华中科技大学计算机科学与技术学院

“你能做的 岂止如此”代码能力提升计划

语言基础

（Part 1）

2024 年 10 月

第二版

前　言

华中科技大学计算机科学与技术学院历史悠久、人才辈出。在武敏颜学姐的倡议、带领下，经过一批优秀的学长学姐的精心工作，学院洛谷题单诞生了。洛谷题单为维持与提升计算机学院本科生代码能力发挥了重要作用。

随后，赵云霏学姐、周辰宇学长接过了维护题单的接力棒，在他们的工作下，学院洛谷团队实现了长期稳定运行，题单不断优化与迭代，获得了历代学生的一致好评。

为了实现洛谷团队良性发展，针对之前题单中存在的一些问题，经谭志虎院长批准，启动了这一轮的题单修订，并第一次以任务书的形式发布试题。

在计算机类 2301 班吴迪同学的建议下，我们决定将洛谷题单正式命名为“‘你能做的 岂止如此’代码能力提升计划”，希望我们的工作，能够帮助一届届的计算机学院本科生增加代码书写量，巩固代码能力，对数据结构与算法有更加深入的了解。

我要感谢参与到本轮题单修订过程中的同学，他们是：计算机科学与技术学院 2022 级本科生杨治、汪文卓、李韦成、刘皓铭、黄子益、方昀昊；计算机科学与技术学院 2023 级本科生万睿朋、聂卓然、刘星佳、彭雨洋。我还要感谢谭志虎院长与王多强教授，没有他们的支持与指导，题单修订工作是不可能顺利完成的。

希望本轮修订的题单，能帮助同学巩固 C 语言基础，同时对 STL 等课程内容之外，但非常实用的语言工具有初步的了解。

由于水平有限，题单难免存在疏漏，欢迎同学们批评指正。

华中科技大学计算机科学与技术学院

刘柏年

2024 年 1 月

小 L 在吃饭（hasmeal）

【题目描述】

众所周知，大学生是一种十分喜欢吃的生物，有一点钱就用来吃饭了。小 L 兜里只剩下 a 元 b 角钱了。然而一碗饭竟然只需要 1 元 9 角（怎么这么便宜），请你求出小 L 最多吃多少碗饭。

【输入格式】

输入一行两个整数，分别表示 a 和 b。

【输出格式】

输出一行一个整数，表示小 L 最多可以吃多少碗饭。

【样例 1 输入】

```
10 3
```

【样例 1 输出】

```
5
```

【子任务】

对于 100% 的测试数据，保证 $0 \leq a \leq 10^4$，$0 \leq b \leq 9$。

小 W 在游泳（swim）

【题目描述】

小 W 大三了，她选了游泳课。

虽然游泳很累，但是小 W 还是坚持游泳，可是她很快难过的发现，自己的力气不够，游泳好累哦。已知小玉第一步能游 2 米，可是随着越来越累，力气越来越小，她接下来的每一步都只能游出上一步距离的 98%。现在小 W 想知道，如果要游到距离 s 米的地方，她需要游多少步呢。

【输入格式】

输入一个实数 s（单位：米），表示要游的目标距离。

【输出格式】

输出一个整数，表示小 W 一共需要游多少步。

【样例 1 输入】

```
4.3
```

【样例 1 输出】

```
3
```

【子任务】

对于 100% 的测试数据，保证 $0 \leq s \leq 100$，s 小数点后最多只有一位。

小 C 在上课（finalscore）

【题目描述】

小 C 在华中科技大学计算机科学与技术学院学习了 C 语言程序设计课程，这门课程的总成绩计算方法是：

- 平时成绩占 20%
- 期中成绩占 30%
- 期末成绩占 50%

小 C 想知道，这门课程自己最终能得到多少分。

【输入格式】

三个非负整数 A, B, C，分别表示小 C 的平时成绩、期中成绩和期末考试成绩。相邻两个数之间用一个空格隔开，三项成绩满分都是 100 分。

【输出格式】

一个整数，即小 C 这门课程的总成绩，满分也是 100 分。

【样例 1 输入】

```
100 100 80
```

【样例 1 输出】

```
90
```

【样例 1 解释】

小 C 的平时成绩是 100 分，期中成绩是 100 分，期末考试成绩是 80 分，总成绩是 $100 \times 20\% + 100 \times 30\% + 80 \times 50\% = 20 + 30 + 40 = 90$。

【样例 2 输入】

```
60 90 80
```

【样例 2 输出】

```
79
```

【样例 2 解释】

小 C 的平时成绩是 60 分，期中成绩是 90 分，期末考试成绩是 80 分，总成绩是 $60 \times 20\% + 90 \times 30\% + 80 \times 50\% = 12 + 27 + 40 = 79$。

【子任务】

对于 100% 的数据，0 A, B, C 100 且 A, B, C 都是 10 的整数倍。

好数（gnum）

【题目描述】

对于任意整数，我们定义如下两个性质。

- 性质 1：是偶数；
- 性质 2：大于 4 且不大于 12。

duoluoluo 喜欢这两个性质同时成立的整数；yazhi 喜欢这至少符合其中一种性质的整数；600years 喜欢刚好有符合其中一个性质的整数；盖亚喜欢不符合这两个性质的整数。现在给出一个整数 x，请问他们是否喜欢这个整数？

【输入格式】

输入一个整数 $x(0 \leq x \leq 10000)$。

【输出格式】

输出这 4 个人是否喜欢这个数字，如果喜欢则输出 `1`，否则输出 `0`，用空格分隔。

输出顺序为：duoluoluo、yazhi、600years、盖亚。

【样例 1 输入】

```
12
```

【样例 1 输出】

```
1 1 0 0
```

排序（sort）

【题目描述】

给出三个整数 $a, b, c (0 \leq a, b, c \leq 100)$，要求把这三个整数从小到大排序。

【输入格式】

输入三个整数 a, b, c，以空格隔开。

【输出格式】

输出一行，三个整数，表示从小到大排序后的结果。

【样例 1 输入】

```
1 14 5
```

【样例 1 输出】

```
1 5 14
```

【样例 2 输入】

```
2 2 2
```

【样例 2 输出】

```
2 2 2
```

谁更短（leauingz）

【题目描述】

LeauingZ 非常懂造题，如果让 LeauingZ 造题，一个题只需要造 3 分钟，如果你自己造题，则一个题需要花 5 分钟。但是如果让 LeauingZ 造题，因为 LeauingZ 很忙，他需要额外花费 11 分钟用来休息。（注意，总共多花费 11 分钟而不是每造一个题休息 11 分钟）

