example, "`2 3 4 + *`" is the postfix expression equivalent to the example infix expression.

In this exercise, you are to use a stack to convert an input infix expression to the corresponding postfix expression. Refer to the lecture notes for the algorithm. You are required to use the STL `stack`.

Add your code only to the parts of the files indicated. Do not modify any other part of the given code, and do not add new files.

Inputs

The input is a single line that contains an infix expression. The input ends when the end-of-file is reached (which can be entered from the keyboard using CTR-D).

The input infix expression can contain only the following *tokens*:

- positive integer numbers,
- the operators: `+`, `-`, `*` and `/`,
- the left and right parentheses: `(` and `)`.

In the input line, every two tokens are separated by a space. You can assume that every input infix expression is valid.

Outputs

A single line that contains the corresponding postfix expression of the input expression. Each output token is followed by a space (that means the last character of the output line is a space).

Sample Input

```
4 - ( 11 + 7 * 6 ) / 9
```

Sample Output

```
4 11 7 6 * + 9 / -
```

Submission

You need to submit **ALL** your completed skeleton **`*.cpp`** and **`*.h`** files to CodeCrunch (https://codecrunch.comp.nus.edu.sg/) before the specified deadline. We will take only your latest submission.

Late submissions will not be accepted. The submission system in CodeCrunch will automatically close at the deadline.