

Problem A. 可见字符

题目背景

刘老师在课上详细得介绍了 *ASCII* 码，然而小明翘课了，小红上课玩手机，都没听。眼看就要上机了，为了能够通过上机第一题，他们让mentor给他们出了道题。

题目描述

输入整数 N ，且 $0 \leq N \leq 10000$ 。

如果 N 比最大 *ASCII* 值还大，输出 "N IS TOO BIG !"。

如果 N 对应的字符不可见，输出 "INVISIBLE !"。

如果 N 对应的是可见字符（不包括空格），输出该字符。

输入格式

多组输入，每组一行，为 N 。

输出格式

每行输入对应一行输出，要求见题目描述。

样例输入

```
10000
3
65
```

样例输出

```
N IS TOO BIG !
INVISIBLE !
A
```

Hint

多组数据的常用输入方式如下：

```
int x;
while (scanf("%d", &x) != EOF) {
    // do something
}
```

让我看看哪个小可爱这次输出不复制粘贴。

ASCII 码的最大值为 127，不包括空格的可见字符的 *ASCII* 值范围为 33 ~ 126。

Problem B. 算分

题目描述

一次上机共 10 道题，每道题的分值为 a_1, a_2, \dots, a_{10} 。蛙宝想知道某一次上机得了多少分，但是评测机只告诉了他每道题所得分的百分比 b_1, b_2, \dots, b_{10} 。你能帮他算算在这次上机中他得了多少分吗？

输入格式

输入共两行：

第一行用空格分割的 10 个整数 a_1, a_2, \dots, a_{10} ，表示每道题的分值。

第二行用空格分割的 10 个整数 b_1, b_2, \dots, b_{10} ，表示每道题得分的百分比（%）。

数据范围： $1 \leq a_i, b_i \leq 100$ 且均为整数。

输出格式

输出一行一个数，表示蛙宝本场的得分（结果保留两位小数）。

样例输入

```
20 10 10 10 10 10 10 10 10 10
100 90 80 70 60 50 50 0 0 0
```

样例输出

```
60.00
```

样例说明

蛙宝的得分为：

$$20 \times 100\% + 10 \times 90\% + 10 \times 80\% + 10 \times 70\% + 10 \times 60\% + 10 \times 50\% + 10 \times 50\% + 10 \times 0\% + 10 \times 0\% + 10 \times 0\% = 60.00$$

AUTHOR: Blore

Problem C. 大小写转换

题目描述

给定一个长度未知的字符串 A （含空格），请你对它进行大小写转换操作后得到字符串 B 并输出。大小写转换操作定义如下：

对 A 的每个字符：

1. 若该字符为小写字母，将其转换为大写字母。
2. 若该字符为大写字母，将其转换为小写字母。
3. 若该字符既不为小写字母，又不为大写字母，则该字符保持不变。

输入格式

一行，长度若干的字符串 A （含空格）。

输出格式

一行，字符串 A 经大小写转换操作后得到的字符串 B 。

样例输入1

```
To be or not to be, that's a question.
```

样例输出1

```
TO BE OR NOT TO BE, THAT'S A QUESTION.
```

样例输入2

```
abCDeFg1234567
```

样例输出2

```
ABcDeFg1234567
```

Hint

- 保证字符串 A 长度小于等于 100000。
- 保证字符串 A 中不可见字符只有空格。
- 如果你对读入没有思路，可以复制以下代码：

```
char ch;
while(scanf("%c",&ch)!=EOF){
    //    do something...
}
```

- 以上代码表示依次读入字符串 A 的每个字符 ch ，分别进行处理。
- 在本地测试样例时需要手动输入 Ctrl+z 结束程序哦~

Author : yzh

Problem D. 后撤步

题目背景

少壮不努力，老大后跳带剑气。

题目描述

一个棋子处在二维平面的 (x_0, y_0) 处。现在有一个以 (p, q) 为圆心，半径为 R 的圆。这个棋子想要跳出这个圆，但是我们都知，棋子是无法看见这个世界的，所以棋子只能按照自己脑海中的方向前进。每次棋子都会确定一个向量 (x_i, y_i) ，并按照这个方向前进，即从原来的 (x, y) 直接跳到 $(x + x_i, y + y_i)$ 的位置。

