

A. We love Gromah!

题面描述

今天，你写完了所有的作业，肝完了所有的 DDL，糊弄完了所有的 pre。你，终于获得了一个悠闲的周末。于是你打开了 `icpc.fudan.edu.cn` 参加复旦大学 2019 年度程序设计竞赛校赛的网络预选赛，跃跃欲试，准备屠榜。但是你发现本场比赛最水的题目的样例数据非常奇怪：它只有一行输入，这一行里面包含了若干个英文单词，每个英文单词仅由大写或小写英文字母组成。此外，相邻两个英文单词之间用一个空格隔开了。奇怪的是，这一行里面的英文单词的字母大小写十分混乱（例如 “sOmE TIImES nAiVe”），导致你难以理解其中的含义。于是你准备写一个程序，把这些奇怪的大小写变成规范的标题格式的大小写的形式。

具体而言，你的程序需要能够读取一行包含若干个英文单词的语句，并且将其转为标题格式后输出。即对于语句中的每个单词，将单词的第一个字母改为大写，其余字母均改为小写，并重新输出这一行单词。

输入格式

输入数据仅一行，包含至多 100 个单词，并且每个单词包含至多 20 个大写或小写的英文字母，每两个相邻单词之间有且仅有一个空格。

输出格式

输出仅一行，输出这一行语句转为标题格式后的结果。

样例 1

样例输入

```
boTH u and I lOvE GROMAH whO IS a ReAlLy undoubtEdly superB mAn
```

样例输出

```
Both U And I Love Gromah Who Is A Really Undoubtedly Superb Man
```

样例解释

无可奉告。

B. We love Gromah!

题面描述

给定分数 a/b ，要求将其化为最简分数。

输入格式

输入一个分数 a/b ，以格式 “a/b” 输入。即分子在前，分母在后，两者之间有一个分数线斜杠 “/” 分隔。

输出格式

输出最简分数。格式与输入格式相同。注意你需要保证输出的分子和分母均为正整数。

数据限制

$1 \leq a, b \leq 10^{18}$ 。

样例 1

样例输入

8/12

样例输出

2/3

样例解释

无可奉告。

样例 2

样例输入

12/6

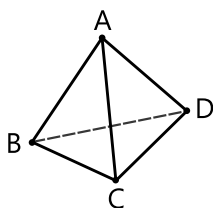
样例输出

2/1

C. We love Gromah!!!

题面描述

一只小蚂蚁在一个四面体的边上爬行。一开始的时候（即时刻为 0 的时候），小蚂蚁在下图中 A 点的位置。接下来每过一秒钟，小蚂蚁将会从当前所处的点处出发，到达四面体上另外三个点中的某一个点处。并且小蚂蚁到达另外三个点中任意一个点的概率是相等的，即到达每个点的概率均为 $1/3$ 。例如，如果在时刻 t 时小蚂蚁站在 C 点，则在 $t+1$ 时刻，小蚂蚁可能处在 A 点、 B 点或者 D 点。Codeplay 想知道，在时刻 t 的时候，小蚂蚁待在点 A 的概率究竟是多少。



这是一个四面体

输入格式

输入数据仅有一行，包含一个非负整数表示时刻 t 。

输出格式

输出时刻 t 时小蚂蚁待在 A 点的概率。设你输出的答案为 x ，标准答案为 y ，则只要 $|x - y| < 10^{-7}$ 我们就会认为你输出了正确的答案。我们建议你输出到小数点后至少 10 位。注意，请不要输出过长的答案（如长度超过 40 个字符）。

数据范围

$$0 \leq t \leq 10^{18}。$$

样例 1

样例输入

0

样例输出

1.000000000000000000000000

样例解释

一开始小蚂蚁一定在 A 点上，故概率为 1。

样例 2

样例输入

2

样例输出

0.33333333333333334

样例 3

样例输入

5

样例输出

0.24691358024699999

D. We love Gromah!!

题面描述

你是一个土豪，你准备做一笔交易，需要给某人支付一大笔人民币现金，这笔现金的总额为 n 元。众所周知，目前市面上流通纸质人民币主要有 100 元、50 元、20 元、10 元、5 元和 1 元共 6 种面值。由于你是一个土豪，所以每种面值的人民币在你家的金库里都有无限多张。虽然是个土豪，为了方便，你不希望使用太多的纸币来支付这一笔现金。于是你想知道最少用多少张纸币才能恰好支付总额为 n 元的现金？

输入格式

输入数据仅一行，包含一个正整数 n ，表示需要支付的现金总额。

输出格式

输出一行包含 6 个非负整数，用空格隔开，从左至右依次是面值为 100 元、50 元、20 元、10 元、5 元和 1 元的人民币使用的张数。如果有多种支付方案，输出任意一种即可。你只需要保证使用的纸币总数最少。

