

C2 - Statement

A ASCII 2024

题目描述

ASCII（美国标准信息交换码）是最常用的字符编码标准之一，其中包括了可见字符（如字母、数字和符号）以及一些控制字符（如换行符、制表符等）。控制字符虽然对程序员和计算机系统有重要意义，但它们在普通显示设备上是不可见的。计算机中的字符就是按照 ASCII 对应到 0 到 127 这 128 个整数，以整数形式存储在计算机中的。

现在，你需要实现一个程序，判断输入的 ASCII 码对应的是不是一个可见字符。

输入

一个正整数 n ，保证 $0 \leq n \leq 127$ 。

输出

如果 n 作为 ASCII 对应的是一个可见字符，则输出该字符。

如果 n 作为 ASCII 对应的是一个不可见字符，则输出 `Invisible Character!`。

输入样例 1

```
97
```

输出样例 1

```
a
```

输入样例 2

```
15
```

输出样例 2

```
Invisible Character!
```

Hint

可见字符对应的 ASCII 范围是 32 到 126（包括 32 和 126）。

你可以试着运行如下代码

```
#include <stdio.h>

int main() {
    printf("%c", 97);

    return 0;
}
```

B 不会算成绩的 shtog

题目描述

shtog 是北航的一名本科生，修过很多课程却不会算数。现在他想知道自己的算数平均分和加权平均分分别是多少，所以他便拿着成绩单找到了聪明的你。

输入

第一行一个整数 n ($1 \leq n \leq 100$)，表示参与计算成绩的课程数量。

接下来 n 行，每行 1 个整数 a ($0 \leq a \leq 100$) 和一个小数 f ($0.0 < f \leq 6.0$ ，均以保留一位小数的形式给出)，两者以一个空格间隔，每行给出了 shtog 的一门课的课程成绩 a 和学分 f 。

输出

输出两个浮点数 f_1 , f_2 （保留两位小数），分别表示 shtog 的算数平均成绩和加权平均成绩，二者以一个空格间隔。

输入样例

```
2
94 2.0
100 6.0
```

输出样例

```
97.00 98.50
```

HINT

加权平均分的计算公式为：

$$\frac{\sum_{i=1}^n (a_i \times f_i)}{\sum_{i=1}^n f_i}$$

算数平均分的计算公式为：

$$\frac{\sum_{i=1}^n a_i}{n}$$

保留两位小数输出浮点数可以用 `printf("%.2f", f1);`

C 贪吃蛇 (easy version)

题目描述

有一条饥饿的贪吃蛇，初始长度为 x 。在它的行进过程中，会先后遇到 n 个食物 a_1, a_2, \dots, a_n 。 $a_i \geq 0$ 时，贪吃蛇吃掉这个食物会变长 a_i 。 $a_i < 0$ 时，贪吃蛇吃掉这个食物会变短 $|a_i|$ 。如果贪吃蛇吃掉某个食物后，长度会缩短至 0 甚至缩短至负数，那么它就不会吃掉这个食物，**否则它一定会吃掉这个食物**。

