

C6 - 2021级程序设计基础第六次上机

A - 正确率

题目描述

助教想计算一道oj上机题的“正确率”。

定义一道上机题的“正确率”为“答案正确的提交数量与正常编译的提交数量之比”，即：

$$\text{正确率} = \frac{AC\text{的提交数量}}{\text{非}CE\text{的提交数量}}$$

想必聪明的你一定知道如何编程求解本题的“正确率”~

输入描述

多行输入，每行代表一个提交记录的评测结果，以大写字母组成的字符串表示。只会有 AC CE PE WA TLE MLE REG OE 这几种字符串。所给数据保证至少有一个非 CE 的提交。

输出描述

一行一个小数，表示正确率。保留3位小数。

样例输入

```
AC
CE
PE
WA
TLE
MLE
REG
OE
```

样例输出

```
0.143
```

样例说明

非CE的提交有7个，AC的提交有1个，故正确率为 $\frac{1}{7} \approx 0.143$ 。

数据范围

提交记录个数在100以内。

HINT

使用如下代码,每次循环读入一行不带空格的字符串,存储在字符数组 `s` 中。`s[0]` 为字符串的第一个字符,可以思考一下有什么用处。

```
char s[5];
while(scanf("%s",s) != EOF)
{
    //do something...
}
//...
```

AUTHOR: inf

B - Monica的策划

题目描述

*Monica*是游戏《陈叙社稷OL》的游戏策划,为了评估玩家们对最新推出的活动的反馈,*Monica*准备查看一些玩家的活跃度。

现在给出 n 个玩家的活跃度(保证没有两个相同的活跃度)以及 k 个询问,分别为 q_1, q_2, \dots, q_k , 你需要回答活跃度第 q_i 大的玩家的活跃度是多少。

输入描述

输入第一行两个整数 n, k 。

接下来一行 n 个整数,表示每个玩家的活跃度 a_i 。

接下来一行 k 个整数,表示 k 个询问 q_i 。

输出描述

输出 k 行,每行一个整数,表示活跃度第 q_i 大的玩家的活跃度。

样例输入1

```
5 2
7 2 11 10 13
3 2
```

样例输出1

```
10
11
```

样例输入2

```
8 5
10 7 16 12 2 5 18 1
6 1 3 7 8
```

样例输出2

```
5
18
12
2
1
```

样例解释

第一个样例中，排序后的活跃度为

```
13 11 10 7 2
```

第3大的数是10，第2大的数是11

数据范围

$$1 \leq n, k \leq 2000$$

$$1 \leq a_i \leq 2^{31} - 1$$

$$1 \leq q_i \leq n$$

Hint

排序

AUTHOR: Monica

C - sizeof

题目背景

老师讲 C 语言数组后，比如 `int` 类型的数组求 `sizeof`，`int` 类型占 4 个字节，乘上长度就是数组占的字节数。

现在老师提了个思考题，如果 `int` 这种算基本类型，对于一种“组合类型”，它由 `int`, `char` 这些基本类型组成，占的字节是这些字节数的简单加和，然后问题是给出“组合类型”的数组，计算占用空间即 `sizeof(数组名)` 会返回的值。

题目描述

给定这样一个“组合类型”，由 N 个基本类型组成，每个类型占字节 b_i ，基本类型和对应的字节数如下表所示，“组合类型”数组的长度为 L 。

计算这个“组合类型”占的字节数 $S = \sum_{i=1}^n b_i$ ，然后计算 $S \times L$ 的值并输出。

基本类型表：

基本类型	占用字节
<code>char</code>	1
<code>int</code>	4
<code>double</code>	8
<code>long long</code>	8

输入描述

输入一共 $N + 1$ 行。

第一行两个正整数 N, L

后面 N 行每行一个字符串，代表基本类型（只出现基本类型表里有的类型）

输出描述

一行，数组占字节数。

样例输入

```
2 5
int
long long
```

样例输出

```
60
```

数据范围

$$1 \leq N \leq 20, 1 \leq L \leq 10^3$$

HINT

- $N = 1$ 时就是普通的 `int a[]; char s[]`；这类数组的求 `sizeof(a)` 啥的。
- 判断输入哪个类型时，是不是有比字符串比较更快的方法？
- 针对某些同学：不要想太多，不考虑对齐。
- `gets()` 在某些情况下可能会读进来空行，怎么解决呢？