现在要造 n 个题，你可以选择自己造完或者让 LeauingZ 造完所有题。如果你自己造题配置花费的总时间短，请输出 `Local`，否则输出 `Luogu`。

【输入格式】

输入一个正整数 n，表示需要造的题目量。

【输出格式】

输出一行，一个字符串。如果自己造花费的总时间短，请输出 `Local`，否则输出 `Luogu`。

【样例 1 输入】

```
2
```

【样例 1 输出】

```
Local
```

【样例 2 输入】

```
50
```

【样例 2 输出】

```
Luogu
```

【子任务】

数据保证 $1 \leq n \leq 100$。

倍减（double）

【题目描述】

倍增算法是算法领域中常用的一种优化算法，不过在这道题中，你并不需要用到这种算法。

对一个数 x ，规定一次操作为：将 x 除以 2 并向下取整，即将 x 变为 $\lfloor \frac{x}{2} \rfloor$。

请问这个数需要多少次操作 $2x$ 才能变为 1 。

【输入格式】

输入一个正整数 x 。

【输出格式】

输出一个正整数，表示要多少次操作后 $2x$ 变为 1 。

请输出答案 $+1$ 的结果。

【样例 1 输入】

```
101
```

【样例 1 输出】

```
7
```

【子任务】

数据范围：$1 \leq x \leq 10^9$。

有说服力的评分算法（rating）

【题目描述】

数组是一种基础常用的数据类型，不过在这道题中，你并不需要用到这种数据类型。

某评分网站特有的评分算法如下：

该网站上，你可以给出 0 到 10 的评分。统计最终得分时，网站会去掉一个最高分和一个最低分（如果最高或最低不止一个，也只去掉一个），再计算其平均分。

现在有 n 人参与了评分，请问最终得分为多少，精确到 2 位小数。

【输入格式】

第一行输入一个正整数 n。

第二行输入 n 个整数，第 i 个整数表示第 i 个人的评分。

【输出格式】

输出一行一个两位小数，表示最终得分。

【样例 1 输入】

```
5
9 4 6 8 10
```

【样例 1 输出】

```
7.67
```

【子任务】

数据范围：$3 \leq n \leq 1000$。

多项式筛素数（poly）

【题目描述】

线性筛素数是一种常用的素数筛法，不过在这道题中，你并不需要用到这个算法。

给出一个正整数 S ，请问最多可以选出多少个质数，使它们的和小于等于 S 。

【输入格式】

一行一个正整数 S 。

【输出格式】

将这些质数从小往大输出，随后输出质数个数。所有数单独占一行。

【样例 1 输入】

```
100
```

【样例 1 输出】

```
2
3
5
7
11
13
17
19
23
9
```

【样例 2 输入】

```
5
```

【样例 2 输出】

```
2
3
2
```

【样例 3 输入】

```
11
```

【样例 3 输出】

```
2
3
5
3
```

【子任务】

数据范围：$1 \leq S \leq 10^5$。

数位枚举（enum）

【题目描述】

数位 DP 是一种常用的计数方法，不过在这道题中，你并不需要用到这个算法。

给出两个整数 n 和 x ，统计所有 1 到 n 的正整数中 x 出现了多少次。例如在 1 到 14 之间的正整数，5 出现了 1 次，4 出现了 2 次，1 在 $1, 10, 12, 13, 14$ 中各出现 1 次，在 11 中出现 2 次，共计 7 次。

【输入格式】

一行两个整数 n, x，用空格隔开。

【输出格式】

一行一个整数，表示 x 出现的次数。

【样例 1 输入】

```
14 1
```

【样例 1 输出】

```
7
```

【子任务】

数据范围：$1 \leq n \leq 10^6$，$0 \leq x \leq 9$。

阅读论文（read）

【题目描述】

小明十分热爱学习数学，为了方便他查找相关数学定理，他有一个 M 页的笔记本。小明十分勤俭节约，他只会用铅笔在笔记本上书写，如果笔记本写完了，就把最早记录的那一页用橡皮擦擦干净。

小明现在在阅读一篇数学论文，在证明中依次提到了 N 个定理，同一定理可能被多次提及。由于小明的记性不太好，每次遇到一个定理，小明都会在自己的笔记本里面查询这个定理，假如没有这个定理，就翻看教材，然后用 1 页将这个定理记下。假如笔记本已经满了，那么他会将当前的笔记本中最先记录的那一页定理擦除，再记下这个定理。

小明现在想要知道，在阅读整篇论文的过程中，他需要翻阅多少次教材？起初，小明的笔记本是空的。

【输入格式】

共 2 行。每行中两个数之间用一个空格隔开。

第一行为两个正整数 M, N，代表笔记本页数和论文提及定理的次数。

第二行为 N 个非负整数，按照论文的顺序，每个数（大小不超过 1000）代表一个数学定理。论文中提及的两个定理是同一个定理，当且仅当它们对应的非负整数相同。

【输出格式】

一个整数，为小明需要翻教材的次数。

【样例 1 输入】

```
3 7
1 2 1 5 4 4 1
```

【样例 1 输出】

```
5
```

【样例 1 解释】

整个阅读论文的过程如下：每行表示遇到了一个定理后的笔记本内容：

1. **1**：查找定理 1 并写进笔记本。

2. **1 2**：查找定理 2 并写进笔记本。
3. **1 2**：在笔记本中找到定理 1。
4. **1 2 5**：查找定理 5 并写进笔记本。
5. **2 5 4**：查找定理 4 并写进笔记本替代定理 1。
6. **2 5 4**：在笔记本中找到定理 4。
7. **5 4 1**：查找定理 1 并调入笔记本替代定理 2。

【子任务】

对于 10% 的数据有 $M = 1$，$N \leq 5$；
对于 100% 的数据有 $1 \leq M \leq 100$，$1 \leq N \leq 1000$。

【提示】

请注意表示数学定理的数可能为 0，你需要采取有效的手段，分辨数组中“尚未存储有效元素的位置”和“已经存储了 0 的位置”。

在线购物（shopping）

【题目描述】

某在线购物平台上有许多商品，每个商品都有一个商品特征码，这个特征码是一个正整数。每个消费者手中有一个需求码，也是一个正整数。如果一个商品的特征码恰好以消费者的需求码结尾，那么这个商品就是消费者所需要的。小 Z 刚刚成为该购物平台的后台管理员，他知道所有商品的特征码，他请你帮他写一个程序，对于每一位消费者，求出他所需要的商品中特征码最小的那个特征码，如果没有他需要的商品，请输出 -1。

【输入格式】

第一行，包含两个正整数 n, q，以一个空格分开，分别代表该在线购物平台上商品的数量和消费者的数量。

接下来的 n 行，每行包含一个正整数，代表在线购物平台上某商品的商品特征码。

接下来的 q 行，每行包含两个正整数，以一个空格分开，第一个正整数代表在线购物平台上消费者的需求码的长度，第二个正整数代表消费者的需求码。

【输出格式】

共 q 行，每行包含一个整数，如果存在第 i 个消费者所需要的书，则在第 i 行输出第 i 个消费者所需要的商品特征码最小的那个商品的商品特征码，否则输出 -1。

【样例 1 输入】

```
5 5
2123
1123
23
24
24
2 23
3 123
3 124
2 12
2 12
```

【样例 1 输出】

```
23
1123
-1
-1
-1
```

【子任务】

对于 20% 的数据，$1\ n\ 2$。
另有 20% 的数据，$q = 1$。
另有 20% 的数据，所有消费者的需求码的长度均为 1。
另有 20% 的数据，所有的商品特征码按从小到大的顺序给出。
对于 100% 的数据，$1\ n\ 1000, 1\ q\ 1000$，所有的商品特征码和需求码均不超过 10^7。

lhm 玩 01（lhma）

【题目描述】

lhm 喜欢 01，于是他写了一个由 $N \times N$ 的 0 和 1 的矩阵。

但 lhm 感觉这个矩阵太大了，于是他要压缩！他要把矩阵压缩成一串数字，其中第一个数字是 N，剩下的数字为：从矩阵的第一行第一个符号开始计算，按书写顺序从左到右，由上至下。第一个数表示连续有几个 0，第二个数表示接下来连续有几个 1，第三个数再接下来连续有几个 0，第四个数接着连续几个 1，以此类推……

例如: 以下矩阵：

```
0001000
0001000
0001111
0001000
0001000
0001000
1111111
```

对应的压缩数字序列是：7 3 1 6 1 6 4 3 1 6 1 6 1 3 7（第一个数是 N , 其余各位表示交替表示 0 和 1 的个数，压缩数字序列保证 $N \times N =$ 交替的各位数之和）

【输入格式】

数据输入一行，由空格隔开的若干个整数，表示压缩数字序列。

其中，压缩数字序列的第一个数字就是 N，表示这个矩阵应当是 $N \times N$ 的大小。

接下来的若干个数字，含义如题目描述所述。

【输出格式】

输出一个 $N \times N$ 的 01 矩阵（点阵符号之间不留空格）。

【样例 1 输入】

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

【样例 1 输出】

```
0001000
0001000
```

```
0001111
0001000
0001000
0001000
1111111
```

