

大连大学 2020 年 ACM/CCPC 程序设计竞赛校内赛(网赛)

正式赛



主办单位：大连大学创新创业学院

承办单位：大连大学程序设计工作室

2020 年 5 月 4 日

Problem A. Yet Another Find Your Miku

Input file: standard input
 Output file: standard output
 Time limit: 1 second
 Memory limit: 512 megabytes



在过去的 2019 年, Jyo 成功用你的程序输出的咒语解开了魔法冰块, 救出了被困其中的 Miku。但这次 Miku 又被封印在了二向箔中! 不过好在现在已经开发出了曲率引擎技术, 只要知道 Miku 在二向箔中的坐标即可成功解救 Miku。众所周知, Miku 左臂上有两个数字 01, 这也成为了找到她的线索。现在给出整张二向箔, 请你帮助 Jyo 破解 Miku 的坐标, 把 Miku 从二向箔中解救出来!

二向箔是一个 $n \times m (1 \leq n, m \leq 1000)$ 的字符矩阵(仅由字符 `.` 和字符 `*` 组成的矩阵)。每个 `.` 或 `*` 都代表一个像素点, 其中 `*` 代表黑点, `.` 代表白点。这些像素点会组成 `0` 和 `1` 的字符画, 其中满足:

1. 不会出现 `0` 和 `1` 以外的字符画。
2. `0` 和 `1` 都是四连块(一个字符画的某个像素点的上下左右相邻四格, 至少有一格必定存在同属于该字符画的像素点; 一个字符画的某个像素点的上下左右斜对角相邻八格不存在其他字符画的像素点), 且字符画不会相互接触, 也不会相互包含或在边界以残缺不全的方式出现。
3. 字符画的形状一定和下表中的图形拓扑等价(可以随意拉伸但不能拉断, 拉伸的过程中不会使该字符画的一部分接触到另外一部分)。

字符画 0	字符画 1

现在已知字符画 `0` 的数量就是 Miku 所在的横坐标, 字符画 `1` 的数量就是 Miku 所在的纵坐标, 请你输出 Miku 的坐标以帮助 Jyo 用曲率引擎将 Miku 带离二向箔!

Input

输入第一行包含两个正整数 $n, m (1 \leq n, m \leq 1000)$, 分别表示二向箔的行高和列宽。

接下来有一个 $n \times m$ 的二向箔, 二向箔中只包含两种字符: `.` 和 `*`。其中 `.` 表示白点, `*` 表示黑点。

Output

输出为一行，包含两个非负整数 x 和 y ，即 Miku 在二向箔中的坐标，两数中间用空格隔开。

Examples

standard input	standard output
<pre> 10 20***..... ..***.....*.*..... .***.*.....*.*..... .*.*.....***..... .*.*.....*.*..... .***.*.....*.*.....**..... ...**.....*..... ...*.....*..... ...***.....***..... </pre>	<pre> 2 2 </pre>

standard input	standard output
<pre> 10 10*****. ..*****. .**...**. .**...**. .**...**. .**...**. .**...**. ..*****. ..*****. </pre>	<pre> 1 0 </pre>

Explanation

第一个样例，二向箔中包含了 2 个字符画 **0** ,2 个字符画 **1**，所以 Miku 的坐标就是 (2,2)。

Problem B. ISLANDERS

Input file: standard input
 Output file: standard output
 Time limit: 1 second
 Memory limit: 256 megabytes



ISLANDERS 是一款城市建造类游戏，游戏从极简主义出发，通过慢慢的积累，建设美轮美奂的小岛。游戏玩法相对比较直观，从一组建筑物开始，摆放建筑物将获得分数，而分数取决于放置建筑物周围的环境。例如如果已有了市中心，再放置田地的时候会扣分，而放置豪宅则会加分。现在已知有三种建筑物以及他们互相之间的分数影响，请你计算出将他们全部摆放后的最大得分。

在游戏中，你有一座小岛，你需要将你所拥有的**所有建筑物**放在这座小岛上，注意，你即将放下的建筑的得分取决于**该区块已有的建筑物**。

得分规则是这样的：我们分别将三种建筑物依次记为第 1、2、3 种建筑物，对应的数量分别为 a 、 b 、 c ($0 \leq a, b, c \leq 100$)，输入一个 3×3 的矩阵，其中矩阵的第 i 行第 j 列 ($1 \leq i, j \leq 3$) 记为 $M_{i,j}$ ($-100 \leq M_{i,j} \leq 100$)。若此时决定放一个第 i 种建筑物，则你会获得 $\sum_{j=1}^3 (M_{i,j} \times d_j)$ 的得分，其中 d_j 为你之前已经放下建筑物 j 的数量。

