



Transform the Expression

Link submit: <http://www.spoj.com/problems/ONP/>

Solution:

C++	http://ideone.com/cGW8qV
Java	https://ideone.com/VHwhm1
Python	https://ideone.com/qW50mq

Tóm tắt đề:

Biến đổi các biểu thức, từ biểu thức gốc ban đầu chứa nhiều dấu ngoặc sẽ trở thành dạng đơn giản hơn. Dạng đơn giản hơn là dạng chỉ chứa các phép toán $+$, $-$, $*$, $/$, $^$ và các chữ cái a, b, \dots, z gọi là Ký pháp Ba Lan ngược (Reverse Polish Notation).

Trong một biểu thức có nhiều dấu ngoặc dùng để quy định phép toán nào tính trước phép toán nào tính sau, đối với con người thì rất dễ giải quyết bài toán đó khi chúng ta dùng mắt để phân biệt các biểu thức. Nhưng với máy tính việc tính toán phép nào trước phép nào sau khi có nhiều dấu ngoặc là rất khó khăn, vì thế để máy tính dễ dàng tính toán được người ta thường chuyển nó về một dạng gọi là Reverse Polish Notation. Nên RPN là bài toán kinh điển trong lĩnh vực trình biên dịch.

Input:

Dòng đầu tiên chứa số t ($t \leq 100$) là số lượng biểu thức cần phải biến đổi.

t dòng tiếp theo lần lượt chứa các biểu thức cần được biến đổi, mỗi biểu thức nằm trên một dòng. Chiều dài mỗi biểu thức không vượt quá 400 ký tự.

Output:

Ứng với mỗi biểu thức in mẫu biến đổi biểu thức theo định dạng (Reverse Polish Notation) trên từng dòng.

Ví dụ:

3 (a+(b*c)) ((a+b)*(z+x)) ((a+t)*((b+(a+c))^(c+d)))	abc*+ ab+zx+* at+bac++cd+^*
--	-----------------------------------

Hướng dẫn giải:

Cho các ký tự cần chuẩn hóa vào chuỗi. Duyệt chuỗi từ ký tự đầu tiên đến ký tự cuối cùng:

Nếu gặp dấu ngoặc "(" thì bỏ qua, gặp ký tự thì in ký tự ra, gặp các phép toán thì bỏ vào stack. Nếu gặp dấu đóng ")" thì in phép toán ở đầu stack ra và bỏ nó ra khỏi stack. Lần lượt làm như vậy cho đến khi nào đi hết chuỗi thì dừng. Bạn sẽ có được kết quả đề bài yêu cầu.

Độ phức tạp: $O(T * N)$ với T là số lượng test case, N là độ dài chuỗi của từng test case.

Big-O Coding