Polish Notation

`polish.c`, `polish.C`, `polish.java`

Prefix notation is a non-conventional notation for writing arithmetic expressions. The standard way of writing arithmetic expressions, also known as infix notation, positions a binary operator between the operands, e.g., $3 + 4$, while in prefix notation the operator is positioned before the operands, e.g., `+ 3 4`. Similarly, the prefix notation for $5 - 2$ is `- 5 2`.

A nice property of prefix expressions with binary operators is that parentheses are not required since there is no ambiguity about the order of operations. For example, the prefix representation of $5 - (4 - 2)$ is `- 5 - 4 2`, while the prefix representation of $(5 - 4) - 2$ is `- - 5 4 2`. The prefix notation is also known as *Polish notation*, due to Jan Łukasiewicz, a Polish logician, who invented it around 1920.

Similarly, in *postfix* notation, or *reverse Polish notation*, the operator is positioned after the operands. For example, postfix representation of the infix expression $(5 - 4) - 2$ is `5 4 - 2 -`.

Your task is to write a program that translates a prefix arithmetic expression into a postfix arithmetic expression.

Input Specification

Each line contains an arithmetic prefix expression. The operators are "`+`", "`-`", "`*`", and "`/`", and numbers are all single-digit decimal numbers. The operators and numbers are separated by exactly one space with no leading spaces on the line. The end of input is marked by "`0`" on a single line. You can assume that each input line contains a valid prefix expression with less than 200 operators.

Output Specification

Translate each expression into postfix notation and produce it on a separate line. The numbers and operators are separated by at least one space. The final `0` is not translated.

Sample Input

```
- - 5 4 2
7
+ 1 2
- 2 2
* 2 - 2 1
- / 3 + 2 1 9
0
```

Output for Sample Input

```
5 4 - 2 -
7
1 2 +
2 2 -
2 2 1 - *
3 2 1 + / 9 -
```