AUTHOR: Shederay

D - 字符串匹配

题目描述

给定两个字符串 A, B （可能含有空格），请你判断 B 字符串是否在 A 字符串中出现过。

如果出现过，请输出 B 最早在 A 的第几位出现过（位数从0开始计数）。

输入描述

两行，代表两个字符串 A, B 。

输出描述

如果 B 字符串在 A 字符串中出现过，输出 YES，并输出最早出现的位置；否则输出 NO。

样例输入1

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

样例输出1

```
YES 0
```

样例输入2

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

样例输出2

```
NO
```

数据范围

保证 A, B 字符串长度 ≤ 100 。

HINT

本题可以先求出 A, B 字符串的长度，然后枚举 B 最早在 A 的第几位出现过，然后按位比较。如果有某一位不相等，那么说明 B 没有在 A 的这一位出现；否则，若全部相等，则说明 B 在 A 的这一位出现过，输出即可。

另一种方法是使用 `<string.h>` 库中的 `strstr` 函数，它的用法如下：

`strstr(a,b)` 返回 a 字符串中第一次出现 b 字符串的位置，如果未找到则返回 `NULL`。返回值为 `char` 类型的指针变量，所以如果要知道 b 字符串在 a 字符串的第几位首次出现，需要减去 a 数组的地址，即 `strstr(a,b)-a`。

AUTHOR: yzh

E - 矩阵乘法

题目背景

程小设同学刚刚在高代课上学完了矩阵乘法，现在正在做作业,算矩阵乘法。本着能偷懒绝不动手的原则，他决定写一个C程序帮他算矩阵乘法，一劳永逸。

题目描述

给定2个方阵 A_1, A_2 ，求 $A_1 \times A_2$ 。

不了解矩阵的人快去看hint。

输入描述

第一行，一个整数 t ，表示方阵的大小，为 t 行 t 列。

接下来输入 $2 \times t$ 行，每行 t 个整数，表示方阵 A_1, A_2 。

具体来说：第 $1 + (i - 1) \times t + j$ 行对应第 i 个矩阵第 j 行，每行第 k 个整数对应该矩阵该行中第 k 个元素。

输出描述

t 行，每行 t 个整数，表示相乘结果。

样例输入

```
3
1 0 0
0 1 0
0 0 1
0 0 1
0 1 0
1 0 0
```

样例输出

```
0 0 1
0 1 0
1 0 0
```

数据范围

数据保证 $t \leq 10$ ，且计算过程中产生的值不会超过 int 的表示范围。

HINT

设矩阵 A, B, C 都是 t 阶矩阵,且 $A \times B = C$ 。

用 $A[i][j]$ 表示矩阵 A 第 i 行第 j 列的元素。

则有
$$C[i][j] = \sum_{k=1}^t A[i][k] \times B[k][j]$$

同学们可以在课下尝试写c来解线性方程组、求矩阵。真人实测很爽。

AUTHOR: Mentor_D

F - 应急食品

题目背景

韶光抚月，天下人间。

派蒙和旅行者在珊瑚宫遇到了一个奇怪的机关，这个机关是一个 3×3 的九宫格，需要将 $0 \sim 8$ 这些数字依次填进这些格子中，使得九宫格每一行的数字和相等、每一列的数字和相等、对角线的数字和相等。

题目描述

本来派蒙觉得获得了珍贵的宝箱是一件快乐的事，但现在旅行者给派蒙出了一个难题：把这个九宫格变成 9×9 的数独棋盘，也就是一共 9 个 3×3 九宫格，形成三行三列的排布，**每行都将 $1, 2, 3, \dots, 9$ 这些数字填入格子中，一个数字在一行或者一列或者一个九宫格里只能用一次，仍然使得每一个 3×3 的九宫格的数字之和相等。**