注意，贪吃蛇不能往回走，也就是说，它无法返回吃那些原来没吃掉的食物。

请输出贪吃蛇总共能吃掉多少个食物，以及它的最终长度。

输入

第一行两个整数 n, x ， $1 \leq n \leq 10^5, 1 \leq x \leq 10^9$ 。

第二行 n 个整数 a_1, a_2, \dots, a_n ， $-10^9 \leq a_i \leq 10^9$ ，代表吃掉食物后贪吃蛇长度的变化量。

输出

输出两个整数，分别代表贪吃蛇吃掉的食物个数以及它的最终长度，两者之间用一个空格隔开。

输入样例

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

输出样例

```
4 7
```

样例解释

贪吃蛇吃掉第一、二个食物后，长度为 4，此时它无法吃掉第三个食物，否则它的长度就会变成 0。因此它能吃 4 个食物，最终长度为 7。

Hint

请仔细看本题的数据范围，估计一下贪吃蛇的最大长度。可能需要使用 `long long` 哦！

D CVB

题目描述

cvbb 是著名高校，但是不知道为什么，cvbb 的某个横幅上只剩下了 cvb。

为了补上最后一个 `b`，Xhesica 将会给你一个字符串，你能告诉他字符串内是否含有字符 `b` 吗？

输入

两行，第一行一个正整数 n ，标识字符串的长度,满足 $1 \leq n \leq 20$ 。

第二行一个字符串，可能包括所有的可见字符。

本题目保证换行符仅有 `\n`

输出

如果此字符串含有字符 `b`，输出 `CVBB`；

如果没有，输出 `CVB`。

输入样例

```
3
abc
```

输出样例

```
CVBB
```

样例解释

`abc` 中存在字符 `b`，所以输出 `CVBB`。

Hint

考虑到第一行结尾有一个换行符 `\n`，你可以使用

```
scanf("%d",&n);  
getchar();
```

来解决换行符的问题。

E 小懒獭与距离

题目描述

小懒獭终于学完了高等代数中的空间解析几何部分！

在学习空间解析几何的过程中，小懒獭学到了如何求解三维空间上的点到平面的距离。

三维空间上的点 (x_0, y_0, z_0) 到平面 $Ax + By + Cz + D = 0$ （其中 A, B 和 C 不全为 0）的距离 d 为：

$$d = \frac{|Ax_0 + By_0 + Cz_0 + D|}{\sqrt{A^2 + B^2 + C^2}}$$

现在，小懒獭想实现一个程序自动去计算三维空间上的点到平面的距离，你能帮帮小懒獭吗？

输入

第一行输入四个一位小数，分别表示平面方程中的 A, B, C 和 D 系数，保证 A, B, C 和 D 的值在 $[-100, 100]$ 范围内，且 A, B, C 不同时为 0。

第二行输入点的坐标 (x_0, y_0, z_0) ，其中 x_0, y_0 和 z_0 也是一位小数，且它们的值在 $[-100, 100]$ 范围内。输入格式为没有空格的形式，直接使用括号和逗号分隔坐标值。

输出

点到平面的距离，保留到小数点后四位。

输入样例

```
1.0 1.0 1.0 1.0  
(1.0,1.0,1.0)
```

输出样例

Hint

输入可以参考下面一个示例：

◆从键盘读入圆的信息，包括圆心和半径（double类型），输出圆的周长和面积（PI值取3.14），输入格式为：

- ✓第一行半径：r
- ✓第二行圆心：(x, y)
- ✓样例数据：
 - 2
 - (3.5, 4.8)

```
#include <stdio.h>
int main()
{
    double x, y, r;
    double PI= 3.14;

    scanf("%lf", &r); //读入半径信息
    //(前面一定要有空格，以便处理半径输入后剩余的回车
    scanf(" (%lf,%lf)", &x, &y);

    printf("%6.2f\n%6.2f", 2*PI*r, PI*r*r);
}
```

提示：除%c外，其它格式控制符自动忽略掉输入前面的空格；但其他的普通字符则必须原样输入，否则出错！

例中3.5、4.3数字前面空格不需要处理，但‘(’、‘,’这种格式串前面空白字符则必须单独处理

对浮点数取绝对值可以使用 `<math.h>` 中的 `fabs(double f)`；开根号可以使用 `<math.h>` 中的 `sqrt(double f)`

`fabs(double f)` 可以对输入的数取绝对值后返回一个浮点数，比如尝试 `printf("%lf", fabs(-2 * 5))`；`sqrt(double f)` 可以对输入的数开根号后返回一个浮点数，比如尝试 `printf("%lf", sqrt(3.0))`；

F 检查 AI 代码

题目描述

「James Eugene Raynor」是今年程序设计基础课程的助教。上机赛结束后，他想要检查同学们的代码是不是由人工智能生成的。

现在「Raynor」手中有 n 段同学们的代码，从 1 到 n 编号。对于每段代码，首先他会用特别的程序粗略排查这段代码是否由人工智能生成。如果这个程序认为这个代码由人工智能生成的概率很大，那么这段代码就会被标记为可疑代码。程序运行结束后，「Raynor」会喊上其他两名助教，总共三人按顺序评估可疑代码是否由人工智能生成。如果三人中至少两人认为代码是由人工智能生成的，那么就认定这段代码使用了人工智能。

现在，我们知道了程序对每段代码的评估，和三名助教对可疑代码的评估，请你输出，这 n 段代码中，哪些代码被认定为人工智能生成的。

输入

第一行，一个整数 n ($1 \leq n \leq 1000$)，表示代码段数；

第二行， n 个由空格分开的整数，其中第 i 个数表示程序对第 i 篇代码的评估；如果这个数是 1，表示程序认为这段代码属于可疑代码；如果这个数是 0，表示程序认为这段代码不属于可疑代码。

接下来若干行（行数等同于可疑代码的数量），每行三个由空格分开的整数，表示三名助教对这段可疑代码的评估。某个整数为 1，代表这个助教认为这段代码是由人工智能生成的，否则这个整数为 0。