但是由于某种原因，这个棋子的所有前进都变成了后撤，也就是棋子实际上到达的是 $(x - x_i, y - y_i)$ 的位置。

这个棋子并不知情，它还是依次确定了 n 个向量，并准备依次前进。

虽然这个棋子不知道能否跳出这个圆，而作为局外人的你，一定能给出答案吧。

输入格式

第一行一个数 n 。

第二行是棋子的初始坐标 (x_0, y_0) 。

第三行三个数，分别是圆心坐标 p, q 和半径 R 。

接下来 n 行，每行都是一个向量 (x_i, y_i) 。

输出格式

如果棋子**最终**能跳出这个圆，输出棋子的**最终**位置 (x, y) 。

如果不能，则输出 "No way!"。

样例输入1

```
5
(2,3)
0 0 8
(6,2)
(-2,3)
(-3,-7)
(4,-2)
(-9,6)
```

样例输出1

```
No way!
```

样例输入2

```
4
(1,0)
1 -2 5
(4,0)
(-2,-1)
(1,-7)
(0,2)
```

样例输出2

```
(-2,6)
```

数据范围

$0 \leq n \leq 1000, 0 \leq r \leq 10^9$;

$p, q \in [-10^6, 10^6]$;

$\forall i \in [0, n], x_i, y_i \in [-10^6, 10^6]$;

保证所有数均为整数。

Hint

- 可以尝试用如同 `scanf("(%d,%d",&x,&y)` 形式的输入来读入一个点的坐标。
- 可以使用 `scanf("%d\n",&x)` 的形式来读入一个整数并强制忽略数字后面的换行符。
- 定义：如果棋子恰好处在圆的边界上，我们认为它还是没有跳出这个圆。

AUTHOR: Stockholm

Problem E. Monica的加密

题目描述

*Monica*和好朋友*Lily*交往密切。由于某些特殊的原因，*Monica*和*Lily*约定对两人的信息进行加密。在约定的密码规则中，首先将**字母 a-z 分别用数值 0-25 代替**，然后对于明文字母*A*，密文字母*B*，密钥*K*，若在*A*之前已经加密了字母，则*M*为前一个被加密字母的密文代表的数值，否则若*A*是全部信息中第一个需要被加密的字母，则*M* = 1。密码的加密规则如下：

$$B = (A + K * (M - 13)) \mod 26$$

由于*Monica*忙着和*Lily*玩耍，于是她把编写加密程序这个任务交给了她的另一个好朋友，也就是你。信息中只有小写字母。

输入格式

第一行一个整数*K*，表示密钥。

接下来一行字符串，表示需要加密的信息。

输出格式

输出一行，表示加密后的信息。

样例输入 1

```
19990903
know
```

样例输出 1

```
uszm
```

样例输入 2

```
19990526
have
```

样例输出 2

```
rofc
```

样例解释

第一个样例中，第一个需要加密的字符为 "k"，对应数值为10，密钥 $K = 19990903$ ， $m = 1$ 。计算得到密文对应数值：

$$B = (10 + 19990903 * (1 - 13)) \mod 26 = 20$$

数值20对应的字符为 "u"

数据范围

对于100%的数据：

$1 \leq K \leq 2000000000$ 且 字符串长度不超过100。

Hint

1. 请注意输入数据第一行的行尾换行符不应被输出。
2. 小写字母的 `ascii` 码范围为97~122，且字母a的 `ascii` 码为97。
3. 本题我们约定， mod 取模运算的结果为一最小非负整数，具体的：

在C语言中的取模 $\%$ 运算采用的是truncate算法，负数模正数的结果仍然是负数。例如 $-6 \% 5 = -1$ 。为了使结果为正数我们可以做类似 $-6 \% 5 = -6 \% 5 + 5 = 4$ 的处理。

同时取模 $\%$ 运算具有如下性质：

$$\begin{aligned}(A \pm B) \% P &= (A \% P \pm B \% P) \% P \\ A \times B \% P &= (A \% P \times B \% P) \% P\end{aligned}$$

AUTHOR: Monica

Problem F. 淑芬来啦

题目背景

sheep在国赛建模时遇到了一个很阴间的FAST望远镜的反射面调节的题目，由于离散化函数会对课程后面的Abel和Dirichlet判别法有所帮助，爱淑芬的sheep决定把这一算术题目分享给大家。

