

NOI '14 P1 - Getting-Up Syndrome

Time limit: 0.6s **Memory limit:** 512M

National Olympiad in Informatics, China, 2014

In the 21st century, many people have contracted an odd disorder – getting-up syndrome. Symptoms include having great difficulty getting out of bed in the morning and feeling out of sorts even after getting up. As a generally high-spirited teenager, ATM has been struggling with getting-up syndrome. Through extensive research, he has found the cause of the disorder. On the vast seabeds of the Pacific Ocean, there lives a giant dragon by the name of DRD who holds the essence of sleep itself and can extend everyone's sleeping time at his will. So with DRD's every movement, getting-up syndrome intensifies for everyone, growing at a frightening rate to affect more and more people in the world. To put an end to this terrible malady, ATM has decided to travel to the seabed of the Pacific and slay this evil dragon once and for all.

After untold hardships, ATM has finally reached DRD's resting place. He now braces for the arduous battle that lies ahead. DRD has a very special tactic. His line of defense involves transforming the attack power of the opponent using a series of calculations to minimize the damage done to himself. Roughly speaking, DRD's line of defense consists of n defense gates. Each gate contains an operator op and a parameter t . The operator is guaranteed to be one of `OR`, `XOR`, or `AND`, while the parameter is guaranteed to be a nonnegative integer. If one's power before going through a defense gate is x , then the power after going through it is $x \text{ op } t$. Finally, the damage done to DRD is equal to his opponent's initial striking power x **after it has been through all n defense gates**.

Since ATM's skill is limited, the initial attack power of his strike can only be an integer between 0 and m (he may pick any number of $0, 1, \dots, m$ to be his initial attack power), but the final attack power after it has been through the gates is not bounded by m . To conserve energy, he will have to pick the optimal initial power with which to attack to maximize the damage done to DRD. Please help him calculate how much damage he can do to DRD with one strike.

Input Specification

The first line contains two integers n and m , indicating that DRD uses n defense gates and that ATM can pick any integer between 0 and m as his initial striking power.

The next n lines each describes a single defense gate. Each line consists of a single string of characters representing op , followed by a space, followed by a nonnegative integer t representing the parameter number of that gate.

Output Specification

Output one line consisting of a single integer, representing the maximum possible damage that ATM could do to DRD in one strike.

Sample Input

3 10
AND 5
OR 6
XOR 7

Sample Output

1

Explanation

ATM can choose any one of the numbers $0, 1, \dots, 10$ as one of his initial striking power. If he chooses 4, then his final damage is calculated as follows:

4 AND 5 = 4
4 OR 6 = 6
6 XOR 7 = 1

Similarly, we can calculate to find out that initial striking powers of 1, 3, 5, 7, and 9 will result in a damage of 0 at the end. Initial striking powers of 0, 2, 4, 6, 8, and 10 will all result in final damages of 1. Thus, ATM's maximum possible damage with one strike is 1.

Constraints

Test Case	n, m	Guarantees	Notes
-----------	--------	------------	-------

1	$2 \leq n \leq 100, m = 0$	$0 \leq t \leq 10^9$ op is one of <input type="button" value="OR"/> , <input type="button" value="XOR"/> , <input type="button" value="AND"/>	
2	$2 \leq n \leq 1\ 000$		
3	$1 \leq m \leq 1\ 000$		
4	$2 \leq n, m \leq 10^5$		One gate with <input type="button" value="AND 0"/> will exist.
5			All gates will have the same operator.
6			
7	$2 \leq n \leq 10^5$ $2 \leq m \leq 10^9$		All gates will have the same operator.
8			
9			
10			

Hint

For this problem, competitors will have to first convert operands into binary. If two operands have different number of digits, then the shorter operand must be padded with leading 0's before carrying out the operation.

is the bitwise OR operation. For two binary operands of equal length, pairs of corresponding digits will yield a result of 1 if either digit is 1, otherwise a result 0.

is the bitwise XOR operation. For two binary operands of equal length, pairs of corresponding digits will yield a result of 1 if the digits are different (distinct), otherwise a result 0.

is the bitwise AND operation. For two binary operands of equal length, pairs of corresponding digits will yield a result of 1 if both digits are 1, otherwise a result 0.

For example, we can perform the , , and operations on the decimal values 5 and 3, resulting in the following:

```

0101 (5 in base-10)
OR 0011 (3 in base-10)
= 0111 (7 in base-10)

```

```

0101 (5 in base-10)
XOR 0011 (3 in base-10)
= 0110 (6 in base-10)

```

```

0101 (5 in base-10)
AND 0011 (3 in base-10)
= 0001 (1 in base-10)

```

Problem translated to English by [Alex](#).