

C2 - Statement

A 这里是 ASCII!

题目描述

请你根据输入的ASCII码，输出其对应的字符（保证为可见字符）。

输入

多组数据输入

每组数据输入一行一个整数 n ，满足 $32 \leq n \leq 126$

输出

对于每组数据，输出一行一个字符，为 n 作为 ASCII 码对应的字符。

输入样例

```
51
114
```

输出样例

```
3
r
```

Hint 1

对于多组（不定组）数据输入，我们可以这样做：

```
while(scanf("%d", &n) != EOF)//读入多组数据，每次输入一个整数n
{
    //补全你的代码
}
```

在Windows系统上，同学们本地调试的时候，可以按下ctrl+z以结束输入。

Hint 2

一定要写95个 `if` 吗？试试下面的代码，然后想一想本题应该怎么做？

```
#include <stdio.h>
int main()
{
    char c;
    scanf("%c", &c);
    printf("%d", (int)c); //强制类型转换
    return 0;
}
```

Author:某咸鱼同学

B 这里是 BUAA

题目描述

czx 作为刚来到 BUAA 的新同学，不小心将学校的简称记错了！他将 BUAA 记成了 CVBB，从而在一些字符中将它拼错了。

为了避免遭受批评，现在他需要将 n 个字符中打错了的 CVBB 删除。并且，czx 是一个很懒惰的人，他为了偷懒，决定无差别的删去所有的字母 C，V，B。

请你帮它完成这项工作吧！

输入

第一行一个正整数 n ，保证 $1 \leq n \leq 500$ 。

第二行 n 个大写字母。

输出

一行，表示 n 个字符删去字母 C，V，B 后得到的结果

输入样例

```
26
ABCDEFGHIJKLMNOPQRSTUVWXYZ
```

输出样例

```
ADEFGHIJKLMNOPQRSTUVWXYZ
```

Hint

换行符也是一个字符，进行第二行字符的读入之前，需要处理第一行末尾的换行符。

可以采用如下代码：

```
scanf("%d", &n);
getchar(); //读入第一行末的换行符
//继续读入第二行的字符
```

C 小亮的乱码书信

题目背景

小亮想要给他的女神写一封信。但他想让这封信足够与众不同。于是他将信的内容进行翻转变成乱码，再将这串乱码写在纸上送给女神。

题目描述

给定一行字符串 S （只包含小写字母、数字和空格），每一个字符的翻转规则如下：

- 如果该字符是数字，请将 0 变成 9，1 变成 8，2 变成 7，以此类推。
- 如果该字符是小写字母，请将 a 转换成 z ， b 转换成 y ，以此类推。
- 空格字符保持不变。

请输出翻转后的字符串。

输入

一行，仅由小写字母、数字和空格组成的字符串 S ，字符串长度 ≤ 100 。

输出

一行翻转后的字符串。

输入样例

```
i love buaa 1314
```

输出样例

```
r olev yfzz 8685
```

Hint

读取单个字符并赋值给 c ，可以使用 `scanf("%c", &c);` 或 `c = getchar();`。

每次循环读取单个字符，直至输入结束（遇到文件尾）的参考代码：

```
char c;
while(scanf("%c", &c) != EOF)
{
    //补充你的代码
}
```

或者：

```
char c;
while ((c = getchar()) != EOF)
{
    //补充你的代码
}
```

D violet的回寝之路

题目描述

violet 最近每天都和他的小伙伴在学校附近的麦当劳打建模大赛，经常很晚才开始回寝室。从麦当劳走到学校的东门和北门都同样需要 10 分钟，北门距离寝室很近（时间可忽略），但只有在每天的 6 : 30 到 22 : 30 才会开放，东门虽然全天开放，但是骑车到寝室要 10 分钟，疲惫的 violet 想尽可能早点回宿舍休息，已知出发时间，聪明的你一定帮助他选择最合适的校门！（violet 不想骑车，所以两个门时间相同时会选择北门）