在FAST反射面的调节中，每一个反射面板均为一个大致的圆弧面三角形，为了准确地将光线反射到馈源舱，在整体面板统筹考虑之前需要讨论对每一块圆弧面三角形的反射面板进行优化调节，找到对单一考虑面板时对每一块反射面板的评估函数。

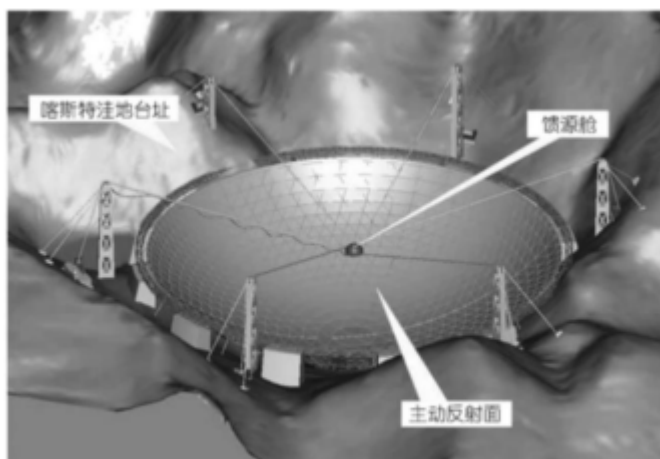


图1 FAST 三维示意图

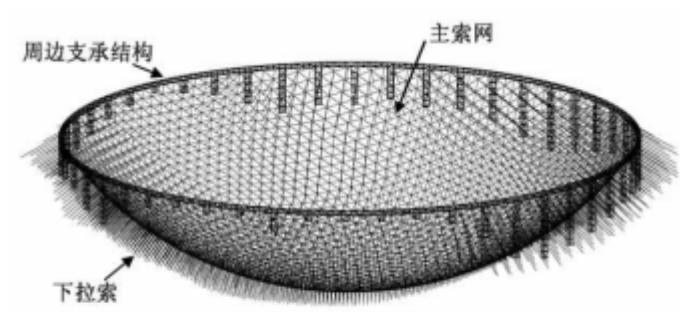


图2 整体索网结构

你的任务是帮sheep计算单一考虑某一弧面反射面板优化时的涉及光反射部分的单一光线的反射位置。

题目描述

为了简化题目难度，sheep已经帮大家计算好了相应的公式之一，仅需要带入公式计算数值就可以得到馈源舱的对单点光源的接收情况啦（题目所需函数提示中已给出）

$$x = \begin{cases} \frac{1}{\sqrt{2\pi c}} e^{-\frac{(x_1-p)^2+(y_1-p)^2}{2c^2}} \left(x_1 + \frac{|z_1| \tan 2\theta}{\sqrt{1+b^2/a^2}} \right) & , x_1 < 0 \\ \frac{1}{\sqrt{2\pi c}} e^{-\frac{(x_1-p)^2+(y_1-p)^2}{2c^2}} \left(x_1 - \frac{|z_1| \tan 2\theta}{\sqrt{1+b^2/a^2}} \right) & , x_1 > 0 \end{cases}$$
$$y = \frac{1}{\sqrt{2\pi c}} e^{-\frac{(x_1-p)^2+(y_1-p)^2}{2c^2}} \left(y_1 + \frac{b}{a} (x - x_1) \right)$$

为了能将上述公式离散化绘图，便引入了变量 z ：

$$z = \frac{\arctan(p \cos x)^{\ln(1+|\sin x|)}}{2 + |\sinh y|}$$

你的任务便是计算 x, y, z

输入格式

第一行输入三个非零数字 x_1, y_1, z_1 ；

第二行输入五个非零数字 a, b, c, θ, p 。

输出格式

输出三行三个数字分别对应 x, y, z ，保留两位小数。

样例输入1

```
2 1.5 4
1 2.3 1.1 0.8 3
```

样例输出1

```
5.62
0.98
0.34
```

样例输入2

```
-84.6037 201.326 -206.2772
67.8465 -94.5718 162.2734 0.2625 7.8234
```