【子任务】

数据保证，$3 \leq N \leq 200$。

bngg 与 hmgg 的决斗（fight）

【题目描述】

bngg 和 hmgg 在战斗。

两人一共开辟了 n 个战场，战场排列成一条线段，依次编号为 $1 \sim n$，相邻编号的战场之间相隔 1 厘米，即所有战场为长度为 $n-1$ 厘米的线段。i 号战场现在有 c_i 名士兵在战斗。

bngg 的大本营在 1 号战场左侧，hmgg 的大本营在 n 号战场右侧，他们的势力范围以 m 号战场分界，m 号战场左侧 bngg 占优，m 号战场右侧 hmgg 占优，而 m 号战场是两人正在争夺的战场，没有人占优。

一个战场的士气为该战场的士兵数 $\times$ 该战场到 m 号战场的距离，一方的士气为自己占优战场士气之和。

bngg 奇袭了 hmgg，将 s_1 名士兵派往了 p_1 号战场。如果双方士气差距太大，会导致 hmgg 或者 bngg 红温。hmgg 还有 s_2 名士兵可供调动，请你确定一个战场 p_2，使得 hmgg 将全部 s_2 名士兵派遣往战场 p_2，两人士气差距尽可能小。

【输入格式】

输入共三行。

输入的第一行为一个正整数 n。

接下来一行包含 n 个正整数，第 i 个为 c_i。

接下来一行包含四个正整数，分别代表 m, p_1, s_1, s_2。

【输出格式】

输出一行一个整数 p_2，若存在多个答案，输出编号最小的那个。

【样例 1 输入】

```
6
2 3 2 3 2 3
4 6 5 2
```

【样例 1 输出】

```
2
```

【样例 2 输入】

```
6
1 1 1 1 1 16
5 4 1 1
```

【样例 2 输出】

```
1
```

【子任务】

数据保证：$1 \le n \le 10^5$，$1 \le s_1, s_2 \le 10^9$，$1 < m < n$，$1 \le p_1 \le n$。

【hints】

请考虑不同类型可以表示的数据范围，并选择合适的那个。

abs 函数是求绝对值函数。请注意，C 语言中 abs 函数只能接受 int 类型参数，若你传入 long long 类型参数，将发生隐式类型转换，存在溢出的可能性。

你可以选择使用 llabs 函数，这个函数将接受 long long 类型的参数。你也可以直接选用 C++ 语言提交，C++ 完全向下兼容 C 语言。C++ 的 abs 函数经过函数重载，可以接受多种数据类型。

lhm 玩数字（lhmb）

【题目描述】

lhm 有 n 个正整数，他很喜欢它们。lhm 突然想知道这些正整数中第 k 个最小整数（大小一样的整数只算一次），你能帮帮他吗？

【输入格式】

第一行为 n 和 k。
第二行开始为 n 个正整数的值，整数间用空格隔开。

【输出格式】

第 k 个最小整数的值；若无解，则输出 `NO RESULT`。

【样例 1 输入】

```
10 3
1 3 3 7 2 5 1 2 4 6
```

【样例 1 输出】

```
3
```

【子任务】

数据保证：$n \leq 10000$，$k \leq 1000$，正整数均小于 30000。

小 S 与 NLP（nlp）

【题目描述】

小 S 是一位世界记录员，她的职责是将来自不同异世界的文字材料整理归档。

虽然异世界的文明不知道为何都是用日语交流，但是他们的文字各异。所以处理这些材料并非易事。

幸好小 S 是一位 NLP（自然语言处理）高手，通过机器学习，她的模型帮她轻松地把所有文本翻译成了英文。

但是小 S 发现，她的语言模型不能很好的处理不同进制数值的转换，所以她只好进行人脑 NLP，手动完成异世界数字文本的替换工作。

幸好，小 S 有一份巨大的字典，其中包含 n 个异世界数字代表的数值。她只需要单纯的依此对翻译后的英文文本进行简单替换即可。

不过由于小 S 正在用电脑玩肉鸽游戏 tarjan lusa，无法使用，所以这项任务就被丢给了你。

为了方便本世界的计算机处理，需要进行替换的异世界数字都被替换成了小写英文字符串，并处于 {} 之间。简单的说，你需要把给出的英文文本里每一句话中全部的 {异世界数字} 替换为变量的值并输出。同时小 S 保证，给你的每句话都满足仅由大小写英文字母、空格、半角逗号、半角句号和 {、} 组成。同时每句话由在 {} 之间的，必然为字典中 N 个异世界数字中的一个。

例如，有 $a = 20, b = 10$，对于句子 `Kitsuki achieved a {b} game winning streak on advanced difficulty of {a}.`，替换后将得到 `Kitsuki achieved a 10 game winning streak on advanced difficulty of 20.`。

【输入格式】

输入共 $n + m + 1$ 行。

输入的第一行为两个整数 n, m。

接下来 n 行，每行一个小写英文字符串、一个整数，分别代表异世界数字和其代表的数值。

接下来 m 行，每行一个需要进行替换的句子。

【输出格式】

输出 m 行，每行一个标注好的句子。

【样例 1 输入】

```
4 2
shinki 1
a 3
tarjanlusa 4
d 5
We have {a} apples.
We {d}onot have pencils.
```

【样例 1 输出】

```
We have 3 apples.
We 5onot have pencils.
```

【子任务】

对于 100% 的测试数据，$1 \leq n \leq 5000$，$1 \leq m \leq 20$。保证 {} 成对合法出现，需要替换的句子长度不超过 5×10^4。异世界数字为长度不超过 20 的英文小写字母串，且对应的数值在 int 范围内。每句话由大小写英文字母、空格、半角逗号、半角句号和 {、} 组成。

【提示】

string 类型是 C++ 提供的有力类型，它封装了很多常用的字符串成员函数。

cin/cout 是 C++ 风格的输入方式，它在读取字符串时尤其好用。

本题使用字符数组同样方便，但在后面的题目中，你可以尝试学习并使用上面提示的内容，并选择自己喜欢的那个。

我们计划为你提供 string 类型的学习材料，有关内容正在编写中。

小 S 与 MMORPG（mmorpg）

【题目描述】

小 S 是一名 MMORPG 玩家，她经常玩一款名为“狒狒食柿”的游戏。

（如果你不知道什么是 MMORPG，你可以将其简单理解为 ***Many Men Online Role Playing as Girls*** 的缩写。）

在狒狒食柿中，有一个叫做潜水艇的每日玩法，简单来说，你有一个海图，由大量互相连接的节点组成，每个节点用一个字符串表示名字，一开始你只发现了一个节点，每次派遣潜艇探索完一个节点以后，与之相连的节点有可能会被发现，从而允许你探索更深的节点。该玩法的目的是探索尽可能多的节点，因此我们每天会且仅会向所有未探索过的节点，或是存在相连节点未被发现的节点派遣潜艇。

潜水艇是一个每天都要处理的日常玩法，但是遗憾的是，小 S 打了一晚上高难度副本 P10S 变成脑瘫了。所以她只能把这个任务交给你来完成。

不过好心的小 S 已经帮你处理好了一部分任务。具体而言，她会告诉你三组节点信息：

- 昨天派遣潜艇的节点（即所有可能发现新节点的节点）
- 今天收回潜艇后，哪些节点的所有相连节点已被发现。
- 新发现了哪些节点

最后，你要给出所有今天应该探索的节点的列表，按字典序顺序输出。

【输入格式】

第一行有三个整数，依次表示昨天派遣潜艇的节点数量 n，所有相连节点已被发现的节点数量 m，和新发现的节点数量 k。

接下来 n 行，每行一个字符串，表示一个昨天派遣潜艇的节点名字。

接下来 m 行，每行一个字符串，表示一个所有相连节点已被发现的节点名字。

接下来 k 行，每行一个字符串，表示一个新发现的节点名字。

【输出格式】

输出若干行，每行一个字符串。

按字典序从小到大的顺序输出今天应该探索的节点的名字。

【样例 1 输入】

```
5 4 1
PureWhiteShallowBeach
DrowningSea1
```

```
DrowningSea2
MatteBasin
DrowningSea3
PureWhiteShallowBeach
DrowningSea1
DrowningSea2
MatteBasin
LimilalaTrench
```