输入

第一个数为数据组数 n ，保证 $1 \leq n \leq 20$

接下来 n 行，每行两个整数，分别代表当前的小时和分钟

输出

对于每组数据，输出一行

若选择东门，输出 E

若选择北门，输出 N

输入样例

```
3
21 0
6 0
22 20
```

输出样例

```
N
E
N
```

##Hint

如果能更快到达寝室的话，violet不介意在校门口稍等一会~

Author: violet

E violet 打保龄球

题目描述

刚刚打完国赛的 violet 和 Lilsio 打算一起去打保龄球放松一下，保龄球馆的规则如下：

保龄球比赛由 n 轮组成，每轮的瓶子数均为 10。

假如玩家在第 i 轮击中了 x_i 个瓶子。玩家第 i 轮的得分为：

- 如果玩家在该轮的前两轮中的任何一轮中击中了10个瓶子，则为 $2x_i$ 。
- 否则，为 x_i 。

玩家的得分是其 n 轮得分的总和。最终得分高者获胜，若得分相同则为平局。

输入

共三行

第一行为比赛轮次 n ，保证 $1 \leq n \leq 10^6$

第二行为 violet 在这 n 轮中击中的瓶子数量 x_1, x_2, \dots, x_n ，保证 $0 \leq x_i \leq 10$

第三行为 Lilsio 在这 n 轮中击中的瓶子数量 y_1, y_2, \dots, y_n ，保证 $0 \leq y_i \leq 10$

输出

共两行

第一行为两个整数，分别代表 violet 和 Lilsio 的得分

第二行为一行字符串，若非平局，输出胜利者 `violet` 或者 `Lilsio`；若平局，输出 `Let's play again!`

输入样例1

```
3
10 5 8
5 8 10
```

输出样例1

```
36 23
violet
```

输入样例2

```
5
7 8 9 10 8
8 10 6 6 8
```

输出样例2

```
50 50
Let's play again!
```

样例解释

对于样例1，共三轮。

violet 在第一轮击中 10 个瓶子，得到 10 分，第二轮击中 5 个瓶子，但由于上一轮击中了 10 个瓶子，所以得到 $2 \times 5 = 10$ 分，第三轮击中 8 个瓶子，由于前两轮中击中过10个瓶子，得到 $2 \times 8 = 16$ 分，3轮结束后 violet 共得到 $10 + 10 + 16 = 36$ 分。

Lilsio 在三轮中分别击中 5, 8, 10 个瓶子，得分为 $5 + 8 + 10 = 23$ 分。

由于 $36 > 23$ ，violet 获胜。

对于样例2，共5轮。

violet 的总得分为 $7 + 8 + 9 + 10 + 8 \times 2 = 50$ 分。

Lilsio 的总得分为 $8 + 10 + 6 \times 2 + 6 \times 2 + 8 = 50$ 分。

由于 $50 == 50$ ，平局，输出 `Let's play again!`

*Author : violet *

F 向量计算器

题目描述

GYCY为帮助懒惰的DeNeRATE线性代数不挂科，决定帮他制作一个向量计算器，给定三个向量 $\vec{r}_1 = (a_1, a_2, a_3)$, $\vec{r}_2 = (b_1, b_2, b_3)$, $\vec{r}_3 = (c_1, c_2, c_3)$ ，它能进行**6种**向量计算：

1. 向量加法： $\vec{r}_1 + \vec{r}_2 = (a_1 + b_1, a_2 + b_2, a_3 + b_3)$
2. 向量减法： $\vec{r}_1 - \vec{r}_2 = (a_1 - b_1, a_2 - b_2, a_3 - b_3)$
3. 向量内积： $\vec{r}_1 \cdot \vec{r}_2 = a_1 \times b_1 + a_2 \times b_2 + a_3 \times b_3$
4. 向量外积： $\vec{r}_1 \times \vec{r}_2 = (a_2b_3 - a_3b_2, a_3b_1 - a_1b_3, a_1b_2 - a_2b_1)$
5. 向量混合积： $(\vec{r}_1 \times \vec{r}_2) \cdot \vec{r}_3 = a_1b_2c_3 + a_2b_3c_1 + a_3b_1c_2 - a_3b_2c_1 - a_2b_1c_3 - a_1b_3c_2$
6. 向量双重外积： $(\vec{r}_1 \times \vec{r}_2) \times \vec{r}_3 = (\vec{r}_1 \cdot \vec{r}_3) \cdot \vec{r}_2 - (\vec{r}_2 \cdot \vec{r}_3) \cdot \vec{r}_1$