样例输出2

```
-0.20
1.09
0.32
```

Hint

π 可取3.1415926535

e 可取2.718281828

(馈源舱可以等效为一个半径为1m的圆，实际求得的 x, y 就是反射光线在以馈源舱底面为基准面的平面坐标)

当我们需要使用一些常见的数学公式时，可以利用 C 语言常用的函数库 `math.h`，在程序的开头部分加上一行 `#include<math.h>` 就可以使用其中的函数了。下面是一些 `math.h` 中常见的函数：

1.三角函数


```
double sin (double);正弦  
double cos (double);余弦  
double tan (double);正切
```

2.反三角函数

```
double asin (double); 结果介于 $[-\pi/2, \pi/2]$   
double acos (double); 结果介于 $[0, \pi]$   
double atan (double); 反正切（主值），结果介于 $[-\pi/2, \pi/2]$   
double atan2 (double, double); 反正切（整圆值），结果介于 $[-\pi, \pi]$ 
```

3.双曲三角函数

```
double sinh (double);  
double cosh (double);  
double tanh (double);
```

4.指数与对数

```
double frexp (double value, int *exponent);  
把浮点数 x 分解成尾数和指数。返回值是尾数，并将指数存入 exponent 中。  
其中尾数取值在  $(-1, -0.5]$  或  $[0.5, 1)$  或 0。  
double ldexp (double x, int exp); 这个函数刚好跟上面的 frexp 函数功能相反，它的返回值是 x 倍的 2 的指数幂  
double log (double); 以 e 为底的对数  
double log10 (double); 以 10 为底的对数  
double pow (double x, double y); 计算 x 的 y 次幂  
float powf (float x, float y); 功能与 pow 一致，只是输入与输出皆为单精度浮点数  
double exp (double); 求取自然数 e 的幂  
double sqrt (double); 开平方根
```

5.取整

```
double ceil (double); 取上整，返回不比 x 小的最小整数  
double floor (double); 取下整，返回不比 x 大的最大整数，即高斯函数  $[x]$ 
```

6.绝对值

```
double fabs (double); 求实型的绝对值  
double cabs (struct complex znum); 求复数的绝对值
```

7.标准化浮点数

```
double frexp (double f, int *p); 标准化浮点数， $f = x * 2^p$ ，已知 f 求 x, p (x 介于  $[0.5, 1]$ )  
double ldexp (double x, int p); 与 frexp 相反，已知 x, p 求 f
```

8.取整与取余

```
double modf(double value, double *iptr); 拆分 value 值，返回它的小数部分，iptr 指向整数部分。  
double fmod (double, double); 返回两参数相除的余数
```

9.其他

```
double hypot(double x,double y);已知直角三角形两个直角边长度，求斜边长度
double ldexp(double x,int exponent);计算x（2的指数幂）
double poly(double x,int degree,double coeffs []);计算多项式
int matherr(struct exception *e);数学错误计算处理程序
```

AUTHOR: Oh so many sheep

Problem G. 开还是关?

题目描述

有 N 盏灯，编号从 1 到 N ，最开始这 N 盏灯都处于关闭状态。

现在有 N 个人依次来改变灯的状态。其中，第 k 个人会改变第 $k, 2k, 3k, \dots$ 盏灯的状态。换言之，第 k 个人会改变所有编号为 k 的倍数的灯的状态。改变状态指将灯由关变开，或由开变关。

例如，第一个人将灯全部打开，第二个人将编号为 2 的倍数的灯关闭，第三个人将编号为 3 的倍数的灯做相反处理.....

请你回答，在 N 个人都结束自己的操作后，有哪些灯是开着的？

输入格式

输入仅一行，一个整数 N ，含义见上。

输出格式

在一行中从小到大输出还开着的灯的编号，两个编号之间用空格隔开。

样例输入

10

样例输出

1 4 9

Hint

保证 $N \leq 100000$ 。

如果没有思路的话，试着手动模拟一下开关灯操作，多试几个 N ，尝试发现规律。

Problem H. 一元线性回归

题目描述

在某个实验中，对于自变量 x 和因变量 y ，可以得到 n 组测量数据 $(x_i, y_i), i = 1, 2, \dots, n$ 。为得到 y 关于 x 的关系，常采用最小二乘法得到最符合测量值的拟合直线 $y = a + bx$ 。其中 a, b 称为回归系数，可通过以下公式得到（推导过程已省略）：