我们可以保证，编号小的可疑代码总是先于编号大的可疑代码被助教检查。

输出

一行，以 `Answer:` 为开头，后接若干个由空格隔开的、严格单调递增的整数，表示被认定为人工智能生成的代码的编号。冒号与第一个数之间无空格。

输入样例

```
5
0 1 0 1 1
0 1 1
0 1 0
1 0 1
```

输出样例

```
Answer: 2 5
```

样例解释

第 2, 4, 5 段代码是可疑代码，但第 4 段只有一个助教认为是人工智能生成的，因此不被认定为人工智能生成的。

请同学们诚信考试，即使是上机赛和练习赛，也不允许使用人工智能生成代码！

G ddz 饮马

题目描述

在一个平面直角坐标系中， x 轴和 y 轴分别是一条河，ddz 现在骑着他的马在点 (a, b) 处，他现在要前往目标点 (c, d) ，但是马口渴了，需要先去河边喝水再去往目标点。ddz 想知道去往目的地要走的最短路程是多少。（河的宽度可以看成 0，只需要去任意一条河喝水即可）

输入

四个整数 $a, b, c, d (0 < a, b, c, d < 100)$

表示初始位置是 (a, b) ，目标位置是 (c, d)

输出

输出一个浮点数（保留2位小数），表示从初始位置先到河边喝水再前往目标位置的最短路程

输入样例

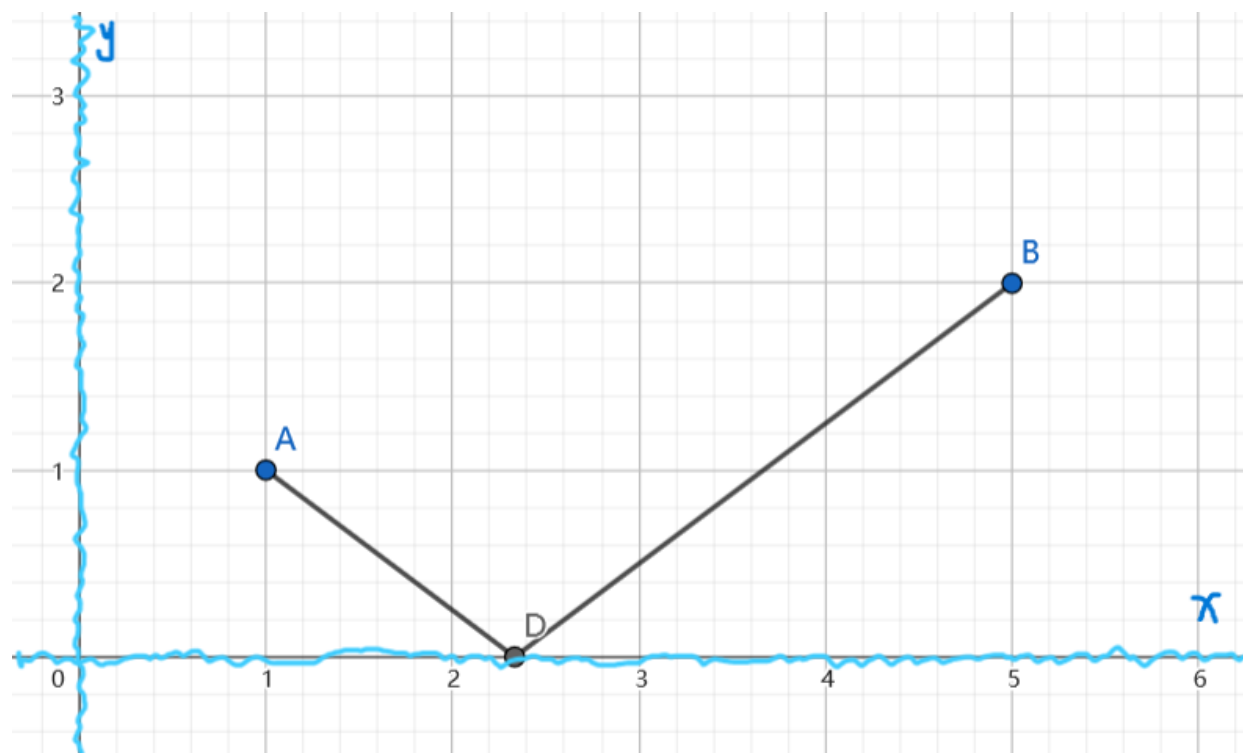
```
1 1 5 2
```

输出样例

```
5.00
```

样例解释

如图所示 DDZ 先从 $(1, 1)$ 去往 $(\frac{7}{3}, 0)$ 喝水，再前往目标点 $(5, 2)$ ，此时总路程最短为 5.00。



H 板凳龙

题目描述

“板凳龙”，又称“盘龙”，是浙闽地区的传统地方民俗文化活动。人们将少则几十条，多则上百条的板凳首尾相连，形成蜿蜒曲折的板凳龙。盘龙时，龙头在前领头，龙身和龙尾相随盘旋，整体呈圆盘状。一般来说，在舞龙队能够自如地盘入和盘出的前提下，盘龙所需要的面积越小、行进速度越快，则观赏性越好。

