

# 版本号绝对差

## 题目背景

在计算机软件中，我们习惯用版本号来标记不同时间下可以通过运行的代码。

## 题目描述

版本号是由一个字母 $V$ 开头作为提示，由一个数字代表主版本号，和一个数字代表副版本号组成的，其中主副版本号之间用 $'.'$ 分割。

例如： $V1234.5678$ 就是一个版本号，主版本号为1234，副版本号为5678。

副版本号到达上限的时候，下一个版本主版本号就会 $+1$ ，例如设副版本号的上限为10，那么 $V1.10$ 的下一个版本就是 $V2.0$ ， $V3.1$ 的上一个版本就是 $V3.0$

现在给出一个软件的两个版本号，给出副版本号上限，请你求出这两个版本的绝对差是多少

注意：绝对差表示的是两个版本号的差值的绝对值。

## 输入格式

第一行包含一个字符串 $a$ ，表示版本号1。

第二行包含一个字符串 $b$ ，表示版本号2。

第三行 $k$ ，表示副版本号的上限值。

保证输入的版本号一定是有效版本号。

## 输出格式

一行，表示版本号的绝对差。

## 样例 #1

### 样例输入 #1

```
1 | v3.2
2 | v3.0
3 | 5
```

### 样例输出 #1

```
1 | 2
```

## 样例 #2

## 样例输入 #2

```
1 v12.10
2 v15.7
3 11
```

## 样例输出 #2

```
1 32
```

## 提示

设 $n = a.size(), m = b.size(), l = k.size()$

对于30%的数据, 有 $1 \leq n, m, l < 18$ 。

对于70%的数据, 有 $1 \leq n, m \leq 100, 1 \leq l \leq 5$ 。

对于100%的数据, 有 $1 \leq n, m, l \leq 5000$ 。

样例解释1:  $V3.0$ 需要经过 $V3.1, V3.2$ , 两个版本到 $V3.2$ 。