$$b = \frac{\bar{x}\bar{y} - \overline{xy}}{\bar{x}^2 - \overline{x^2}}$$

$$a = \bar{y} - b\bar{x}$$

其中：

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

$$\bar{y} = \frac{1}{n} \sum_{i=1}^n y_i$$

$$\overline{x^2} = \frac{1}{n} \sum_{i=1}^n x_i^2$$

$$\overline{xy} = \frac{1}{n} \sum_{i=1}^n x_i y_i$$

分别表示自变量 x 的平均值，因变量 y 的平均值， x^2 的平均值， xy 的平均值。

此外，定义“偏差值” s 为：

$$s = \sum_{i=1}^n [y_i - (a + bx_i)]^2$$

即测量值 y 与回归值 $a + bx$ 之差的平方和。该数值越小，表示拟合程度越好。

小H通过实验测量得到了 n 组数据，并通过上述方法得到了回归系数 a, b 以及对应的“偏差值” s ，但他觉得“偏差值” s 太大了。为了减少“偏差值”，现在小H可以删除某一组数据并重新计算剩余 $(n - 1)$ 组数据的 a, b, s 。请帮帮小H，编程求解何时可以使“偏差值” s 最小，并输出此时 s 的值。

输入格式

输入共3行。

第一行为一个正整数 n 表示测得的数据组数。

第二行为 n 个整数 x_1, x_2, \dots, x_n 表示自变量 x 。

第三行为 n 个整数 y_1, y_2, \dots, y_n 表示对应的因变量 y 。

输出格式

输出一行一个小数 s ，表示“偏差值”的最小值，保留两位小数。

样例输入1

```
5
1 2 3 4 5
1 3 3 3 5
```

样例输出1

```
0.69
```

样例输入2

```
5
1 2 3 4 5
1 3 3 4 5
```

样例输出2

```
0.00
```

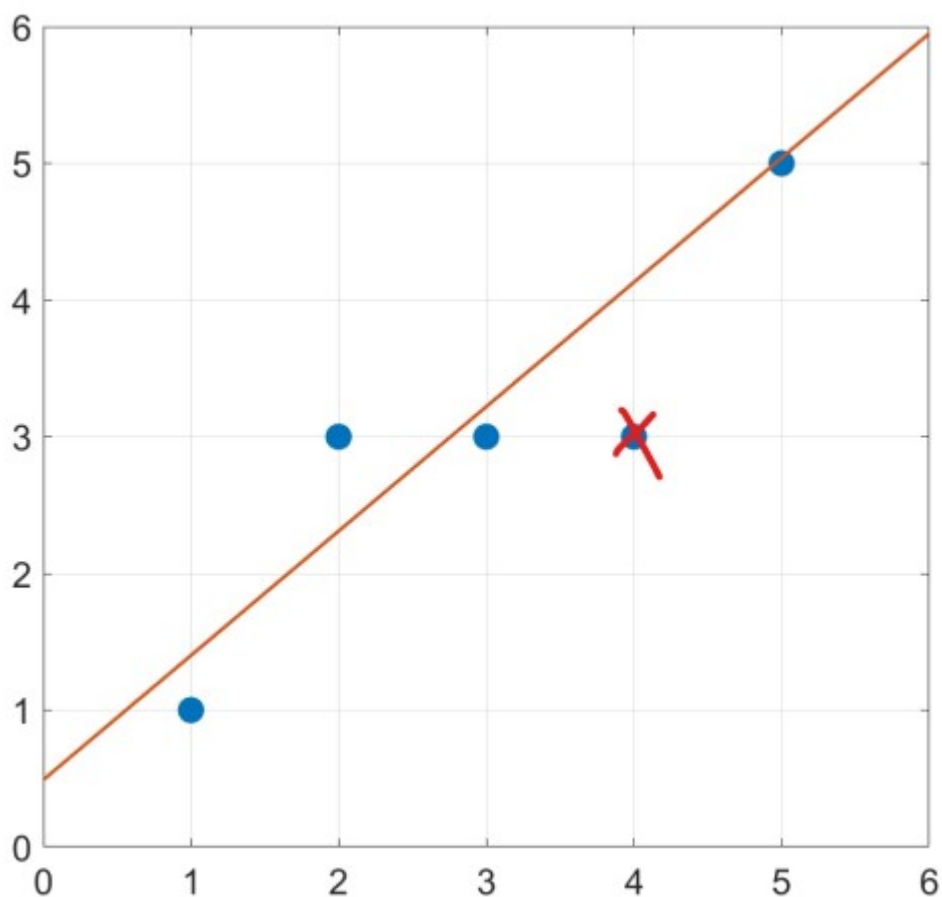
数据范围

$$4 \leq n \leq 100$$

$$-100 \leq x_i, y_i \leq 100, i = 1, 2, \dots, n$$

Hint

- 第一个样例中，我们可以删除第2组数据 $(2, 3)$ ，或者第4组数据 $(4, 3)$ ，以获取最小的 s 。删除第4组数据时，拟合直线的示意图如下。



AUTHOR: Inf

Problem I. 节日快乐吗

题目描述

某个地方逢节日放假时，会把中间的周末也算作因节日而放的假。这导致有的同学不满，他们认为这样相当于少了一些假期；但有时候足够长的假期也会让大家忘记这一点。现在告诉你某学期每次节假日的放假情况，请你判断每次节假日快不快乐。

输入格式

第一行一个数字 N ($1 \leq N \leq 20$)，表示总共有 N 个节假日。

