Problem C - A BLP solver

Description

A Binary Linear Programming (BLP) problem is defined as follows:

maximize	$c^T x$
subject to	$Ax \le b$
and	$x \in \{0,1\}^n$

where x and c are vectors of size n, b is a vector of size m and A is a matrix of size m×n. The goal in the BLP problem is to maximize $c^T x$ by assigning values (0 or 1) to each element of x such that the constraints defined in $Ax \le b$ are never violated. In our model, it is possible to have "minimize" instead of "maximize" and constraints of equality (=) and inequality (≤ or ≥).

Input

Each test case starts with the word "minimize" or "maximize" in the first line. Then, $c^T x$ is given in the second line as a linear sum expression of the following type

$c_1 x_1 + c_2 x_2 + \dots c_n x_n$

Consider that the indices are not ordered in the input data; see example of input data below. It may even be possible that some indices do not show up in the expression when $c_i = 0$; they may, however, be present in the constraints. Also, note that c_i may be positive or negative; if negative, the c_i is always preceded by a negative sign, otherwise by a positive sign, except in the first term of the expression. In addition, if the absolute value of c_i is 1, then this value is not shown in the expression.

In the following line there is string "st", which indicates that the following lines define the constraints. There may exist more than one constraint. In that case, each constraint is defined in a single row.

Each constraint is defined as a linear sum expression as follows:

$a_1 x_1 + a_2 x_2 + \dots + a_n x_n \le b$

or

$a_1 x_1 + a_2 x_2 + \dots + a_n x_n = b$

or

$a_1x_1 + a_2x_2 + \ldots + a_nx_n \geq b$

The notes above for c^Tx also apply to each of the three type of constraints.

The input data terminates with the string "BINARY".

Output

For each test case, print the optimal value of c^Tx or report "INFEASIBLE" if it is not possible to find a solution that satisfies all the constraints.

Constraints

- $n \leq 45$

Example

Example input:

```
maximize
-62x1+35x3-97x2
st
25x9-57x3+76x4=4
-45x3-11x2+23x1<=57
BINARY

minimize
84x3-64x6-98x4-70x2+77x9
st
96x9+100x4+26x2-100x7-8x6<=67
56x1-9x6+34x5+75x2-76x3>=25
x6<=0
BINARY
```

Example output:

```
INFEASIBLE

-168
```