DeNeRATE非常想知道结果，所以你只有**1秒**的时间。

输入

共三行，第一行三个数 a_1, a_2, a_3

第二行三个数 b_1, b_2, b_3

第三行三个数 c_1, c_2, c_3

输入和输出保证为 `long long int` 范围内整数，并且保证整个运算过程不会超出 `long long int` 范围

输出

输出六行，第一行 $\vec{r}_1 + \vec{r}_2$ 结果

第二行 $\vec{r}_3 - \vec{r}_2$ 结果

第三行 $\vec{r}_3 \cdot \vec{r}_2$ 结果

第四行 $\vec{r}_1 \times \vec{r}_2$ 结果

第五行 $(\vec{r}_1 \times \vec{r}_3) \cdot \vec{r}_2$ 结果

第六行 $(\vec{r}_3 \times \vec{r}_2) \times \vec{r}_1$ 结果

输入样例

```
59 -93 36
92 70 52
31 87 49
```

输出样例

```
151 -23 88
-61 17 -3
11490
-7356 244 12686
-414806
-438306 -383590 -272606
```

Author: GYCY

G 解方程 2023

题目描述

给定一个方程 $ax^2 + bx + c = 0$ ，在实数范围内求根。

输入

一行，三个空格分隔的整数 a, b, c ， $-100 \leq a, b, c \leq 100$ 。

输出

若方程无实根，输出 `No real root`；

若方程有一个实根，输出唯一实根（多个值相同的实根视为同一个实根），保留两位小数；

若方程有不同实根，按从小到大输出，保留两位小数；

若方程有无穷多解，输出 `infinite solutions`。

输入样例

```
1 2 -3
```

输出样例

```
-3.00 1.00
```

HINT

考虑 a, b, c 可能为 0 的情况。

可能会用到 `math.h` 库中的 `sqrt` 函数，该函数的作用是求一个 `double` 类型变量的算术平方根。

H Gino 的 rks

题目描述

Gino 特别喜欢演奏，可是他并不是每个谱子都能演奏的很完美。对于他演奏的每一个完美的音符，他会得到「perfect」的评价。对于每一个演奏偏早或偏晚的音符，他会得到「good」的评价，对于每一个演奏过早的音符，他会得到「bad」的评价，对于每一个演奏过晚或没有演奏的音符，他会得到「miss」的评价。

对于单个音符，得到 perfect 能够获得 100% 的准度，得到 good 能够获得 65% 的准度，而得到 bad 或 miss 会丢失该音符的全部准度。而一整个谱子的准度 acc 就是该谱子每个音符的准度的平均值。

可是，每一个谱面的「定数」不是相同的。「定数」是一个一位小数，反映一个谱面的难度，定数越高，难度越高。为了便利地比较在不同定数 m 的谱面中 Gino 的综合得分 rks ，我们使用下面这个公式：

$$rks = \begin{cases} \left(\frac{acc - 55\%}{45\%} \right)^2 \cdot m, & acc \geq 70\% \\ 0, & acc < 70\% \end{cases}$$

请你根据 Gino 每个谱面的完成情况算出单曲的 acc 与 rks 吧！

输入

多组数据输入，每组数据输入一行

每行有一个浮点数 m 与四个整数 a, b, c, d ，表示在定数为 m 的谱面中，perfect, good, bad, miss 的数量分别为 a, b, c, d ， $1.0 \leq m \leq 16.8$ ， $0 \leq a, b, c, d \leq 2330$ ，保证谱面至少有一个音符，且计算得到的 $acc > 1\%$ ，数据组数不超过 10^5 。

输出

对于每组数据，输出一行两个浮点数，分别代表该谱面的 acc 与 rks 。 acc 以百分数形式输出，保留四位有效数字； rks 保留小数点后四位输出，两个数之间用一个空格隔开。

