

A 2023 Mid - 诚信考试

时间限制：1000ms 内存限制：65536kb

Where is there dignity unless there is honesty? — Cicero.

没有诚信，何来尊严？——西塞罗

诚信承诺

我承诺：诚信考试，遵守考试规定，自觉维护考场秩序，包括但不限于：

1. 考试全程不使用个人电话，不做与考试无关的事情。
2. 个人独立答题，不通过网络获取答案或提示；除监考老师以外，不与他人进行任何形式的交流。
3. 考试时按监考老师指示进行考试系统的登录，不尝试任何异常登录行为（尝试登录别人的帐号，个人账号在多个设备登录），妥善保管好个人的帐号和密码并保证不透露给他人。

阅读完“诚信承诺”后，请编写程序，输出如下字符串

```
Where is there dignity unless there is honesty? - Cicero  
I promise I won't cheat in the exam.
```

输入样例

无。

输出样例

```
Where is there dignity unless there is honesty? - Cicero  
I promise I won't cheat in the exam.
```



B 2023 Mid - 计算罚时

时间限制：1000ms 内存限制：65536kb

题目描述

期中考试结束之后，dyc 想要统计一下大家的成绩，其中包含一项数据罚时，计算罚时的规则如下：

- 只有通过的题目才计算罚时，不通过的题目罚时为 0 秒。
- OJ 上每道题目的罚时计算如下：

$$\text{单题罚时} = \text{题目第一次 AC 距比赛开始的时间} + 20\text{分钟} \times \text{题目第一次 AC 前的总错误次数}$$

总罚时为各单题罚时之和，**单位为秒**。

已知 dyc **不会在通过这个题之后继续交代码**，现在 dyc 给你了一名学生比赛时每个题目的提交信息，请你帮助 dyc 算出他在期中考试中的总罚时。

请以秒为单位进行计算。

输入

不定行输入，保证输入不超过 15 行，每一行表示一道题目的提交情况，每道题目都不相同。

每行三个由空格隔开的正整数 op, t, n 。其中 $op \in \{0, 1\}$ ，0 表示最终没有通过这个题，1 表示通过了这个题； t 表示这个题最后一次提交距比赛开始的时间，**以秒为单位**，保证 $0 \leq t < 7200$ ； n 表示这个题提交的次数，保证 $1 \leq n \leq 100$ 。

输出

输出一行一个正整数，表示这名学生的总罚时。

输入样例

```
1 370 1
1 670 2
0 2900 10
```

输出样例

```
2240
```

样例解释

$$[370s + 20min \times (1 - 1)] + [670s + 20min \times (2 - 1)] = 2240s$$

Hint

注意分和秒单位之间的转换。

C 2023 Mid - 字符处理

时间限制：1000ms 内存限制：65536kb

题目描述

给定一个长度为 n 的字符串，将其中所有的小写字母变成大写字母，大写字母变成小写字母，同时记录下出现次数最多的字符。

输入

第一行一个正整数 n , $1 \leq n \leq 10000$.

第二行 n 个字符，字符的 ASCII 码只可能在 33 到 126 之间 (包括 33 和 126)。

输出

第一行 n 个字符，对应着原来的 n 个字符：

- 如果原字符是个小写字母，对应字符是其对应的大写字母；
- 如果原字符是个大写字母，对应字符是其对应的小写字母；
- 否则对应字符是原字符。

第二行 1 个字符，表示在**原字符串**中出现次数最多的字符。

第三行 1 个整数，表示在**原字符串**中出现次数最多的字符出现的次数。

保证只有一个出现次数最多的字符！

输入样例

```
30
MitakeMocaLovesBreadVeryMuch~!
```

输出样例

```
mITAKEmOCA1OVESbREADVERYmUCH~!
e
4
```

D 2023 Mid - 最大汉明

时间限制：1000ms 内存限制：65536kb

题目描述

对于两个 `unsigned int` (32 位无符号整型) 范围内的自然数 a, b , 设它们的 32 位二进制表示分别为 $a_{31}a_{30}\dots a_0$ 和 $b_{31}b_{30}\dots b_0$.

a 和 b 的汉明距离 $d(a, b)$ 定义为它们的 32 位二进制表示中有多少位不相同, 即满足 $a_i \neq b_i$ 的 i 的数量。

设 a 与 b 进行按位异或运算得到的结果为 c , 显然, $d(a, b)$ 等于 c 的 32 位二进制表示有多少位是 1.