注意这里九宫格是指没有重合的 9 个 3×3 的九宫格。

比如下面这个棋盘：

4	1	3	8	6	2	9	5	7
6	2	7	5	1	9	8	4	3
5	9	8	4	7	3	2	6	1
7	4	6	1	5	8	3	9	2
8	3	9	7	2	6	5	1	4
1	5	2	9	3	4	6	7	8
3	6	1	2	9	7	4	8	5
9	8	5	3	4	1	7	2	6
2	7	4	6	8	5	1	3	9

这就是一个已经填好并且正确的棋盘。

在旅行者拿着衔海珠皇的照看下，派蒙把这些数字填进了九宫格，派蒙求你帮忙看看她填的对不对。

输入描述

共九行，每行 9 个数字，表示派蒙已经填好的棋盘，一行中相邻两个数字用一个空格隔开。

输出描述

如果这个棋盘是正确的，输出 `Cheers! o(^_^)o`；

否则输出 `Emergency food! ~>_<~`。

样例输入1

```
4 1 3 8 6 2 9 5 7
6 2 7 5 1 9 8 4 3
5 9 8 4 7 3 2 6 1
7 4 6 1 5 8 3 9 2
8 3 9 7 2 6 5 1 4
1 5 2 9 3 4 6 7 8
3 6 1 2 9 7 4 8 5
9 8 5 3 4 1 7 2 6
2 7 4 6 8 5 1 3 9
```

样例输出1

```
cheers! o(^_^)o
```

样例输入2

```
4 1 3 8 6 2 9 5 7
6 2 7 5 1 9 8 4 3
5 9 8 4 7 3 2 6 1
7 4 6 1 5 8 3 9 2
8 3 9 7 2 6 5 1 4
1 5 2 9 3 4 6 7 8
3 6 1 2 9 7 5 8 4
9 8 5 3 4 1 6 2 7
2 7 4 6 8 5 1 3 9
```

样例输出2

```
Emergency food! ~>_<~
```

Hint

派蒙不知道自己到底都填了哪些数字，所以也就不知道到底都犯了什么错误，可能会在某一行中把某一个数字填了两次哦~

但是但是，派蒙可以肯定，自己只填了 $1, 2, \dots, 9$ 当中的数字。

AUTHOR: Stockholm

G - 二分查找·改

题目背景

在C++的STL内，有两个关于二分查找的函数：`lower_bound()` 和 `upper_bound()`。

STL 全称 `Standard Template Library`，即“标准模板库”，是一个C++特有的包含一些标准数据结构的库。

题目描述

现在，给定一个已经从小到大排好序的数组（编号从 1 开始），你需要实现这两个函数的功能（注意：和 STL 提供的函数功能有所区别）：

`lower_bound()`：返回该数组中第一个大于或等于查找值的编号，如果没有输出 -1。

`upper_bound()`：返回该数组中第一个大于查找值的编号，如果没有输出 -1。

输入描述

第一行两个数字 n, m 分别表示数组的长度，查找次数。

第二行 n 个数字，表示被查找数组里编号为 i 的值 a_i (保证 a_i 单调不减)

接下来 m 行，每行一个数字 e ，表示查找值。

输出描述

对于每一次询问，输出一行两个数字，分别表示 `lower_bound()` 和 `upper_bound()` 的返回值。

样例输入

```
5 7
1 2 2 4 5
0
1
2
3
4
5
6
```

样例输出

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

数据范围

对于 20% 的数据， $1 \leq n, m \leq 10^3$ 。

对于 100% 的数据， $1 \leq n, m \leq 10^6, 0 \leq a_i, e \leq 10^6$ ，保证 a_i 单调不减。

Author: Blore

H - 康威生命游戏

题目描述

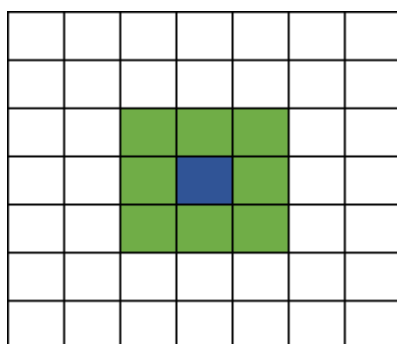
这天陆木缘和杉叶嘉在玩一个叫做“康威生命游戏”的游戏，游戏规则是这样的：

现有一 $n \times n$ 的矩形棋盘记作 M ，棋盘中的每一个格子称作一个“细胞”，有两种取值，即1或0，代表这个细胞的生或死。在确定一个初始状态 M_0 后，按照以下规则演进：