【样例 1 输出】

```
DrowningSea3
LimilalaTrench
```

【子任务】

对全部的测试点，保证：- $1 \leq n \leq 10^5$ - $0 \leq m \leq n$ - $0 \leq k \leq 10^5$ - 除样例外，输入字符串的长度不超过 10。- 输入字符串仅含有大小写字母和数字。- 昨天派遣潜艇的节点名字。- 所有相连节点已被发现的节点名字互不相同，且均是昨天派遣潜艇的节点名字。- 新发现的节点名字互不相同，且均不是昨天派遣潜艇的节点名字。

小 S 与时间逆流（time）

【题目描述】

小 S 是一名时空旅行者，她喜欢在不同的异世界见证文明的发展史。

一个持续时间为 $|S|$ 的文明的发展史可以被记录为一个长度为 $|S|$ 的 01 串，0 代表这个文明处于和平时期，1 代表处于战争时期。

但是旅行结束后，小 S 发现这个世界发生了正好一次可怕的时间逆流！这打乱小 S 已经做好的文明记录！但是由于在异世界中感受不到时间逆流，小 S 并不知道它发生在了哪一段时间中。

简单来说，一次时间逆流对文明记录的影响是，将这段时间内的记录翻转。

形式化的，一次作用于时间区间 $[l,r]$（$1 \le l \le r \le |S|$ ）的时间逆流会将原记录 S 变成 T，同时 T 满足：

$$T_i = \begin{cases} S_i, & i < l \text{ 或} i > r \\ S_{r-(i-l)}, & l \le i \le r \end{cases}$$

这里 01 串的下标从 1 开始。

现在小 S 需要修复这个记录，虽然他不知道具体哪一段时间区间受到了翻转。但是她印象中这个文明的战争更多的发生在晚期。也就是说，字典序最小的修复后的文明记录 T 应该就是正确的。

不过小 S 现在在被她的妹妹拉去玩游戏，所以请你帮帮她找到这个正确的记录。

【输入格式】

输入只有一行一个字符串，表示被打乱的文明记录 S。

【输出格式】

输出一行一个字符串，表示得到的正确文明记录 T ，其字典序应该是所有可能的文明记录中最小的。

【样例 1 输入】

```
101
```

【样例 1 输出】

```
011
```

【样例 2 输入】

```
0010100
```

【样例 2 输出】

```
0000101
```

【子任务】

对 100% 的数据，$1 \leq |S| \leq 100$。S 只含字符 $0, 1$。其中 $|S|$ 表示输入字符串的长度。

小 S 与历史长河（history）

【题目描述】

小 S 是历史长河。

一条历史长河是对一个文明从发展到衰亡的记录，若有好事者对这个文明感兴趣，便可以从历史长河中一探究竟。

历史长河是一段仅包含大小写字母的字符串，记录了每一个时间点这个文明发生的重要事件。字典序越小的字母代表的事件越重要。

现在，有两条不同的历史长河 S 和 T，作为好事者的你正打算对比研究它们。在研究过程中，你遇到了 Q 个问题，在每个问题中，你想要比较两个文明中特定时段的事件重要程度。

形式化的，每次给出 l_s, r_s 和 l_t, r_t，你需要判断 $S[l_s, r_s]$ 和 $T[l_t, r_t]$ 谁的字典序更小。

其中，$S[l, r]$ 表示从字符串 S 的第 l 个字符到第 r 个字符连起来构成的字符串。例如，若 S 为 `kitsuki`，则 $s[3, 5]$ 为 `tsu`。

【输入格式】

第一行是一个字符串 S。

第二行是一个字符串 T。

第三行是一个整数，表示询问次数 Q。

接下来 Q 行，每行四个整数 l_s, r_s, l_t, r_t，表示一次询问。

【输出格式】

对每次询问，输出一行一个字符串：

- 如果 $S[l_s, r_s]$ 的字典序更小，请输出 `yifusuyi`。
- 如果 $T[l_t, r_t]$ 的字典序更小，请输出 `erfusuer`。
- 如果两者的字典序一样大，请输出 `ovo`。

【样例 1 输入】

```
Yifusuyi
yifusuYi
3
1 2 7 8
1 2 1 2
7 8 7 8
```

【样例 1 输出】

```
ovo
yifusuyi
erfusuer
```

【子任务】

对 100% 的数据，$1 \leq |S|, |T|, Q \leq 10^3$，$1 \leq l_s \leq r_s \leq |S|$，$1 \leq l_t \leq r_t \leq |T|$。输入字符串仅含大小写英文字母。其中 $|S|$ 表示历史长河 S 的长度，$|T|$ 表示历史长河 T 的长度。

任务管理（task）

【题目描述】

你现在有 N 个任务，依次编号为 $1 \sim N$。部分任务有一些前置任务。现在，由于你编号为 1 的任务已经接近截止时间，十分紧急。你想要知道，如果要完成 1 号任务，至少一共要完成多少任务？

【输入格式】

输入共 $N+1$ 行。

输入的第一行为一个正整数 N。

接下来 N 行，第 i 行描述了编号为 i 的任务的前置任务：

- 第 i 行共有 C_i+1 个数。
- 第一个数为 C_i。
- 接下来 C_i 个数，描述了该任务的前置任务的编号。

【输出格式】

输出一行一个整数，表示至少需要完成的任务数量。

【样例 1 输入】

```
5
1 2
1 3
1 4
0
0
```

【样例 1 输出】

```
4
```

【子任务】

对于 100% 的测试数据，$1 \leq N \leq 5000$，$0 \leq C_i < N$，保证任务之间不会构成循环依赖。

直接输出（output）

【题目描述】

任何正整数都可以以 2 的幂次方形式表示。例如，137 可以写成 $2^7+2^3+2^0$。
同时，我们约定使用括号表示次方，即 a^b 表示为 $a(b)$。
因此，137 可以表示为 $2(7)+2(3)+2(0)$。
进一步拆解：
7 可以表示为 2^2+2+2^0（其中 2^1 用 2 表示），而 3 可以表示为 $2+2^0$。
因此，最终可以将 137 表示为 $2(2(2)+2+2(0))+2(2+2(0))+2(0)$。
另一个例子是 $1315=2^{10}+2^8+2^5+2+1$。
因此，1315 最终可以表示为 $2(2(2+2(0))+2)+2(2(2+2(0)))+2(2(2)+2(0))+2+2(0)$。

【输入格式】

一行一个正整数 n。

【输出格式】

符合约定的 n 的 $0,2$ 表示（在表示中不能有空格）。

【样例 1 输入】

```
1315
```

【样例 1 输出】

```
2(2(2+2(0))+2)+2(2(2+2(0)))+2(2(2)+2(0))+2+2(0)
```

【子任务】

对于 100% 的数据，$1\le n\le 2\times 10^4$。

走（walk）

【题目描述】

在一个 $m \times m$ 的棋盘上，每个方格可能是红色、黄色或无色。任务是从最左上角移动到最右下角。在任何时刻，所站的位置必须是有颜色的，只能向上、下、左、右四个方向移动。当从一个格子移动到另一个格子时，如果两个格子的颜色相同，则无需支付金币；如果颜色不同，则需要支付 1 个金币。

