

## 起床困难综合症

### 【问题描述】

21 世纪, 许多人得了一种奇怪的病: 起床困难综合症, 其临床表现为: 起床难, 起床后精神不佳。作为一名青春阳光好少年, atm 一直坚持与起床困难综合症作斗争。通过研究相关文献, 他找到了该病的发病原因: 在深邃的太平洋海底中, 出现了一条名为 drd 的巨龙, 它掌握着睡眠之精髓, 能随意延长大家的睡眠时间。正是由于 drd 的活动, 起床困难综合症愈演愈烈, 以惊人的速度在世界上传播。为了彻底消灭这种病, atm 决定前往海底, 消灭这条恶龙。

历经千辛万苦, atm 终于来到了 drd 所在的地方, 准备与其展开艰苦卓绝的战斗。drd 有着十分特殊的技能, 他的防御战线能够使用一定的运算来改变他受到的伤害。具体说来, drd 的防御战线由  $n$  扇防御门组成。每扇防御门包括一个运算  $op$  和一个参数  $t$ , 其中运算一定是  $OR, XOR, AND$  中的一种, 参数则一定为非负整数。如果还未通过防御门时攻击力为  $x$ , 则其通过这扇防御门后攻击力将变为  $x \text{ op } t$ 。最终 drd 受到的伤害为对方初始攻击力  $x$  依次经过所有  $n$  扇防御门后转变得到的攻击力。

由于 atm 水平有限, 他的初始攻击力只能为  $0$  到  $m$  之间的一个整数 (即他的初始攻击力只能在  $0, 1, \dots, m$  中任选, 但在通过防御门之后的攻击力不受  $m$  的限制)。为了节省体力, 他希望通过选择合适的初始攻击力使得他的攻击能让 drd 受到最大的伤害, 请你帮他计算一下, 他的一次攻击最多能使 drd 受到多少伤害。

### 【输入格式】

从文件 *sleep.in* 中读入数据。

输入文件的第 1 行包含 2 个整数, 依次为  $n, m$ , 表示 drd 有  $n$  扇防御门, atm 的初始攻击力为  $0$  到  $m$  之间的整数。

接下来  $n$  行, 依次表示每一扇防御门。每行包括一个字符串  $op$  和一个非负整数  $t$ , 两者由一个空格隔开, 且  $op$  在前,  $t$  在后,  $op$  表示该防御门所对应的操作,  $t$  表示对应的参数。

### 【输出格式】

输出到文件 *sleep.out* 中。

输出一行一个整数, 表示 atm 的一次攻击最多使 drd 受到多少伤害。

### 【样例输入 1】

```
3 10
AND 5
OR 6
XOR 7
```

### 【样例输出 1】

```
1
```

## 【样例说明 1】

atm 可以选择的初始攻击力为  $0, 1, \dots, 10$ 。

假设初始攻击力为 4，最终攻击力经过了如下计算

$$4 \text{ AND } 5 = 4$$

$$4 \text{ OR } 6 = 6$$

$$6 \text{ XOR } 7 = 1$$

类似的，我们可以计算出初始攻击力为  $1, 3, 5, 7, 9$  时最终攻击力为 0，初始攻击力为  $0, 2, 4, 6, 8, 10$  时最终攻击力为 1，因此 atm 的一次攻击最多使 drd 受到的伤害值为 1。

## 【样例输入输出 2】

见选手目录下的 *sleep/sleep.in* 与 *sleep/sleep.ans*。

## 【数据规模与约定】

所有测试数据的范围和特点如下表所示

测试点编号	$n, m$ 的规模	约定	备注
1	$2 \leq n \leq 100, m = 0$	$0 \leq t \leq 10^9$  op 一定为 OR, XOR, AND 中的一种	
2	$2 \leq n \leq 1,000$		
3	$1 \leq m \leq 1,000$		
4	$2 \leq n, m \leq 10^5$		存在一扇防御门为 AND 0
5			所有防御门的操作均相同
6			
7	$2 \leq n \leq 10^5$ $2 \leq m \leq 10^9$		所有防御门的操作均相同
8			
9			
10			

## 【运算解释】

在本题中，选手需要先将数字变换为二进制后再进行计算。如果操作的两个数二进制长度不同，则在前补 0 至相同长度。

OR 为按位或运算，处理两个长度相同的二进制数，两个相应的二进制位中只要有一个为 1，则该位的结果值为 1，否则为 0。XOR 为按位异或运算，对等长二进制模式或二进制数的每一位执行逻辑异或操作。如果两个相应的二进制位不同（相异），则该位的结果值为 1，否则该位为 0。AND 为按位与运算，处理两个长度相同的二进制数，两个相应的二进制位都为 1，该位的结果值才为 1，否则为 0。

例如，我们将十进制数 5 与十进制数 3 分别进行 OR, XOR 与 AND 运算，可以得到如下结果：

$$\begin{array}{lll}
 0101 \text{ (十进制 5)} & 0101 \text{ (十进制 5)} & 0101 \text{ (十进制 5)} \\
 \text{OR } 0011 \text{ (十进制 3)} & \text{XOR } 0011 \text{ (十进制 3)} & \text{AND } 0011 \text{ (十进制 3)} \\
 = 0111 \text{ (十进制 7)} & = 0110 \text{ (十进制 6)} & = 0001 \text{ (十进制 1)}
 \end{array}$$