记当前棋盘状态为 M_i ，有：

- 如果一个细胞在 M_i 中是死的
 - 在 M_i 中，它周围有3个存活细胞，则下一个状态 M_{i+1} 该细胞变成存活状态
 - 其它状态则维持现状（死亡）
- 如果一个细胞在 M_i 中是活的
 - 在 M_i 中，当前它周围的存活细胞**低于**两个时，则下一个状态 M_{i+1} 该细胞变成死亡状态
 - 在 M_i 中，当前它周围的存活细胞**高于**三个时，则下一个状态 M_{i+1} 该细胞变成死亡状态
 - 其它状态则维持现状（存活）

上面说的“周围”定义为以其为中心的九宫格所包含的区域，共8个细胞，如下图所示



上图中，所有绿色细胞即是蓝色细胞的“周围”。

陆木缘和杉叶嘉发现，给定一些初始状态时，最终棋盘的演变会进入全部死亡，或者永远静止的情况。对于另一些初始状态，生命游戏的棋盘则会永远运作，例如。

0	0	0	0	0	0	0				0	0	0	0	0	0	0
0	1	2	2	1	0	0				0	1	2	2	1	0	0
0	2	3	3	2	0	0				0	2	3	3	2	0	0
0	2	3	3	2	0	0		→		0	2	3	3	2	0	0
0	1	2	2	1	0	0				0	1	2	2	1	0	0
0	0	0	0	0	0	0				0	0	0	0	0	0	0
0	0	0	0	0	0	0				0	0	0	0	0	0	0

上图是一个永远静止的例子，其中绿色代表存活（1），白色代表死亡（0），格子上的数字代表这个格子周围的活细胞数目。

0	0	0	0	0		0	1	1	1	0		0	0	0	0	0
1	2	3	2	1		0	2	1	2	0		1	2	3	2	1
1	1	2	1	1	→	0	3	2	3	0	→	1	1	2	1	1
1	2	3	2	1		0	2	1	2	0		1	2	3	2	1
0	0	0	0	0		0	1	1	1	0		0	0	0	0	0

上图是一个一直循环的例子，为了方便理解，我将即将“获得生命”的死细胞上的数字标成了绿色，将“即将死去”的活细胞上的数字标成了白色。

陆木缘想，有没有一种算法能够判断生命游戏的某个初始状态会不会在有限时间内停止运作（全部死亡或永远静止）呢？

但是杉叶嘉发现自己是笨蛋，解决不了陆木缘提出的问题，于是她转而研究一个更简单的问题，即**给出生命游戏的一个初始状态，从而求得第 k 个状态**。杉叶嘉找到了会编程的你，希望你帮忙解决这个问题

输入描述

第一行为两个字母 n 和 k ，意义如题所述， n 代表棋盘大小， k 表示要求第 k 个状态。（以初始状态为第0个状态）

接下来是一个 $n \times n$ 的矩阵，其中只有0或者1，代表初始状态

输出描述

一个 $n \times n$ 的矩阵，代表第 k 个状态。

样例输入1

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

样例输出1

```
0 0 0 0 0
0 0 1 0 0
0 0 1 0 0
0 0 1 0 0
0 0 0 0 0
```

样例输入2

```
13 20
0 1 0 0 0 0 0 0 0 0 0 0 0
0 0 1 0 0 0 0 0 0 0 0 0 0
1 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0
```

样例输出2

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

数据范围

$$n, k \leq 100$$

HINT

康威(John Horton Conway)生命游戏是在上世纪70年代提出的，关于其的研究和可计算性理论、图灵机、停机问题、哥德尔不完备定理等等都有着千丝万缕的联系。如果各位有兴趣而且有时间的话，可以思考一下陆木缘提出的问题，即有没有一种算法能够判断生命游戏的某个初始状态会不会在有限时间内停止运作（全部死亡或永远静止）呢？

