

全国信息学分区联赛模拟试题（二）

按照 NOIP 比赛要求建立文件夹保存, 8 月 14 日前发到

wushenguang@sina.com 邮箱

【试题概览】

试题名称	倒水	单词缩写	求和	象棋比赛
提交文件	water.pas/c/cpp	abbr.pas/c/cpp	sum.pas/c/cpp	chess.pas/c/cpp
输入文件	Water.in	Abbr.in	Sum.in	Chess.in
输出文件	Water.out	Abbr.out	Sum.out	Chess.out
时间限制	1s	1s	1s	1s
空间限制	50MB	50MB	50MB	50MB
题目来源	广东省题	广东省题	广东省题	广东省题

1. 倒水

【题目描述】

一天, 树树买了 N 个容量可以认为是无限大的瓶子, 初始时每个瓶子里有 1 升水。树树发现瓶子实在太多了, 于是他决定保留不超过 K 个瓶子。每次他选择两个当前含水量相同的瓶子合并, 把一个瓶子的水全部倒进另一个瓶, 然后把空瓶丢弃(不能丢弃有水的瓶子)。

显然在某些情况下树树无法达到目标, 你如 $N=3, K=1$ 。此时树树会重新买一些新的瓶子(新瓶子容量无限, 开始时有 1 升水), 以达到目标。

现在树树想知道, 最少需要买多少新瓶子才能达到目标呢?

【输入文件】

一行两个正整数 N, K ($1 \leq N \leq 10^9, K \leq 1000$)。

【输出文件】

一个非负整数, 表示最少需要买多少新瓶子。

【样例输入】

3 1

【样例输出】

1

【数据规模】

对于 30% 数据, $N \leq 3 \times 10^5$;

对于 100% 的数据如题目。

2. 单词缩写

【题目描述】

树树发现好多计算机中的单词都是缩写的, 如 GDB, 它是全称 Gnu DeBug 的缩写。但是, 有时缩写对应的全称会不固定, 如缩写 LIUNIX, 可以理解为:

- (1) LINus's UniX
- (2) LINUs's miniX
- (3) Linux Is Not Unix

现在树树给出一个单词缩写, 以及一个固定的全称 (若干个单词组成, 空格隔开)。全称中可能会有无效的单词, 需要忽略掉, 一个合法缩写要求每个有效单词至少有一个字符出现在缩写中, 缩写必须按顺序出现在全称中。

对于给定的缩写和一个固定的全称, 问有多少种解释方法? 解释方法为缩写的每个字母在全称每个有效单词中出现的位置, 有一个字母位置不同, 就认为是不同的解释方法。

【输入文件】

第一行输入一个 N , 表示有 N 个无效单词;

接下来 N 行分别描述一个有小写字母组成的无效单词;

最后是若干个询问, 先给出缩写 (只有大写字母), 然后给出一个全称, 读入以 "LAST CASE" 结束。

【输出文件】

对于每个询问先输出缩写, 如果当前缩写不合法, 则输出 "is not a valid abbreviation", 否则输出 "can be formed in i ways" (i 表示解释方法种数)。

【样例输入】

```
2
and
of
ACM academy of computer makers
RADAR radio detection and ranging
LAST CASE
```

【样例输出】

ACM can be formed in 2 ways

RADAR is not a valid abbreviation

【数据规模】

$1 \leq N \leq 100$ ，每行字符串长度不超过 150，询问不超过 20，最后方案数不超过 10^9 。

3.求和

【题目描述】

高斯在他还是小 P 孩的时候就求出 $1 + 2 + \cdots + n = n*(n+1)/2$;

LT 在他还是小 P 孩的时候就求出 $1/(1*2) + \cdots + 1/((n-1)*n) = 1-1/n$;

现在，在你还是小 P 孩的时候，你要求出：

$$\frac{1}{1*2*\cdots*m} + \cdots + \frac{1}{n*(n+1)*\cdots*(n+m-1)} = S;$$

【输入文件】

输入两个整数 n、m。

【输出文件】

输出占两行，第一行一个整数 X，第二行一个整数 Y，表示 $S=X/Y$ ，且 X，Y 互质。

【样例输入】

1 2

【样例输出】

1

2

【数据规模】

$m>1, n>0$; 50%的数据满足 $n \leq 50$; 100%的数据满足 $n+m \leq 500$

4.象棋比赛

【题目描述】

有 N 个人要参加国际象棋比赛，该比赛要进行 K 场对弈。

每个人最多参加两场对弈，最少参加零场对弈。

每个人都有一个人等级（用一个正整数来表示）。

在对弈中，等级高的人必须用黑色的棋子，等级低的人必须用白色的棋子。

每人最多只能用一次黑色的棋子和一次白色的棋子。

为了增加比赛的客观度，观众希望 K 场对弈中双方的等级差的总和最小。

比如有 7 个选手，他们的等级分别是 30；17；26；41；19；38；18.，要进行 3 场比赛。

最好的安排是 Player 2 vs Player 7，Player 7 vs Player 5，Player 6 vs Player 4，此时等级差的总和等于 $(18-17)+(19-18)+(41-38)=5$ 达到最小。

【输入文件】

第一行有两个正整数 N, K ；

接下来有 N 行，第 i 行表示第 $i+1$ 个人的等级。

【输出文件】

在第一行输出最小的等级差的总和。

【样例输入】

```
7 3
30
17
26
41
19
38
18
```

【样例输出】

```
5
```

【数据规模】

在 90%的数据中， $1 \leq N \leq 3000$ ；

在 100%的数据中， $1 \leq N \leq 100000$ ；

保证所有输入数据中等级的值小于 10^8 ， $1 \leq K \leq N-1$ 。