数据限制

$1 \leq n \leq 50000$ 。

样例 1

样例输入

17

样例输出

0 0 0 1 1 2

样例解释

由小学数学我们可以知道 $1 \times 10 + 1 \times 5 + 2 \times 1 = 17$ 。

E. We love Gromah!!

题面描述

在围棋的一种非正式的规则中，执白棋的一方可以有一次连落两粒棋子的机会，但是使用这次机会的回合数必须在棋局开始之前就说好，即只能在自己规定的某一手时连落两子。现在到了 FSX 和 Gromah 对局的关键时刻：FSX 手持白棋并且到了可以连落两子的机会的时候了。由于 FSX 迫切地想挖空 Gromah 的布局，因此他希望能利用这一步提掉尽可能多的黑棋。你非常好奇 FSX 这一手究竟最多能干掉 Gromah 多少目黑棋，于是你拿到了当时的棋盘局面，准备计算出这个最大值。

注意：连续落子意味着你不需要考虑你落下的两粒棋子是否为活棋。即你可以在任意没有落子的地方落子。此外，为了方便，在本题中你不需要考虑全局同形的情况。

如果你不知道围棋的基本规则，可以参考这里：<https://baike.baidu.com/item/%E4%B8%AD%E5%9B%BD%E5%9B%B4%E6%A3%8B%E8%A7%84%E5%88%99/5425065?fr=aladdin#1>

输入格式

输入恰好 19 行，每行仅 19 个字符，每个字符表示棋盘上对应位置处的棋子状态：白棋、黑棋或者没有棋子。数据中使用字符“o”表示白棋，用字符“+”表示黑棋，以及字符“.”表示该位置上没有落子。

输出格式

输出一行一个非负整数，表示最多能提掉的黑棋数目。

数据限制

保证初始局面所有棋子都是活棋，并且至少有两个点上没有落子。

样例 1

样例输入

```
.....
.....
.....
.....
.....
.....
.....
.....O.....
.....O+O+O.....
.....O+.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
```

样例输出

```
2
```

样例解释

最优的围法如下：

```
.O...
OXO+O
OXO..
.O...
```

上面展示的局面中字符“x”表示被提掉的黑棋。

F. We love Gromah!!!!

题面描述

小喵喵要去经过迷途竹林拜访辉夜姬（Houraisan Kaguya）。现在小喵喵被困在迷途竹林里面了，他需要找到正确的竹子才能找到永远亭。在迷途竹林里面，所有的竹子排成一行，编号从左至右依次为 $1, 2, 3, \dots, |T|$ ，每个竹子 i 上面有一个英文字母 c_i ，可以按照顺序把竹子上的字母排成一个字符串 T 。同时辉夜姬给了小喵喵一个字符串 S 作为密码。对于每个竹子 i 对应字母开始的后缀字符串 T_i （即由 $c_i c_{i+1} \dots c_{|T|}$ 构成的字符串），可以定义一个正确度 C_i 。正确度为如果以 S 在 T_i 上进行匹配，能匹配上的字符总数。特别的，这里的匹配是可以循环进行的，即如果完整地匹配了一次 S ，那么可以从 T_i 当前最后匹配到的字符的位置的下一个位置开始，继续重新匹配 S ，最终直到失配或者 T_i 结束为止。那么正确的竹子就是正确度 C_i 最大的那个竹子 i 。如果有多个竹子的正确度同时最大，那么正确的竹子是编号最小（最靠左）的那个。所以小喵喵要怎么找到永远亭呢？

输入格式

第一行输入一个字符串 T ，表示从左到右的竹子上的字母连起来组成的字符串。

第二行输入一个字符串 S ，表示作为密码的字符串。

输出格式

输出正确的竹子的编号。（最左边的竹子编号为 1 ，之后依次为 $2, 3, \dots, |T|$ ）。

数据范围

$1 \leq |T|, |S| \leq 10^5$ 。

样例 1

样例输入

```
aabaaa  
a
```

样例输出

```
4
```

样例解释

从第四个竹子开始能匹配 3 次，这是最大的正确度。

样例 2

样例输入

abcde
f

样例输出

1

样例解释

所有的正确度都是 0，故取最左边的点。

样例 3

样例输入

aabaaa
aba

样例输出

2

样例解释

从第 2 个字符开始，可以匹配一轮“aba”。然后进行第二轮匹配，从第 5 个字符开始，再可以有 1 次匹配，即 T 的第 5 个字符和 S 的第 1 个字符成功匹配。之后就无法继续匹配。所以匹配字符数为 4，匹配起点为 2。