Problem C. Comparator

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	2048 megabytes

Many programming languages let you define custom comparators to sort user-defined objects.

IFFY is a programming language where the only programs are functions intended to be comparators. These functions operate on bitstrings, hereafter called "words". Words are 1-indexed starting with the leftmost character.

To write a function on two words in IFFY, the programmer writes a series of if-statements. Each if statement chooses a bit from the first word, a bit from the second word, and evaluates some expression on those two bits. If the expression evaluates to 1 on those chosen bits, then the function returns immediately with a specified return value. Otherwise, execution falls through to the next if statement, not returning anything.

The syntax of each if statement is *a b expr r*, where the integer a specifies a bit from the first word, integer b specifies a bit from the second word, and *expr* is a boolean expression using variables x and/or y, and r is the value (0 or 1) returned from the function if the expression evaluates to 1. The variable x refers to the specified bit in the first word, and the variable y refers to the specified bit in the second word. For example, consider the if statement '`2 3 x|y 0`'. This expression means: if the second bit of the first word or the third bit of the second word is set, then return 0, otherwise continue with the next statement.

The grammar for valid expressions is as follows:

- `x`, `y`, `0`, and `1` are valid expressions.
- If `E` is a valid expression, then `(E)` is a valid expression.
- If `E` is a valid expression, then `!E` is a valid expression.
- If `E1` and `E2` are valid expressions, then all strings of the form `E1`$\oplus$`E2` are valid expressions, where $\oplus$ is one of `=` (equals), `&` (and), `|` (or), or `^` (xor).

Parentheses take highest precedence, followed by the unary `!`, followed by the binary `=`, `&`, `|`, and `^`, in that order. All binary operators are left-associative. There are no spaces in expressions.

Here are the truth tables for each operator:

x	`!`x
0	1
1	0

Рис. 1: The truth table for the sole unary operator.

x	y	$x = y$
0	0	1
0	1	0
1	0	0
1	1	1

x	y	x&y
0	0	0
0	1	0
1	0	0
1	1	1

x	y	x|y
0	0	0
0	1	1
1	0	1
1	1	1

x	y	x^y
0	0	0
0	1	1
1	0	1
1	1	0

Рис. 2: Truth tables for the binary operators.

In order for a function $f(x, y)$ to operate properly as a comparator, it must satisfy certain properties. Informally, for $f(x, y)$ to be a comparator, it should impose some ordering $<$, where $f(x, y)$ returns 1 if and only if $x < y$. For example, sample 1 is a valid comparator to sort bitstrings of length 2. More

formally, three of the properties that comparators must satisfy are the reflexive, symmetric, and transitive properties, as follows:

1. **Reflexive:** For all x, $f(x, x)$ should return 0.
2. **Symmetric:** For all x, y, if $f(x, y)$ returns 1, then $f(y, x)$ must return 0.
3. **Transitive:** For all x, y, z, if both $f(x, y)$ and $f(y, z)$ return 1, then so must $f(x, z)$.

Given a function in the IFFY language, determine how well it satisfies these three properties by counting how often they are violated. First, among all words of a given length, count the number of words for which the reflexive property fails. Next, count the number of pairs of words for which the symmetric property fails. Finally, count the number of triples of words for which the transitive property fails.

Input

The first line of input contains two integers n ($0 \le n \le 2 \cdot 10^5$) and k ($1 \le k \le 10$), where n is the number of lines in the function and k is the number of bits in the values being compared.

Each of the next n lines contains four tokens of the form a b *expr* r, where a and b ($1 \le a, b \le k$) are the indices of the bits to consider in x and y, *expr* is a valid expression, and r ($0 \le r \le 1$) is the return value. The final line gives the returned value if no if statement is triggered. The total length of all expressions is at most 10^6.

Output

Output a single line with three space separated integers. The first integer is the number of words for which the reflexive property is violated, the second is the number of pairs of words for which the symmetric property is violated, and the third is the number of triples of words for which the transitive property is violated.

Examples

standard input	standard output
3 2 1 1 (x=0)&(y=1) 1 1 1 (x=1)&(y=(x^x)) 0 2 2 (x=1)|(y=0) 0 1	0 0 0
4 3 2 1 x=0&(y=1) 1 1 2 !x^!!y 0 2 3 ((x|1)=y)&1&1 1 3 1 !x&!x&!x 0 1	3 25 52