此外，可以花费 2 个金币使用魔法，将下一个无色格子短暂变为指定颜色。魔法不能连续使用，且持续时间有限。一旦使用了魔法，当走到这个暂时有颜色的格子上时不能再次使用魔法；只有在离开这个位置，走到原本有颜色的格子上时才能再次使用魔法。当离开位置时（通过施展魔法使其变为有颜色的格子），该格子恢复为无色。

现在的问题是，从最左上角移动到最右下角，要求达成的最少金币花费是多少？

【输入格式】

第一行包含两个正整数 $ m, n$，以一个空格分开，分别代表棋盘的大小，棋盘上有颜色的格子的数量。

接下来的 $ n $ 行，每行三个正整数 $ x, y, c$，分别表示坐标为 (x, y) 的格子有颜色 $ c$。

其中 $ c=1$ 代表黄色，$ c=0$ 代表红色。棋盘左上角的坐标为 $(1, 1)$，右下角的坐标为 (m, m)。

棋盘上其余的格子都是无色。保证棋盘的左上角一定是有颜色的。

【输出格式】

一个整数，表示花费的金币的最小值，如果无法到达，输出 -1。

【样例 1 输入】

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

【样例 1 输出】

```
8
```

【样例 2 输入】

```
5 5
1 1 0
1 2 0
2 2 1
3 3 1
5 5 0
```

【样例 2 输出】

```
-1
```

【子任务】

对于 30% 的数据，$1 \leq m \leq 5$，$1 \leq n \leq 10$。

对于 60% 的数据，$1 \leq m \leq 20$，$1 \leq n \leq 200$。

对于 100% 的数据，$1 \leq m \leq 100$，$1 \leq n \leq 1000$。

【提示】

题目中说，只能朝四个方向走。可否对使用金币变化颜色的规则做一些拓展，让主角可以往十二个方向走，同时去部分关于颜色的规则呢？

方向太多，若逐个回溯是否代码太长？请查找资料并学习搜索中的**增量数组**。

选择（choose）

【题目描述】

给你 n 个数和一个整数 k。

问有你有多少种方法从 n 个数里面选 k 个数使得和为质数？

【输入格式】

第一行两个空格隔开的整数 n, k。

第二行 n 个整数，分别为 $x_1, x_2, \cdots, x_n$。

【输出格式】

输出一个整数，表示种类数。

【样例 1 输入】

```
4 3
3 7 12 19
```

【样例 1 输出】

```
1
```

【子任务】

数据保证：$1 \leq n \leq 20$，$k < n$，$1 \leq x_i \leq 5 \times 10^6$。

大物要挂了（nnzdqzrc）

【题目描述】

期末到了，盖亚开始搞学习了。于是在“你能做的，岂止如此”的恐吓下，盖亚开始刷习题集，每科都有一个习题集，一共有四科，分别有 s_1, s_2, s_3, s_4 道题目。虽然盖亚已经完成了超进化，但是完成每道题目都需要一些时间，这些时间并不相等，分别用 A_i, B_i, C_i, D_i 表示。

盖亚有一个能力，他可以拉雷伊过来做题，于是他可以同时计算 2 道不同的题目，但是仅限于同一科。因此，盖亚必须一科一科的复习。

因为大二下要到了，因此盖亚非常急，希望在不挂科的情况下尽快把事情做完，所以他希望知道能够完成复习的最短时间。

【输入格式】

本题包含 5 行数据：第 1 行，为四个正整数 s_1, s_2, s_3, s_4。

第 2 行，为 $A_1, A_2, \ldots, A_{s_1}$ 共 s_1 个数，表示第一科习题集每道题目所消耗的时间。

第 3 行，为 $B_1, B_2, \ldots, B_{s_2}$ 共 s_2 个数。

第 4 行，为 $C_1, C_2, \ldots, C_{s_3}$ 共 s_3 个数。

第 5 行，为 $D_1, D_2, \ldots, D_{s_4}$ 共 s_4 个数，意思均同上。

【输出格式】

输出一行，为盖亚复习完毕最短时间。

【样例 1 输入】

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

【样例 1 输出】

```
20
```

【子任务】

数据保证：$1 \le s_1, s_2, s_3, s_4 \le 20$，$1 \le A_i, B_i, C_i, D_i \le 60$。

世界是一个巨大的二分（binary）

【题目描述】

小 F 会给出一个长度为 n 的数列 a，并进行以下五种询问：

- `1 x`：查询数列中有多少个数**刚好等于** x 。
- `2 x y`：查询数列中有多少个数**大于等于** x 并且**小于等于** y 。
- `3 x y`：查询数列中有多少个数**大于等于** x 并且**小于** y 。
- `4 x y`：查询数列中有多少个数**大于** x 并且**小于等于** y 。
- `5 x y`：查询数列中有多少个数**大于** x 并且**小于** y 。

【输入格式】

第一行读入两个整数 n, m，表示数列的长度为 n，一共有 m 次询问。

第二行读入 n 个整数表示数列 a。

接下来 m 行，每行输入 `p x` 或 `p x y`，表示一个询问，其中 p 表示询问类型。

【输出格式】

对于每次询问输出一行一个整数表示询问的答案。

【样例 1 输入】

```
6 6
3 -7 3 6 -2 3
1 3
1 999
2 -2 3
3 -2 3
4 -2 3
5 -2 3
```

【样例 1 输出】

```
3
0
4
1
3
```

```
0
```

【样例 2 输入】

```
5 1
0 0 0 50 105
5 100 1
```

【样例 2 输出】

```
0
```

【样例 3 输入】

```
5 1
0 50 50 50 105
5 100 1
```

【样例 3 输出】

```
0
```

【子任务】

对于 100% 的数据：$1 \leqslant n, m \leqslant 10^5$, $-10^9 \leqslant a_i, x, y \leqslant 10^9$。

【hints】

如果你直接通过了本题，请尝试使用同一个自定义函数完成全部操作。

STL 为我们提供了二分查找的强大利器 lower_bound 函数与 upper_bound 函数，请你自行查阅资料，学习前面两个函数的用法，并尝试使用这两个函数通过本题。

方程求解（answer）

【题目描述】

小 A 有 n 个关于 x 的方程，第 i 个方程形如 $a_i x_i + b_i = c_i$。方程的解 x 均为正整数，例如下面几个方程都是符合要求的方程：

```
2x+4=10
-3x+13=10
4x-8=16
```

其中，第一组方程的解为 $x_1 = 3$，第二组方程的解为 $x_2 = 1$，第三组方程的解为 $x_3 = 6$。

小 A 想要知道，给定 L, R，在 $L \le x \le R$ 的范围内，有多少个正整数 x 满足 x 是其中至少一个方程的解。为了防止你欺骗他，他会询问你 Q 次。

【输入格式】

第一行输入两个正整数 n, Q，分别表示小 A 有的方程数，以及小 A 想要向你询问的次数。

第二行开始，往下 n 行，每行一个字符串，描述一个方程。

第 $(n+2)$ 行开始，往下 Q 行，每行两个正整数 L, R，表示一次询问，即给定 L, R，询问在 $L \le x \le R$ 的范围内，有多少个正整数 x 满足 x 是其中至少一个方程的解。

【输出格式】

对于每次询问，输出一行一个整数，表示有多少个在 $L \le x \le R$ 的范围内的正整数 x，满足 x 是其中至少一个方程的解。

【样例 1 输入】

```
3 4
2x+4=10
-3x+13=10
4x-8=16
1 6
1 8
3 6
4 5
```

【样例 1 输出】

```
3
3
2
0
```

【样例 1 解释】

对于第一组样例，即为题目中的举例。三组方程的解分别为 $x_1 = 3, x_2 = 1, x_3 = 6$。则：

- 对于 $1 \le x \le 6$ 的范围，有 3 个 x 的取值（$x = 1, 3, 6$）是其中至少一个方程的解；
- 对于 $1 \le x \le 8$ 的范围，同上所述；
- 对于 $3 \le x \le 6$ 的范围，有 2 个 x 的取值（$x = 3, 6$）是其中至少一个方程的解；
- 对于 $4 \le x \le 5$ 的范围，不存在一个 x 是其中至少一个方程的解；
- 因此分别输出 $3, 3, 2, 0$。