接下来 N 行，每行六个数字 $Y_1, M_1, D_1, Y_2, M_2, D_2$ ($2000 < Y_1 \leq Y_2 \leq 5 * 10^8$)，分别表示一次节假日放假的起始日期与结束日期，保证这些日期符合现实日期规则，且结束日期一定大于开始日期。注意起始与结束两天也在放假。**已知：2000年1月1日为周六。输入的日期保证在此日期之后。**

60% 数据确保全部使用循环的做法不会超时。

输出格式

共 N 行，每行一个字符串对应每次放假是否快乐。对每次节日放假，若放假总天数**大于等于**其中周末（周末包括周六和周日）天数之和的三倍，则这次节日放假是快乐的，输出 `Happy`；否则输出 `Sad`。

样例输入

```
2
2021 9 2 2021 9 4
2021 9 10 2021 9 11
```

样例输出

```
Happy
Sad
```

样例说明

第一次节日放假，2、3日分别为周四、周五，4日是周六，故放假总天数为3，周末占1天。总放假天数是当中周末天数之和的3倍，故这次放假是快乐的。

第二次节日放假，10日是周五，11日是周六，故放假总天数为2，周末占1天。总天数仅为周末天数的2倍，故这次放假是不快乐的。

Hint

关于闰年：

- 1, 当年份不是整百时，若能被 4 整除则为闰年；否则为平年。
- 2, 当年份是整百时，若能被 400 整除则为闰年；否则为平年。
- 3, 闰年的 2 月有 29 天，平年的 2 月仅有 28 天。

Problem J. 分解质因数

题目描述

任何一个大于 1 的整数 x 都可表示为一些质数的积，称这些质数为 x 的质因数。请你将所有输入的整数表示为它的质因数的幂的乘积的形式：

$$x = a_1^{e_1} \times a_2^{e_2} \times \cdots \times a_n^{e_n}$$

其中：

- 对任意一项 $a_i^{e_i}$ ， a_i 为素数， $e_i > 0$ ，且若 $e_i = 1$ ，则不输出指数部分 ($\wedge e_i$)
- a_1, a_2, \dots, a_n 升序排列
- 等式中没有空格和多余符号

输入格式

多组输入（组数不超过 1000），每行一个整数 x ($2 \leq x \leq 10^9$)

输出格式

对每一个 x ，输出一行相应的等式

样例输入

```
360
19
```

样例输出

```
360=2^3*3^2*5
19=19
```

Hint

- 多组数据的常用输入方式如下：

```
int x;
while (scanf("%d", &x) != EOF) {
    // do something
}
```

- 在大于 1 的自然数中，除了 1 和它本身以外不再有其他因数的自然数为质数，其他则为合数。
- 测评机在一秒内可进行的简单计算大约在 10^9 数量级，如果你获得了 *TLE*，不妨想一下如何利用“**对任意一个合数 x ，总存在一个因数 y 满足 $2 \leq y \leq \sqrt{x}$** ”这个性质。

AUTHOR: tough1