

题目名称	徐老师的聚合数字	徐老师的密码破解	徐老师的口供笔录	徐老师的无聊游戏
文件名	abc.cpp	pwd.cpp	record.cpp	boring.cpp
输入文件	abc.in	pwd.in	record.in	boring.in
输出文件	abc.out	pwd.out	record.out	boring.out
时间限制	1000ms	1000ms	1000ms	1000ms
空间限制	512MB	512MB	512MB	512MB

徐老师的聚合数字

文件读写

[输入文件abc.in](#)
输出文件abc.out

限制

- 1000ms
- 512MB

题目描述

徐老师最近有了一个新想法——聚合数字

简单一点来说，就是对于一个数字，如果它不是个位数，那么就将它的每一位拆开，然后求和，得到新的数字，这样的过程称之为一次 **聚合**

可以发现每个数字 x 通过一定次数的聚合，最终都会成为一个个位数，这个个位数就是 x 的聚合数字

例如 202402 经过一次聚合会变成 $2 + 0 + 2 + 4 + 0 + 2 = 10$ ，再经过一次聚合会变成 $1 + 0 = 1$ ，那么 1 就是 202402 的聚合数字

现在徐老师想知道，对于 $1 \sim n$ 之间所有数字的聚合数字之和是多少？

输入格式

输入一行包含一个整数 n

输出格式

输出一行，表示答案

数据范围

对于 20% 的数据保证： $n \leq 10$

对于 80% 的数据保证： $n \leq 100000$

对于 100% 的数据保证： $n \leq 10^{12}$

样例输入1

10

样例输出1

46

样例输入2

20

徐老师的密码破解

文件读写

输入文件

输出文件

限制

- 1000ms
- 512MB

题目描述

有一天，石老师留下一张纸条后突然失踪了！

纸条上写着的内容如下

```
abcde 1
abcde -> bcdef
or
abcde -> bcdee
or
abcde -> acdee
```

```
z 1
aza -> bza
or
aza -> azb
```

aza -> bzb no!!

aza -> bab no!!

x n

x -> max(?)

机智的徐老师理解了这张纸条上的信息：石老师给出了一个字符串 x 和操作次数 n

每次操作徐老师可以选择字符串 x 中的连续一段，将这段子串的所有字母变为字母表中的下一个字母

即 a 变为 b 、 b 变为 c ... y 变为 z

但是对于一个字母如果已经是 z 了，则不允许操作这个字母，也就是说如果碰到 aza 这种情况，不能同时变化两个 a ，因为中间的 z 是不允许操作的

最后石老师提出了问题，字符串 x 经过最多 n 次操作后能得到最大的字符串是什么？

徐老师相信只要解开这个问题，他就能找到石老师！

P.S.1 字符串的大小关系就是指 **字典序**，字典序的大小判断方法是：从第一位开始向后比较，直到找到不同的那一位，这一位的字母ASCII码大小关系即是这两个字符串的大小关系

P.S.2 例如 $abcd < ac$, $bbb > bazzzz$

输入格式

输入一行，包含一个字符串 x ，和一个整数 n 。

输出格式

输出最多经过 n 次操作后，最大的字符串

数据范围

对于 40% 的数据， $|x| \leq 6$, $n \leq 24$ 。

对于 50% 的数据， $|x| \leq 10$, $n \leq 100$ 。

对于 100% 的数据, $|x| \leq 1000, n \leq 1000$ 。

样例输入1

z 1

样例输出1

z

样例输入2

abcde 23

样例输出2

xyzzz

徐老师的口供笔录

文件读写

[输入文件record.in](#)

输出文件record.out

限制

- 1000ms
- 512MB

题目描述

大家都知道，徐老师打字打的飞快，可以做到“同声记录”，也就是别人边说话，他就可以直接把说的话直接输入到电脑中！

堪比 AI 转换！正确率还高！

这天徐老师被邀请到警察局去帮助一名警官做口供记录

记录格式如下， P 表示警察说的话， C 表示嫌疑犯说的话

```
P: nihao!  
C: wobuhao!  
P: nishihuairenma?  
C: wobushi!
```

但是由于徐老师今天没有吃午饭，头昏脑涨之下导致他的记录有些出错！

首先徐老师可以保证的是对话内容是不会有错的，但是每句话的的所属可能标错了一部分

但是经过徐老师的回忆，徐老师发现了两个人说话的特点：

1. 警察的话里不会包含 buhao
2. 嫌疑犯的话里不会包含 nihao
3. 警察说的话不超过 m 句

现在徐老师想知道他最少可能记录错了几句，最多可能记录错了几句？

如果记录中必然是警察说的话超过了 m 句，则认为这份口供记录是无效的，请输出 NoNoNo！

输入格式

第 1 行两个正整数 n, m 表示即将出现的对话数和警察说话的数量

接下来 n 行，每行包含一句话，形如 $X: \text{xxxxx}$ 。

其中 X 为 P 和 C 中的一个， xxxxx 代表个长度任意的字符串（由小写英文字母、数字、 $.$ 、 $?$ 、 $!$ 、空格构成）

注意：输入格式中 $X:$ 冒号后存在一个空格！

输出格式

如果警察说的话必然超过 m 句, 输出 NoNoNo!