【样例 2 输入】

```
5 3
5x-2=13
8x+5=45
4x-12=8
-2x+10=4
3x-7=2
1 3
1 5
3 5
```

【样例 2 输出】

```
1
2
2
```

【样例 2 解释】

对于第二组样例，五组方程的解分别为 $x_1 = 3, x_2 = 5, x_3 = 5, x_4 = 3, x_5 = 3$。则：

- 对于 $1 \le x \le 3$ 的范围，只有 $x = 3$ 满足是其中至少一个方程的解；
- 对于 $1 \le x \le 5$ 的范围，有 2 个 x 的取值（$x = 3, 5$）是其中至少一个方程的解；
- 对于 $3 \le x \le 5$ 的范围，有 2 个 x 的取值（$x = 3, 5$）是其中至少一个方程的解；
- 因此分别输出 $1, 2, 2$。

【子任务】

数据保证，$1 \le n, Q \le 2 \times 10^5$，方程中 a_i, b_i, c_i 满足 $1 \le |a_i|, |b_i|, |c_i| \le 10^9$，每一组方程的解 x_i 必定为正整数。询问时的 L, R 满足 $1 \le L \le R \le 2 \times 10^9$。

本题输入数据较大，请注意代码输入输出的运行效率。

【hints】

将方程作为字符串处理，是否有一些繁琐？

可否通过 getchar 函数，自行实现一个输入函数，忽略其他字符，只读入数字？

上面的函数被称为快速读入函数，可查阅资料学习。

IT 中国课程报告（report）

【题目描述】

你的桌子被沁苑鼠鼠啃坏了！但是现在是 1 月 17 日 23:00，距离 IT 中国报告截止还有 1 小时，你必须找一块平整的地方来放置电脑撰写报告。

你决定用你的书摞起书堆，直到书堆不低于你的身高。你一共有 N 本书，第 i 本书的厚度为 H_i，你的身高为 B。

显而易见，书的数目越多，书堆越不稳定。请问你至少使用多少本书，可以达到目标。

【输入格式】

输入共两行。

输入的第一行为两个正整数 N, B。

输入的第二行为 N 个正整数，第 i 个代表 H_i。

【输出格式】

输出一行一个整数，表示最少使用的书的本数。

【样例 1 输入】

```
6 40
6
18
11
13
19
11
```

【样例 1 输出】

```
3
```

【子任务】

对于 100% 的测试数据，$1 \le N \le 20000$，$1 \le H_i \le 10000$，$1 \le B \le \sum H_i \le 2 \times 10^9$。

自助售货机（vem）

【题目描述】

你运营了一个自助售货机，售货机共有 n 层，每层 m 个货槽，每个货槽中放置一种商品。

顾客购买商品后，对应货槽吐出商品，商品经过自由落体摔落到自助售货机的底部。如果商品摆放在从下向上数第 i 层的货槽，商品的摔落距离即为 i。如果摔落距离过高，则可能损坏商品。

现在小 F 需要安排 $k(k \leq n \cdot m)$ 种商品在自助售货机上的摆放位置。第 i 种商品的摔落距离不能超过 H_i。请你判断是否存在这样的一种摆放方式。

【输入格式】

输入共两行。

输入的第一行为三个正整数 n, m, k。

输入的第二行为 k 个正整数，第 i 个代表 H_i。

【输出格式】

输出一行一个字符串，表示是否存在合法的摆放方式：

- 若存在，输出 `Yes`。
- 若不存在，输出 `No`。

【样例 1 输入】

```
5 5 5
20 20 20 20 20
```

【样例 1 输出】

```
Yes
```

【样例 2 输入】

```
5 5 6
1 1 1 1 1 1
```

【样例 2 输出】

```
No
```

【子任务】

本题使用 Subtask，仅有通过该 Subtask 中全部的测试点方可获得该 Subtask 的分数。

对于 100% 的测试数据，$1 \le n, m \le 5000$，$1 \le k \le \min\{n \cdot m, 10^5\}$，$1 \le H_i \le 10000$。

- Subtask 1(7 pts)：$H_i = 1$；
- Subtask 2(8 pts)：$n = 1$；
- Subtask 3(15 pts)：$m = 1$；
- Subtask 4(15 pts)：$k \le 5000$；
- Subtask 5(15 pts)：H_i 单调不降给出。
- Subtask 6(40 pts)：无特殊限制。

hints

冒泡排序的时间复杂度为 $O(n^2)$，一般来说，在时间限制为 1s 的情况下，只能通过 $n \le 5000$ 的数据。

在本题中，你可能需要使用时间复杂度为 $O(n \log n)$ 的排序算法，你可以尝试搜索 sort 函数相关的资料，这是 C++ `algorithm` 头文件中提供的一个函数。

矩阵加速（matrix）

【题目描述】

假设有数列 f，$f(i)$ 为数列的第 i 项，$f(i) = a \cdot f(i-1) + b \cdot f(i-2) + c(i > 2)$。
求 $f(n)$，由于答案可能很大，只需要输出答案对 998244353 取模的结果。

【输入格式】

输入共三行。
输入的第一行为一个正整数 n。
输入的第二行为两个正整数，依次表示 $f(1), f(2)$。
输入的第三行为三个非负整数 a, b, c。

【输出格式】

输出一行一个整数，表示答案。

【样例 1 输入】

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

【样例 1 输出】

```
16
```

【子任务】

对于 100% 的测试数据，$1 \le n \le 10^{18}$，$1 \le f(1), f(2) < 998244353$，$0 \le a, b, c < 998244343$。

【hints】

计算 a^b，存在 $O(\log b)$ 的算法，请自行查找资料学习**快速幂**算法。
假设 A 是 n 阶方阵，A^k 亦可使用快速幂计算，只需要将快速幂中的普通乘法全部换为矩阵乘法。

有递推公式，$f(i) = a \cdot f(i-1) + b \cdot f(i-2) + c$，我们可以用矩阵来表示这个递推式。

$$\begin{bmatrix} f(i) & f(i-1) & c \end{bmatrix}^T = A \times \begin{bmatrix} f(i-1) & f(i-2) & c \end{bmatrix}^T$$

容易发现，A 一定是一个 3×3 的方阵，那么核心就是得到 A，请自行进行推导。

求出 A 后，容易得到，

$$\begin{bmatrix} f(n) & f(n-1) & c \end{bmatrix}^T = A^{n-2} \times \begin{bmatrix} f(2) & f(1) & c \end{bmatrix}^T$$

即可在 $O(\Sigma^2 \log n)$ 的时间复杂度求得答案，其中 Σ 代表 A 的阶。

向量（vector）

【题目描述】

现在给你 n 个 m 维向量，你先对每一个向量找一个所有维度都比他大的向量，如果有多个这样的向量，则输出编号最小的。

【输入格式】

输入共 $n+1$ 行。
输入的第一行包含两个正整数 n 和 m，分别表示向量个数和维数。
接下来 n 行依次输入 m 个向量。

【输出格式】

输出共 n 行，每行一个整数，表示找到的向量编号，如果没有，输出 0 。

【样例 1 输入】

```
4 2
0 0
-1 -1
1 2
0 -1
```

【样例 1 输出】

```
3
1
0
3
```

【子任务】

对于 100% 的测试数据，保证 $0 < m \leq 10$，$0 < n \leq 1000$，所有元素的绝对值均为不大于 10^6 的整数。

没有意义的组成部分（meaningless）

【题目描述】

已知每个数都可以通过唯一分解定理分解成若干个质数相乘的形式, 形如 $a = p_1^{a_1} \times p_2^{a_2} \times ... \times p_n^{a_n}$, 现在我们想求出通过分解 a，忽略的指数小于 k 的质数，剩下数的乘积。