现给出三个 `unsigned int` 范围内的自然数 x, y, z , 请求出它们之间的最大汉明距离, 即 $\max\{d(x, y), d(x, z), d(y, z)\}$.

输入

不定组数据输入, 保证数据组数不超过 1000.

每组数据输入一行, 三个 `unsigned int` 范围内的自然数, 分别表示 x, y, z .

输出

对于每组数据, 输出一行一个正整数, 表示 x, y, z 之间最大的汉明距离。

输入样例

```
1 2 3
0 7 9
```

输出样例

```
2
3
```

样例解释

对于第一组数据:

1 的 32 位二进制表示为: `00000000000000000000000000000001`

2 的 32 位二进制表示为: `00000000000000000000000000000010`

3 的 32 位二进制表示为: `00000000000000000000000000000011`

$d(1, 2) = 2, d(1, 3) = 1, d(2, 3) = 1$

因此最大汉明距离为 2.

E 2023 Mid - 查询次序

时间限制: 1000ms 内存限制: 65536kb

题目描述

给出 n 个自然数，将它们按照从小到大的顺序进行排序。

现进行若干次查询，输入一个自然数 a ：

- 若 a 在这 n 个自然数中，请输出 a 在这 n 个数中从小到大排序后的次序（如果有相同的数，则次序并列）；
- 若 a 不出现在这 n 个数中，输出 `-1`。

输入

第一行一个自然数 n ，保证 $1 \leq n \leq 100$ 。

第二行 n 个自然数，以一个空格分隔，保证每个数都不超过 n 。

接下来若干行，每行代表一次查询，一个自然数 a ，保证 $a \leq n$ 。

输出

输出若干行，每次查询输出一行，对于要查询的数 a ：

- 若 a 在这 n 个自然数中，请输出 a 在这 n 个数中从小到大排序后的次序（如果有相同的数，则次序并列）；
- 若 a 不出现在这 n 个数中，输出 `-1`。

输入样例

```
10
1 7 5 0 9 5 8 8 2 5
2
3
5
9
```

输出样例

```
3
-1
4
10
```

样例解释

排序之后为 $[0, 1, 2, 5, 5, 5, 7, 8, 8, 9]$ ，其中：

- 2 的次序为 3；
- 没有 3；
- 5 的次序为 4；
- 9 的次序为 10。

Hint

n 个数中有可能有相同的数，相同的数次序并列，比如 $[1, 3, 3, 3, 4, 8]$ 这 6 个数中，3 的次序为 2, 4 的次序为 5。

快把今天学的冒泡排序用起来吧！当然，不用排序也是可以做的。

F 2023 Mid - 信息之熵

时间限制：1000ms 内存限制：65536kb

题目背景

信息熵（information entropy）是信息论的基本概念。描述信息源各可能事件发生的不确定性。20 世纪 40 年代，香农（C.E.Shannon）借鉴了热力学的概念，把信息中排除了冗余后的平均信息量称为“信息熵”，并给出了计算信息熵的数学表达式。

题目描述

假设一个信息源总共可能发出 n 种符号，分别是 $\alpha_1, \alpha_2, \dots, \alpha_n$ ，它们被发出的概率分别为 p_1, p_2, \dots, p_n ，那么这个信息源的信息熵 H 为：

$$H = -\sum_{i=1}^n p_i \log_2(p_i)$$

现有一个信息源在一定时间内发出一些字符，**以频率作为概率**，请你统计一下这个信息源的信息熵。请注意：**只统计大写字母、小写字母和数字，其余的符号和空白字符等应予以忽略。统计时区分大小写。**

输入

输入一些字符，不超过 10000 个。

输出

一个实数，保留四位小数，表示仅统计大小写字母和数字时，该信息源的信息熵。

输入样例

```
BUAA accoding
2023.10.26
```

输出样例

```
3.7842
```

样例解释

样例中各字符的出现频次统计表如下：

字符	频数	频率
0	2	0.1
1	1	0.05
2	3	0.15
3	1	0.05
6	1	0.05
A	2	0.1
B	1	0.05
U	1	0.05
a	1	0.05
c	2	0.1
d	1	0.05
g	1	0.05
i	1	0.05
n	1	0.05
o	1	0.05

Hint

可以使用 `math.h` 库中的 `log2(x)` 函数返回 $\log_2(x)$.