否则输出两个数 a, b , a 表示可能的最少的错误记录数, b 表示最多的可能的错误记录数。

数据范围

对于 30% 的数据, 有 $m = 1$

对另外 10% 的数据, 所有记录都是嫌疑犯说的话

对于 100% 的数据, 有 $1 \leq m \leq n \leq 10^3$, 每句话长度不超过 100

输入样例1

```
4 3
C: nihaoaaa
C: bbbbhuhao!
P: nishihuairenma?
C: nihao.
```

输出样例1

```
2 3
```

样例解释1

第一句话包含 nihao, 说明一定是警察说的, 那么这句话就是一句记录错误的话

第二句话包含 buhao, 说明一定是嫌疑犯说的, 那么这句话是一句记录正确的话

第三句话不包含 nihao 和 buhao, 无法判断记录是否正确

第四句话包含 nihao, 说明一定是警察说的, 那么这句话就是一句记录错误的话

最好情况就是第三句话记录是正确的, 错误 2 句

最坏情况就是第三句话记录是错误的, 错误 3 句

输入样例2

4 2
C: nihaoaaa
C: bbbbhuhao!
P: nishihuaiarenma?
C: nihao.

输出样例2

3 3

样例解释2

第一句话包含 nihao , 说明一定是警察说的, 那么这句话就是一句记录错误的话

第二句话包含 buhao , 说明一定是嫌疑犯说的, 那么这句话是一句记录正确的话

第三句话不包含 nihao 和 buhao , 无法判断记录是否正确

第四句话包含 nihao , 说明一定是警察说的, 那么这句话就是一句记录错误的话

由于警察说的话不超过 2 句, 所以第三句话不能是警察说的, 所以第三句话也一定是错误的

最好情况和最坏情况均为 3 句

徐老师的无聊游戏

文件读写

[输入文件boring.in](#)

输出文件boring.out

限制

- 1000ms

题目描述

最近家里断网了，徐老师真的非常无聊

于是石老师给徐老师设计了一个很有(wu)趣(liao)的游戏

石老师会先给出徐老师两个数字 x, y

为了让徐老师能够玩的有意思一些，石老师指定了 n 个不同的变化规则

对于第 i 个操作有三个属性 a, b, c ，使用这个操作需要花费 c ，可以使得数字发生以下两个变化之一

1. $x + a, y + b$

2. $x + b, y + a$

而游戏的目标是用最小的花费使得 x, y 的差值刚好等于 m ，即 $|x - y| = m$

但是这个游戏对于徐老师来说太简单了，于是石老师又给出了新的规则：

在游戏过程中， $x - y$ （无绝对值）的值必须一直保持在 $[L, R]$ 的范围内，即 $L \leq x - y \leq R$

现在徐老师正在玩这个游戏，他想知道最终的答案是多少，以此验证自己的做法是否正确

P.S. 每一个操作可以无限次数使用

输入格式

输入第一行包含两个整数 x, y ，含义如题

输入第二行包含两个整数 L, R ，含义如题

输入第三行包含两个整数 n, m ，表示游戏有 n 种操作，目标值是 m

接下来 n 行，每行包含三个整数 a_i, b_i, c_i 分别表示第 i 种操作的三个属性

输出格式

输出一行，表示最小的花费，如果无解则输出 2147483647，即 $2^{31} - 1$

数据范围

对于前 30% 的数据： $-20 \leq x, y \leq 20, -2 * 10^3 \leq L \leq R \leq 2 * 10^3, n \leq 10$ ，保证答案不超过 100；

对于另外 30% 的数据： $-100 \leq x, y \leq 100, L = -2 * 10^3, R = 2 * 10^3, n \leq 100$ ，保证数据随机。

对于 100% 的数据： $-10^3 \leq x, y \leq 10^3, -2 * 10^3 \leq L \leq R \leq 2 * 10^3, 0 \leq m \leq 2 * 10^3, 0 \leq n \leq 2 * 10^3, -10^4 \leq a_i, b_i \leq 10^4, 0 < c_i \leq 10^4$ 。

样例输入

5 5
-2 5
5 5
3 -1 6
-4 -3 3
-2 -4 1
4 2 1
2 3 1

样例输出

3

样例解释

先进行 3 号操作， $x + (-2) = 3, y + (-4) = 1$ ，花费 1
再进行 4 号操作， $x + 4 = 7, y + 2 = 3$ ，花费 1
再进行 5 号操作， $x + 3 = 10, y + 2 = 5$ ，花费 1
达到目标值 $|x - y| = m = 5$ ，最小花费为 3