但是如果盘龙的面积过小，板凳之间的距离过近，就会发生碰撞而导致表演结束。

今年的数模国赛A题要求Xhesica计算舞龙队盘入的终止时刻，使得板凳之间不发生碰撞（即舞龙队不能再继续盘入的时间）。

Xhesica将这些板凳抽象为长方体，现在，他希望你能够帮助他实现一个碰撞检测的算法，算法描述如下：

给出各边平行于坐标轴的长方体的一条对角线上的两个顶点的坐标 $(x_1, y_1, z_1), (x_2, y_2, z_2)$ 和若干检测点 (x'_i, y'_i, z'_i) ，你能确定这些检测点是位于长方体内部，边界还是外部呢？

输入

前两行每行有三个整数 x, y, z ，描述的是此长方体的一条对角线的其中一个顶点的坐标。保证长方体不会退化成点，线段，面。

接下来一行，一个整数 n ，表示检测点的数量。

接下来 n 行，第 i 行三个整数 x'_i, y'_i, z'_i ，表示一个检测点的坐标。

保证 $1 \leq n \leq 1000$ ，所有坐标均在 -1000 到 1000 内。

输出

对于每个检测点，输出一行。

如果此点位于长方体内部，输出 `Inner`

如果此点位于长方体外部，输出 `Outer`

如果此点位于长方体边缘，输出 `Edge`

输入样例

```
0 0 0
2 2 2
3
-1 -1 -1
0 0 0
1 1 1
```

输出样例

```
Outer
Edge
Inner
```

样例解释

给出的三个点分别位于长方体内部，边缘，外部。

I 最大子段和

题目背景

Orch1d 有一串实验数据想要你帮他处理，但由于 ddl 马上就要到了，他希望你能用最快的速度找出数据最“好”的那部分。

题目描述

给出一个长度为 n 的整数列 a ，选出其中**连续且非空**的一段，使得这一段数的和最大。

输入

第一行一个正整数，表示数列的长度 n 。

第二行 n 个整数，第 i 个整数表示数列的第 i 个整数 a_i ($1 \leq i \leq n$)。

输出

输出一行一个整数表示该数列的最大子段和。

输入样例1

```
6
1 1 -4 5 -1 4
```

输出样例1

```
8
```

数据范围

对于30%的数据，保证 $1 \leq n \leq 2 \times 10^3$ 。

对于100%的数据，保证 $1 \leq n \leq 2 \times 10^5, -10^4 \leq a_i \leq 10^4$ 。

样例解释

选择数列中第 4 到第 6 个数，它们的和最大，为 $5 + (-1) + 4 = 8$ 。

J Orch1d 的幸运数字

题目说明

本题是给有编程基础的同学练手的，编程初学者请忽略；暂不提供本题的题解报告和答疑，有兴趣的同学自行研究。

题目背景

大班破冰活动即将开始，身为文体委员的 Orch1d 决定让大家抽签表演节目，究竟哪些同学抽到的是 Orch1d 的幸运数字呢？

题目描述

如果把一个用十进制表示的数位的奇数位和偶数位分别相加得到的和相同，Orch1d 就认为它是一个幸运的数字。例如：

12345 奇数位相加 $1 + 3 + 5 = 9$ 、偶数位相加 $2 + 4 = 6$ ，因此 12345 不是幸运数字；

2332 奇数位相加 $2 + 3 = 5$ 、偶数位相加 $3 + 2 = 5$ ，因此 2332 是幸运数字。

对于给定的 a 和 b ，请你求出 $a, a + 1, a + 2, \dots, b$ 中幸运数字的数量。

输入

输入数据仅一行，包含空格分隔的两个整数 a 和 b 。

对于 40% 的数据， $1 \leq a \leq b \leq 10^6$ 。

对于 100% 的数据， $1 \leq a \leq b \leq 10^{18}$ 。

输出

输出一个整数，代表 $a, a + 1, a + 2, \dots, b$ 中幸运数字的数量。

输入样例 1

```
1 100
```

输出样例 1

9

输入样例 2

114 514

输出样例 2

29

HINT

你应该使用 `long long` 数据类型来输入 a, b 的值。

```
#include <stdio.h>
int main() {
    long long a, b, ans;
    scanf("%lld%lld", &a, &b);
    //下面填写相应的处理代码
    //如果最后答案也是 long long 类型，那么请这样输出
    printf("%lld\n", ans);
    return 0;
}
```