也可以使用 `log(x)` 函数返回 $\ln(x)$, 利用换底公式计算 $\log_2(x)$.

换底公式:

$$\log_a b = \frac{\log_c b}{\log_c a}$$

G 2023 Mid - 七敲五摇

时间限制: 1000ms 内存限制: 65536kb

题目背景

七敲五摇是聚餐时常玩的一种游戏，规则为：一群人围成一圈从 1 开始轮流报数，5 和 7 是「触发数」，当待报的数是 x 时：

- 若 x 是 7 的整数倍或 x 的十进制表示中带有数字 7（如 477、49 等），则敲一下桌子；
- 若 x 是 5 的整数倍或 x 的十进制表示中带有数字 5（如 54、60 等），则摇一下头；
- 若 x 同时满足 5 和 7 的触发条件，即同时满足以上两条（如 770、757 等），则需同时执行前两条的“敲桌子”和“摇头”操作。
- 若 x 不满足任何「触发数」的触发条件（如 1、2 等），则直接报出 x ，不需要执行任何额外操作。

执行错操作或报错数的人要受到 严~峻~的~惩~罚~~

题目描述

现在你正在聚餐中玩一种该游戏的升级版，规则为：一群人围成一圈从 1 开始轮流报数一直报到 n ，设定了 m 个「触发数」，分别为 $a_1, a_2, a_3, \dots, a_m$ 。

当待报的数是 x 时，「触发数」 a_i 的触发条件为： x 是 a_i 的整数倍或 x 的十进制表示中带有数字 a_i 。

请按照 $x = 1, 2, \dots, n$ 的顺序，按如下要求输出对应的内容：

- 若 x 不满足任何「触发数」的触发条件，输出 `count out`；
- 若 x 满足 $k(k > 0)$ 个「触发数」 $a_{i_1}, a_{i_2}, \dots, a_{i_k}(1 \leq i_1 < i_2 < \dots < i_k \leq m)$ 的触发条件，则将**触发数的序号** i_1, i_2, \dots, i_k 在一行中依次连续输出（中间无空格、无任何符号）。

输入

共 2 行：

第一行 2 个整数 n 和 m ，含义与题目描述相同，保证 $1 \leq n \leq 100000, 1 \leq m \leq 10$ 。

第二行 m 个整数，分别为 $a_1, a_2, a_3, \dots, a_m$ ，保证对任意的 a_i ，都有 $0 \leq a_i \leq 9$ ；且当 $i \neq j$ 时，必有 $a_i \neq a_j$ 。

输出

共 n 行，第 i 行请按要求输出待报的数为 i 时对应的内容。

输入样例

```
14 3
4 3 2
```

输出样例

```
count out
3
2
13
count out
23
count out
13
2
3
count out
123
2
13
```


样例解释

三个触发数分别为 4, 3, 2, 其编号分别为 1, 2, 3.

以数字 14 为例, 既是 2 的整数倍, 又带有数字 4, 故第 14 行输出 13, 分别代表 1 号触发数 4 和 3 号触发数 2.

H 2023 Mid - 陈氏定理

时间限制: 1000ms 内存限制: 65536kb

题目背景

哥德巴赫猜想是“世界三大数学猜想”中唯一尚未被证明的猜想, 被誉为是数学界最难证明的猜想之一, 可以被表述为“任一大于 2 的偶数都可写成两个素数之和”。1966 年, 陈景润在《科学通报》上发表了有关哥德巴赫猜想“ $1+2$ ”的证明, 即“任何一个充分大的偶数都可以表示成两个素数的和或者一个素数及一个二次殆素数的和”。1973 年, 陈景润给出了“ $1+2$ ”的详细证明, 同时改进了 1966 年研究的数值结果。是年四月, 中国科学院主办的《中国科学》上, 公开发表了陈景润的论文《**大偶数表为一个素数及一个不超过两个素数的乘积之和**》。这一结果被国际上誉为“**陈氏定理**”。