AUTHOR: 梁秋月

I - 拖，就硬拖

题目描述

老师给土方布置了 n 项作业，每一项作业都有对应的**难度**和**作业量**，现将 n 项作业排成一列，写完某一项作业所花的**时间**等于，土方的**精力**乘以排在这项作业前面的所有作业的**难度**之积，再除以当前作业的**作业量**并**向下取整**的结果。

若令精力为 s ，每项作业的难度和作业量为 a_i, b_i ，那么做完第 i 项作业所消耗的时间为：

$$t_i = \lfloor s \times \frac{\prod_{j=1}^{i-1} a_j}{b_i} \rfloor$$

但是土方有一项超能力，可以**同时写每一项作业**，于是他希望尽可能拖到ddl再写，所以请你帮他重新排列作业顺序，并求出完成所有作业的**最短时间**是多少。

注意：作业的排列顺序只是用于计算时间，并不代表写作业的顺序，所有作业都是同时开始写的。

输入描述

第一行，两个整数 n 和 s ，表示作业的项数和土方的精力（精力始终保持不变）

接下来 n 行，每行两个整数 a 和 b ，表示该作业的**难度**和**作业量**

输出描述

一个整数，代表写完所有作业的最短时间

样例输入

```
3 1
2 3
7 4
4 6
```

样例输出

```
2
```

样例说明

按1、2、3这样排列作业，所花时间为2；

按1、3、2这样排列作业，所花时间为2；

按2、1、3这样排列作业，所花时间为2；

按2、3、1这样排列作业，所花时间为9；

按3、1、2这样排列作业，所花时间为2；

按3、2、1这样排列作业，所花时间为9；

因此写完作业所花的最短时间为2。

数据范围

对于 60% 的数据，保证答案在 *int* 范围内

对于 100% 的数据，有 $1 \leq n \leq 10^3, 0 < a, b < 10^5$

AUTHOR: touth1

J - 圈Sheep

题目背景

Lily家里养了很多Little Sheep，Lily觉得家里的Little Sheep太多了，于是想把原来的圆形栅栏分出一些区域，方便管理这些Little Sheep，但Lily因为难以承担体力活并且过于忙碌，找了不靠谱的Sheep来帮忙打桩，晚上回家才发现Sheep根本没有询问打桩的具体位置便随便打了三根桩，Lily不希望在原栅栏区域的外部养Little Sheep，因而Sheep打的桩会浪费大量的拦网，心碎的Lily想去找Sheep理论，很希望你能帮忙计算一下Sheep打的桩最后究竟能圈住多大比例的Little Sheep



题目描述

假定原来的圆形栅栏的中心为坐标原点，Lily计算出了Sheep的三根桩在该坐标系下的三个坐标 $(x_1, y_1), (x_2, y_2), (x_3, y_3)$ ，并给出了她的圆栅栏的半径 R ，Lily想知道如果把当前的三个木桩装上拦网之后形成的三角形区域在原圆形栅栏内部的面积与原圆形栅栏围住的总面积之比为多少

输入描述

第一行，一个正整数 $n(1 \leq n \leq 10^4)$ 表示数据组数

接下来 n 行，每行7个浮点数，分别表示 $R, x_1, y_1, x_2, y_2, x_3, y_3$

输出描述

对每组数据输出两行

先输出一行一个面积比（保留两位小数）和一个整数（0~3）为Sheep打在围栏外的木桩数

如果最终的面积比小于0.20或大于0.80，再输出一行一个字符串 `Damn Sheep, come back tomorrow!`

对于其他情况，再输出一行一个字符串 `well, maybe not too bad`

样例输入

```
1.8380 24.1551 -12.0636 -2.5295 -11.7304 0.2741 -0.9617
1.4356 33.3340 103.7778 -3.6293 188.9652 0.0828 -2.9989
```

样例输出

```
0.05 2
Damn Sheep, come back tomorrow!
0.38 3
well, maybe not too bad
```

数据范围

对每组数据，保证 $-100 \leq R \leq 100, -200 \leq x_m, y_m \leq 200 (1 \leq m \leq 3)$

HINT

要注意好情况分类

Sheep打的木桩不止会出现在圆形栅栏外，也有可能出现在栅栏内~

π 取3.1415926535

AUTHOR: Oh So many sheep