【输入格式】

输入共 $q+1$ 行
输入的第一行包含一个正整数 q ，表示询问个数
接下来 q 行每行包含两个整数 a , k 表示一个询问

【输出格式】

输出共 q 行，每行一个整数，表示所求的数。

【样例 1 输入】

```
3
2155895064 3
2 2
10000000000 10
```

【样例 1 输出】

```
2238728
1
10000000000
```

【子任务】

对于 100% 的测试数据，$1 < k, q \leq 10$，$0 < a \leq 10^{10}$。

跳跃（jump）

【题目描述】

现在给你平面直角坐标系上的 m 个点 (x_i, y_i) , 现在让这些坐标做 n 次位移，每次位移由两个参数 (dx_i, dy_i) 表示，将 (x, y) 变化为 $(x + dx_i, y + dy_i)$ ，现在让你求出这 m 个点经过这 n 次位移之后的坐标。

【输入格式】

输入共 $n + m + 1$ 行。

输入的第一行包含空格分隔的两个正整数 n 和 m，分别表示位移和点个数。

接下来 n 行依次输入 n 个操作，其中第 $i(1 \leq i \leq n)$ 行包含空格分隔的两个整数 dx_i、dy_i。

接下来 m 行依次输入 m 个坐标，其中第 $i(1 \leq i \leq m)$ 行包含空格分隔的两个整数 x_i、y_i。

【输出格式】

输出共 m 行，每行两个整数，表示经过位移变化后的坐标。

【样例 1 输入】

```
3 2
10 10
0 0
10 -20
1 -1
0 0
```

【样例 1 输出】

```
21 -11
20 -10
```

【子任务】

对于 100% 的测试数据，$1 \leq n, m \leq 100$，x, y, dx, dy 均为整数且绝对值不超过 10000。

变换（change）

【题目描述】

对于平面直角坐标系上的点 (x, y) , 我们现在有两种坐标变换的方法，方法一是拉伸 k 倍，把坐标变换成 (kx, ky)，方法二是通过旋转 θ，把坐标变换为 $(x\cos\theta - y\sin\theta, x\sin\theta + y\cos\theta)$。其中，$\theta$ 的单位为 rad。

现在有一个包含了 n 个坐标变换的序列 $(t_1, t_2, t_3, ..., t_n)$ ，其中 t_i 表示了一个坐标变换, 然后现在又定义了一个询问，给定 (i, j, x, y) 四元组，询问坐标 (x, y) 经过 $(t_i, t_{i+1}, ..., t_j)$ 坐标变换后的坐标, 共有 m 个这样的询问, 计算出每个询问的结果。

【输入格式】

输入共 $n + m + 1$ 行。

输入的第一行包含空格分隔的两个正整数 n 和 m，分别表示坐标变换和询问个数。

接下来 n 行依次输入 n 个坐标变换，第一个数表示采用的方法 (1 表示拉伸，2 表示旋转)，第二个实数表示 k 或 θ

接下来 m 行依次输入 m 个询问，每行包含空格分隔的四个整数 i, j, x, y。

【输出格式】

输出共 m 行，每行两个实数，表示经过位移变化后的坐标, 需要保证绝对误差不大于 0.1。

【样例 1 输入】

```
10 5
2 0.59
2 4.956
1 0.997
1 1.364
1 1.242
1 0.82
2 2.824
1 0.716
2 0.178
2 4.094
1 6 -953188 -946637
1 9 969538 848081
```

```
4 7 -114758 522223
1 9 -535079 601597
8 8 159430 -511187
```

【样例 1 输出】

```
-1858706.758 -83259.993
-1261428.46 201113.678
-75099.123 -738950.159
-119179.897 -789457.532
114151.88 -366009.892
```

【子任务】

对于 100% 的测试数据，$0 \leq n, m \leq 100000$，x, y 均为整数且绝对值不超过 1000000，单个拉伸操作系数 $0.5 \leq k \leq 2$，任意区间的拉伸系数 k 的乘积满足 $0.001 \leq k \leq 1000$。

【提示】

前缀和、前缀积及其思想是处理区间可加性问题的利器，请自行查阅资料学习。

在本题中，容易发现拉伸和旋转两种操作顺序无关，可对拉伸和旋转分别应用前缀积、前缀和处理。

图案重现（pattern）

【题目描述】

现在给你 n 个 8×8 的字符矩阵, 仅由大小写的 *、k、q、r、b、n、q 组成，你需要判断第 i 个字符矩阵在之前出现过几次。

【输入格式】

输入共 $8 \cdot n + 1$ 行。
输入的第一行包含两个正整数 n ，分别表示有 n 个字符矩阵。
接下来 $8 \cdot n$ 行, 依次表示这 n 个字符矩阵。

【输出格式】

输出共 n 行，表示这个字符矩阵是第几次出现。

【样例 1 输入】

```
8
********
******pk
*****r*p
p*pQ****
********
**b*B*PP
****qP**
**R***K*
********
******pk
*****r*p
p*pQ****
*b******
****B*PP
****qP**
**R***K*
********
******pk
*****r*p
```

```
p*p*****
*b**Q***
****B*PP
****qP**
**R***K*
******k*
******p*
*****r*p
p*p*****
*b**Q***
****B*PP
****qP**
**R***K*
******k*
******p*
*****r*p
p*pQ****
*b******
****B*PP
****qP**
**R***K*
********
******pk
*****r*p
p*pQ****
*b******
****B*PP
****qP**
**R***K*
********
******pk
*****r*p
p*p*****
*b**Q***
****B*PP
****qP**
**R***K*
```

```
********
******pk
******rp
p*p*****
*b**Q***
****B*PP
****qP**
**R***K*
```

【样例 1 输出】

```
1
1
1
1
1
2
2
1
```

【子任务】

对于 100% 的测试数据，$1 \leq n \leq 100$。

【hints】

可用字符串哈希进行判断。

亦可以使用 STL 提供的 map、set、unordered_map 等容器进行去重，请自行查找资料学习。

人工智能（ai）

【题目描述】

Softmax 函数是人工智能重要应用，一些矩阵对其非常重要。

给你 3 个 n 行 d 列的矩阵 Q, K, V，和一个 n 维的向量 W，现在让你求出 $(W \cdot (Q \times K^T)) \times V$，并输出这个计算之后的矩阵。

【输入格式】

输入的第一行包含两个正整数 n, d ，表示矩阵的大小。

接下来一次输入矩阵 Q, K, V，每个矩阵输入 n 行，每行包含 d 个整数。

最后一行输入 n 个整数，表示向量 W。

【输出格式】

输出共 n 行，每行 d 个整数，表示计算的结果。

【样例 1 输入】

```
3 2
1 2
3 4
5 6
10 10
-20 -20
30 30
6 5
4 3
2 1
4 0 -5
```

【样例 1 输出】

```
480 240
0 0
-2200 -1100
```

【子任务】

对于 100% 的测试数据，$0 < n \leq 10000$，$d \leq 20$，输入的元素均为整数，且绝对值均不超过 1000。

大厦建设（constr）

【题目描述】

华中科技大学土地辽阔，现在有 N 块土地已经被用作建设开发，每块土地可以被视作平面直角坐标系下的一块矩形区域，由左下角坐标 (x_1, y_1) 和右上角坐标 (x_2, y_2) 确定。这 N 块土地没有重合，仅在边界上可能重叠。

计算机学院需要土地进行计算机大楼的建设。在学校的大力支持下，计算机学院选定了左下角坐标 $(0, 0)$，右上角坐标 (a, b) 的土地用作大厦建设。学校需要对这块土地内，已经被用于建设开发的土地进行征收，每单位面积的土地征收需要一枚金币。

