TC1018: Estructura de Datos C++
PROG08 – Postfix Expresion Evaluation
Ing. Luis Humberto González G
Forma de Trabajo: *Individual*.

This past week, in his Data Structure course, Carlos learned how to evaluate simple arithmetical expressions written in postfix notation (aka, Reverse Polish Notation). For instance, 7 4 – 5 * written in infix notation is (7 – 4) * 5. The "trick" is to use a data structure called stack, where elements inserted last are extracted first (Last In First Out). Concretely, we use the following algorithm:

Process the sequence of elements from left to right:

- For each number, push it onto the stack.
- For each operator, pop two elements from the stack, say X and Y (where X is the first element popped from the stack), perform the operation Y op X; push the result onto the stack.

After processing all elements, the stack should contain one single element: the value of the postfix expression. For instance, the postfix expression 7 4 - 5 * is evaluated by performing the following actions:

- Reads number 7, push 7 onto the stack.
- Reads number 4, push 4 onto the stack.
- Reads operator -, pop two elements from stack: X = 4, Y = 7, perform operation Y - X = 7 – 4 = 3, push answer 3 onto stack.
- Reads number 5, push 5 onto the stack.
- Reads operator *, pop two elements from stack: X = 5, Y = 3, perform operation Y * X = 3 * 5 = 15, push answer 15 onto stack.
- The value of the expression, 15, sits alone in the stack.

Being curious, Carlos wondered what would be the result if the stack is replaced by other data structures, such as a queue or a minimum priority queue.

Notes:

- A queue is a data structure where elements inserted first are extracted first (First In First Out).
- A priority queue is a data structure where elements with highest priority are extracted first.
- A minimum priority queue is a priority queue where the lowest-valued elements have the highest priority.

Input

There are several cases; each test case consists of one line containing the postfix expression to be evaluated. This expression is guaranteed to be a valid expression written in postfix consisting of a sequence of non-negative integers and operators +, -, *, all separated by a single space. The length of the each line does not exceed 100 characters. All input, intermediate, and final values will fit in a 32-bit signed integer.
The input ending with a line with **#**.

Output

For each test case, output a line with three spaced-separated integers: the values of the expression when evaluated with a stack, queue, and minimum priority queue, respectively.

Sample Input

```
7 4 - 5 *
3 2 4 * +
3 4 2 - *
42 7 - 3 * 2 4 + *
5 3 5 - * 11 2 - 1 * -
#
```

Sample Output

```
15 -15 15
11 10 10
6 2 4
630 -412 630
-19 41 3
```