题目描述

如果一个正整数可以分解为**两个素数的乘积**, 则称之为**半素数**, 如 4, 6, 9, 10, 14, 15, 21, ... 等等。

请你验证陈氏定理: 给出一个大于 2 的偶数 n , 输出所有的算式 $n = a + b$, 要求 $a \leq b$, a, b 中一个是素数, 且另一个是素数或半素数。

对于每个满足要求的算式 $n = a + b$:

- 若 a, b 均为素数, 则输出格式为 `1+1: n=a+b` ;
- 若 a 为素数, b 为半素数, 则输出格式为 `1+2: n=a+b` ;
- 若 a 为半素数, b 为素数, 则输出格式为 `2+1: n=a+b` 。

输入

一行一个大于 2 的偶数 n , 保证 $n \leq 10^5$ 。

输出

输出若干行, 每行输出一个满足要求的算式 $n = a + b$, 按照 a 的升序顺序进行输出。

对于每个满足要求的算式 $n = a + b$:

- 若 a, b 均为素数, 则输出格式为 `1+1: n=a+b` ;
- 若 a 为素数, b 为半素数, 则输出格式为 `1+2: n=a+b` ;
- 若 a 为半素数, b 为素数, 则输出格式为 `2+1: n=a+b` 。

输入样例

28

输出样例

```
1+2: 28=2+26
1+2: 28=3+25
1+1: 28=5+23
1+2: 28=7+21
2+1: 28=9+19
1+1: 28=11+17
1+2: 28=13+15
```

I 2023 Mid - 最速通讯

时间限制：1000ms 内存限制：65536kb

题目描述

在平面直角坐标系中，一束信号从原点 $(0, 0)$ 出发，需要使其到达 (n, m) 。

已知这束信号始终沿着直线传播，而且每当经过一个整点 $(x, y) (x, y \in \mathbb{Z})$ 时就必须改变它的传播方向。

你可以任意指定每一次信号传递的方向，并且信号传递的方向可以改变任意多次，问这束信号最短需要走过的路程是多少？

输入

第一行一个正整数 T ，表示数据的组数。接下来 T 行，每行两个正整数 n, m ，保证 $\Sigma \max(n, m) < 5 \times 10^6$ 。

输出

输出 T 行，每行一个正实数表示最短的路程，你的答案将被视作正确只要其绝对误差或者相对误差小于 10^{-8} 。

输入样例

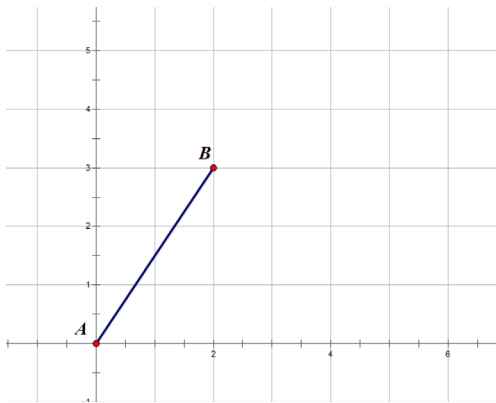
```
2
2 3
2 2
```

输出样例

```
3.605551275463989
3.236067977499790
```

样例解释

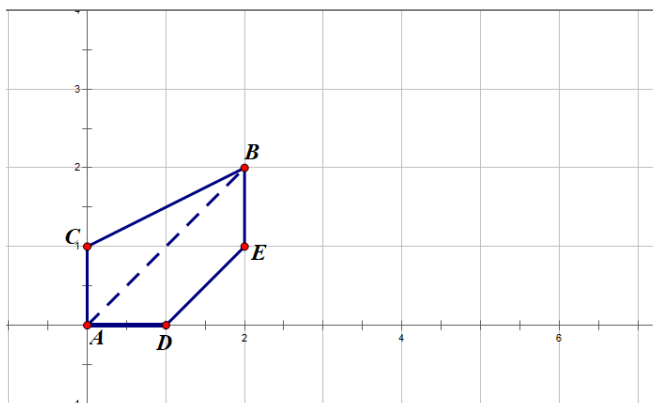
对于第一组样例，可以从原点 A 直接到达终点 B ，最短路程显然为 $\sqrt{13}$ 。