请问学校至少花费多少枚金币，才能为计算机大楼的建设征收完土地？

【输入格式】

从文件 ***constr.in*** 中读入数据。

输入共 $N + 1$ 行。

输入的第一行为三个整数 N, a, b。

接下来 N 行，每行四个整数 x_1, y_1, x_2, y_2，描述了一块已用作建设开发土地的位置。

【输出格式】

输出到文件 ***constr.out*** 中。

输出一行一个整数，代表学校最少花费的金币数目。

【样例 1 输入】

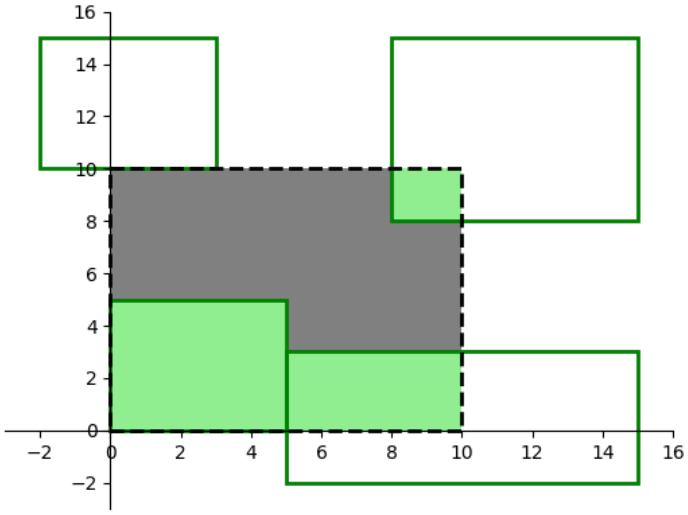
```
4 10 10
0 0 5 5
5 -2 15 3
8 8 15 15
-2 10 3 15
```

【样例 1 输出】

```
44
```

【样例 1 解释】

总花费金币数目为 44。



【样例 2 输入】

```
6 5 5
6 6 7 7
7 7 8 8
11 11 12 12
5 5 6 6
8 8 9 9
9 9 10 10
```

【样例 2 输出】

```
0
```

【子任务】

对于所有测试数据，保证 $1 \leq N \leq 10^5$，$1 \leq a, b \leq 10^7$，$0 \leq |x_i|, |y_i| \leq 10^9$，$x_2 > x_1$，$y_2 > y_1$。

测试点编号	$N \leq$	$a, b \leq$	$\lvert x_i\rvert, \lvert y_i\rvert \leq$	性质 A	性质 B
$1, 2, 3$	1	100	1000	$\times$	$\times$
$4, 5$	15	5000	10000	$\surd$	
$6, 7$				$\times$	$\surd$
$8, 9, 10$					$\times$
$11, 12$	1000			$\surd$	
$13, 14$				$\times$	$\surd$
$15, 16$		10^7	10^9	$\surd$	
$17, 18, 19, 20$				$\times$	$\times$

性质 A：保证 $x_2 = x_1 + 1$，$y_2 = y_1 + 1$。

性质 B：保证 $x_i, y_i > 0$。

技能冷却（skill）

【题目描述】

在一款游戏中，一个英雄有 n 个技能, 冷却时间分别为 t_i，技能升级可以减少冷却时间，现在你有 m 次技能点，使用 c_i 个技能点可以给第 i 个技能减少 1 冷却时间，但是因为使用技能消耗时间，所以技能的冷却时间最少有 k 秒，现在这个英雄想让 $\max(t_1, t_2, ..., t_n)$ 最小，请问最小是多少?

【输入格式】

输入共 $n+1$ 行.

输入的第一行包含三个正整数 n, m, k ，含义见上文

接下来 n 行，每行含有两个正整数 t_i, c_i，表示技能 i 原本的冷却时间和升级所需要的技能点。

【输出格式】

输出一个整数，表示最小化的冷却时间的最大值。

【样例 1 输入】

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

【样例 1 输出】

```
5
```

【样例 2 输入】

```
4 30 2
6 1
5 1
6 2
7 1
```

【样例 2 输出】

```
2
```

【子任务】

对于 100% 的测试数据，$0 < n, t_i, c_i \leq 100000$，$0 < m \leq 10^9$。

信息编码（code）

【题目描述】

小 F 得到了一组由 n 个数组成的信息。第 i 个数 b_i 可能的取值范围为 $0 \sim a_i - 1$。

小 F 发明了一种编码方式，可以用一个数 m 来表示这些信息。记 $c_i = a_1 \cdot a_2 \cdot \ldots \cdot a_i (i \geq 1), c_0 = 1$，则 m 的计算公式如下：

$$m = \sum_{i=1}^{n} c_{i-1} \cdot b_i = c_0 \cdot b_1 + c_1 \cdot b_2 + \cdots + c_{n-1} \cdot b_n$$

现在，给出 m 和数组 a，请你还原出数组 b。

【输入格式】

输入共两行。

输入的第一行为两个整数 n, m。

输入的第二行为 n 个整数，第 i 个表示 a_i。

【输出格式】

输出一行 n 个整数，表示数组 b。

【样例 1 输入】

```
15 32767
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
```

【样例 1 输出】

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

【样例 2 输入】

```
4 0
2 3 2 5
```

【样例 2 输出】

```
0 0 0 0
```

【样例 3 输入】

```
7 23333
3 5 20 10 4 3 10
```

【样例 3 输出】

```
2 2 15 7 3 1 0
```

【子任务】

对于 100% 的测试数据，$1 \leq n \leq 20$，$2 \leq a_i \leq 20$，$\prod a_i \leq 10^9$

【hints】

对于所有的 $j > i$，有 c_j 是 c_i 的倍数。$m \bmod c_i = \sum_{k=1}^{i} c_{k-1} \cdot b_k$。

Emily 家里的饭（meal）

【题目描述】

Emily 想要在 t 时间中做出 n 道菜，其中第 i 道菜耗时 t_i，大部分菜品可以同时制作，但是有的菜品之间存在依赖关系，如果菜品 i 依赖于菜品 j，那么只有在菜品 j 制作完成后才能制作菜品 i，如果 a 时刻开始制作菜品 j，那么菜品 i 只能在第 $a+t_j$ 时刻开始制作，每个菜品至多依赖一个别的菜品，没有依赖的菜品，可以从时刻 1 开始制作。

对于每一个菜品，需要计算：

1. 最早可以制作的时间。
2. 在不耽误 t 时间做完所有菜的前提下，最晚多久开始制作该菜品。

【输入格式】

输入共 3 行。

输入的第一行包含两个正整数 t, n，含义见上文。

输入的第二行包含 n 个正整数，其中第 i 个整数 p_i 表示菜品 i 依赖的菜品，满足 $0 \le p_i < i, p_i = 0$ 表示菜品 i 无依赖。

输入的第三行为 n 个整数，第 i 个为 t_i。

【输出格式】

输出的第一行包含 n 个正整数，表示每个菜品的最早开始时间。

如果可以在 t 时间中做出这 n 道菜，则输出第二行，否则不输出。

输出的第二行包含 n 个正整数，表示每个菜品的最晚开始时间。

【样例 1 输入】

```
10 5
0 0 0 0 0
1 2 3 2 10
```

【样例 1 输出】

```
1 1 1 1 1
10 9 8 9 1
```

【样例 2 输入】

```
10 7
0 1 0 3 2 3 0
2 1 6 3 10 4 3
```

【样例 2 输出】

```
1 3 1 7 4 7 1
```

【样例 3 输入】

```
10 5
0 1 2 3 4
10 10 10 10 10
```

【样例 3 输出】

```
1 11 21 31 41
```

【子任务】

对于 100% 的测试数据，$0 < t \leq 365$，$0 < n \leq 100$。