输入样例

```
14.6 707 59 3 18
15.2 1158 24 0 4
15.6 1204 8 0 1
16.1 1308 16 0 1
16.8 1100 172 13 15
11.4 51 4 1919 810
```

输出样例

```
94.71% 11.3679
98.95% 14.5019
99.69% 15.3836
99.50% 15.7455
93.22% 12.1160
1.925% 0.0000
```

Hint

1. 有效数字的定义：

从一个数的左边第一个非 0 数字起，到末位数字止，所有的数字都是这个数的有效数字。例如 0.28 有两位有效数字，2.80 则有三位有效数字。将 2.8014 保留四位有效数字，就是将它四舍五入保留三位小数，即 2.801。

2. 多组数据输入可以参考以下代码

```
double m;
int a, b, c, d;
while (scanf("%lf%d%d%d", &m, &a, &b, &c, &d) != EOF){
    // do something
}
```

可以每输入一组数据就输出一组结果的啊！别傻乎乎的开数组存啊喂 Σ(っ °Д °;)っ

3. 判断浮点数范围

判断 $x \geq y$ 最好写作 $x > y - \text{eps}$ ，判断 $x == y$ 最好写作 $\text{fabs}(x - y) < \text{eps}$ 。

其中 `eps` 是一个小量，可以取 `1e-10` 等，`fabs` 函数为 `math.h` 库中的函数，作用是取 `double` 类型变量的绝对值。

Author: Gino

I Permutation!

题目描述

今天出题人表示不想写题目背景！

数列 (p_1, p_2, \dots, p_n) 是集合 $\{1, 2, \dots, n\}$ 的一个排列当且仅当从集合的每个元素都在数列中恰好出现一次。例如, $(3, 1, 4, 2, 5)$ 就 $\{1, 2, 3, 4, 5\}$ 的一个排列。

对于给定的 n , 我们想知道是否存在 $\{1, 2, \dots, n\}$ 的一个排列 (p_1, p_2, \dots, p_n) 使得 $2 \leq |p_i - p_{i+1}| \leq 3$ 对任意的 $1 \leq i < n$ 都成立。

输入

一行, 一个正整数 n 表示排列的长度, 保证 $1 \leq n \leq 10^5$ 。

输出

如果不存在这样的排列, 输出一行一个整数 -1 。

如果存在这样的排列, 输出两行, 第一行一个正整数 n , 接下来一行输出 n 个正整数, 表示满足条件的一个排列。如果存在多个满足条件的排列, 你只需要输出任意一组即可。

输入样例

```
5
```

输出样例

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

Author: supercarrydyc

J czx 与小羊函数

题目描述

对正整数 n , 用 $p(n)$ 表示 n 的各位 **非 0 数字** 的乘积, 如果 n 是个位数, 则 $p(n) = n$ 。例如:
 $p(1) = 1$, $p(100) = 1$, $p(123) = 6$ 。给定 n , 求 $p(1) + p(2) + \dots + p(n)$, **结果对 $10^9 + 7$ 取模**。

输入

一行一个正整数 n , 保证 $1 \leq n < 10^{10^5}$ 。

输出

输出一个正整数, 表示 $p(1) + p(2) + \dots + p(n)$ 对 $10^9 + 7$ 取模得到的结果。

输入样例

3

输出样例

6

样例解释

$p(1) = 1, p(2) = 2, p(3) = 3。$

$p(1) + p(2) + p(3) = 6。$

Hint

可以参考以下代码进行读入和存储 n

```
#include <stdio.h>
#include <string.h>
char s[100005];
int main() {
    scanf("%s", s);
    int len = strlen(s);

    // 补充你的代码

    return 0;
}
```

以上代码中， len 为 n 的位数，`s[0]` 到 `s[len-1]` 存储了 n 的每位数字字符

如：若 $n = 123$ ，则 len 为 3，`s[0]` 为 `'1'`，`s[1]` 为 `'2'`，`s[2]` 为 `'3'`

特别鸣谢 *pth* 和小羊对本题的贡献

Author: czx

- End -