对于第二组样例，由于直接沿直线从 A 到 B 会经过点 $(1, 1)$ ，因此不是一条合法的路径。

可以先到达 $C(0, 1)$ 再沿直线到达 $B(2, 2)$ ，路程为 $\sqrt{5} + 1$ 。而从 A 到达 $D(1, 0)$ ，再到达 $E(2, 1)$ ，最后到达 B 也是一条合法的路径，路程为 $\sqrt{2} + 2$ 。

可以证明由 A 只经过 C 到达 B 的这条路径是路程最短的合法路径，最短路程是 $\sqrt{5} + 1$ 。



J 2023 Mid - 前提条件

时间限制：1000ms 内存限制：65536kb

题目描述

谷香织同学正在完成一系列的任务。在有些时候，完成某些任务必须要以完成其它任务为前提。举个例子，假如她想要完成“烤山芋”，那么在此之前就必须完成“买山芋”和“清洗山芋”。因为她是一个认真细致的人，谷香织同学只能同时做一件任务。

话虽如此，有些任务是无论如何也不能完成的。假如她把钥匙落在了家里，那么“打开家门”的前提是“拿出钥匙”，而且“拿出钥匙”的前提是“打开家门”。这样一来，她既没办法打开家门，也没办法拿出钥匙。

现在告诉你单独完成某项任务所需要花费的时间，以及任务之间的依赖关系，请你求出完成某项任务至少需要花费多少时间。

输入

第一行两个正整数，首先是小于 30 的正整数 n ，表示任务数量；然后是不大于 n 的正整数 s ，表示最终要完成的任务编号。

第二行 n 个不超过 10^4 的正整数 t_1, \dots, t_n ，其中 t_i 表示完成编号为 i 的任务所需的时间。

接下来 n 行, 表示任务之间的依赖关系。在这一部分中的第 i 行共有 $K_i + 1$ 个正整数, 其中第一个数为 K_i , 接下来 K_i 个数是完成编号为 i 的任务之前所必须完成的任务编号。

输出

如果在有限时间内编号为 s 的任务可以被完成, 请输出一行一个整数, 表示完成它最少需要花费多少时间。

如果在有限时间内编号为 s 的任务不可以被完成, 请输出一行字符串: `Dead Lock!!`。

输入样例 1

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

输出样例 1

```
Dead Lock!!
```

输入样例 2

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

输出样例 2

```
14
```

样例解释

在样例 1 中, 任务 1 的前提是任务 2, 任务 2 的前提是任务 3, 任务 3 的前提是任务 4, 但是任务 4 的前提是任务 1. 因此在有限时间内不能完成任务 1.

在样例 2 中, 为了完成任务 1. 需要依次完成任务 4 3 2 1. 尽管任务 5 和 6 互为前提, 但是它们和完成任务 1 无关。

K 2023 Mid - 收集春度

时间限制：1000ms 内存限制：65536kb

题目背景

隆冬。

Hakurei Reimu 穿过幽明结界，正飞行在漫无止境的白玉楼阶梯上，找回幻想乡被夺走的春度。

题目描述

再长的阶梯也有尽头，Reimu 从第 0 阶爬上第 n 阶可抵达冥界中央的白玉楼。为了节省时间，Reimu 使用她的亚空间进行空间传送。当她处于第 i 阶时，她可以传送到 $[i + l, i + r]$ 中的任意一阶上。

春意盎然的冥界充满了春度，每一阶上都有着 a_i 的春度，不过第 0 阶上没有。Reimu 每次传送之后都将获得当前阶梯上的春度。为了恢复樱花盛开的幻想乡，Reimu 需要在抵达白玉楼时收集尽可能多的春度。

Reimu 从第 0 阶开始往上爬，只要她下一次传送后的阶梯编号大于 n 就算抵达白玉楼。

输入

第一行三个正整数 n, l, r , 含义见上；

第二行共 $n + 1$ 个整数，第 i 个数表示第 $i - 1$ 阶的春度 a_{i-1} 。

输出

一个整数，表示抵达白玉楼时的最多春度。

输入样例

```
7 2 2
0 7 4 5 8 1 -5 -5
```

输出样例

```
7
```

数据范围

对于 60% 的数据， $n \leq 10^4$ ；

对于 100% 的数据， $0 < n \leq 2 \times 10^5, |a_i| \leq 10^3, 1 \leq l \leq r \leq n$ 。