URI Online Judge | 1726

Friends

By Ralf Engels Germany

Timelimit: 1

You want to plan a big birthday' party with your friends. On planning you notice that you have to do a lot of operations with sets of friends. There is one group which consist of Arthur, Biene and Clemens. Then there is a group of friends you know from snowboarding which consists of Daniel, Ernst, Frida and Gustav. If you want to invite them both, the resulting party group consists of g1 + g2 (the result is the union of both groups). Then you can compute the intersection of the two groups g1 * g2, which consists of the empty set. Maybe you want to invite a group g1, but excluding all members of an other group g2, which is written as g1 - g2. Intersection (*) has precedence over union (+) and set difference (-). All operations are left associative, which means that in A op_1 B op_2 C you first have to evaluate A op_1 B (provided op_1 and op_2 have equal precedence).

Input

The input consists of one or more lines. Each line contains one expression that you have to evaluate. Expressions are syntactically correct and only consist of the characters:

- '**{**' and '**}**'
- the elements '**A**' to '**Z**' meaning friend Arthur to Zora.
- the operations '**+**', '**-**' and '*****'
- '**(**' and '**)**' for grouping operations
- the newline character '**\n**' marking the end of an expression.

A line is never longer than 255 characters.

Output

Output the resulting set in curly braces '**{**' and '**}**', each on a line of its own. Print elements of sets sorted alphabetically.

Sample Input	Sample Output
{ABC} {ABC}+{DEFG}+{Z}+{} {ABE}*{ABCD} {ABCD}-{CZ} {ABC}+{CDE}*{CEZ} ({ABC}+{CDE})*{CEZ}	{ABC} {ABCDEFGZ} {AB} {ABD} {ABCE} {CE}

University of Ulm local Contest 1999/2000