例如现在你的当前得分是 36，已经放下且仅放下 3 座第 1 种建筑物，你现在决定放一个第 2 种建筑物下去。第 2 种建筑物的特性是岛内每存在一座第 1 种建筑物，你的得分就 +1，也就是 $M_{2,1} = 1$ ，因此你现在放下一个第 2 种建筑物以后，你的得分就是 $36 + \sum_{j=1}^3 (M_{2,j} \times d_j) = 36 + 1 = 39$ 。至于第 1 种建筑物的特性，如岛内每存在一个第 2 种建筑物，你的得分就 +2，即 $M_{1,2} = 2$ ，则对你新放的第 2 种建筑物**没有影响**，即你新放的第 2 种建筑物不会因为第 1 种建筑物的特性而使你的总得分增加或减少。

已知三种建筑物你持有的数量 a 、 b 、 c ($0 \leq a, b, c \leq 100$) 和矩阵 M ，现在请你计算出将**所有建筑物**都放置在岛上的最大的得分为多少。

Input

第一行为三个非负整数 a 、 b 、 c ($0 \leq a, b, c \leq 100$)，分别为三种建筑物的个数。

接下来是一个 3×3 的矩阵 M ，代表建筑物相互之间的影响。 $M_{i,j}$ ($-100 \leq M_{i,j} \leq 100$)

第二行代表第 1 种建筑物的特性，即第 1 种建筑物分别对已存在的第 2、3 种建筑物的加分，接下来两行分别是第 2、3 种建筑物的特性。

Output

一个整数，代表将给出的**所有建筑物**放在一块空区块上的最大得分。

Examples

standard input	standard output
1 1 1 3 0 -1 2 1 0 0 0 3	2

standard input	standard output
21 1 46 -17 -12 48 7 -54 -48 -30 -42 -25	14862

Explanation

第一个样例，可以证明以下放置方式可以让你获得最大得分：

1. 放一个第 1 种建筑物，总得分为 0，因为根据第 1 种建筑物的特性，只有岛上已存在第 1、3 种建筑物才会让你的得分变化。
2. 放一个第 3 种建筑物，总得分为 0，注意虽然此时岛上已存在一个第 1 种建筑物，但并不会使你的得分减少。
3. 放一个第 2 种建筑物，总得分为 2，因为之前已经存在了一个第 1 种建筑物。

Problem C. POP TEAM shoryu

Input file: standard input
 Output file: standard output
 Time limit: 1 second
 Memory limit: 256 megabytes



POP 子和 PIPI 美的日常 是由日本漫画家大川武士创作的喜剧四格漫画作品，现已改编为动画。其中的两位主人公分别是 POP 子和 PIPI 美。在动画某集中 POP 子 和 PIPI 美作为黑帮打手前去打压黑恶势力组织并从中获取名望。PIPI 美办事冷静，能够通过合理手段获得名望，但 POP 子过于暴躁，做事比较粗糙，如果经由她手将无法获得名望，请你帮忙计算她们俩最多能够获得多少名望？

现在有 $n(1 \leq n \leq 2 \times 10^4)$ 个黑恶势力单位，分别标号为 $[1, n]$ 。第 i 个黑恶势力有一个喽啰数 p_i 和一个被打压后的名望值 $w_i(1 \leq p_i, w_i \leq 10^3)$ 。

现在已知 PIPI 美的一击秒杀人数为 $a(1 \leq a \leq 10^3)$ ，POP 子的一击秒杀人数为 $b(1 \leq b \leq 10^3)$ 。POP 子和 PIPI 美两人一起去打压这些黑恶势力。首先，POP 子和 PIPI 美去打压第一个黑恶势力直到该黑恶势力瓦解，然后她们再去打压第二个黑恶势力 ... 经受一击后，一个黑恶势力的喽啰数归零或更少即视为瓦解。

打压的每一个回合是这样进行的：

1. PIPI 美先进行打压，使用一击后黑恶势力喽啰数 p_i 相应减少 a ，如果此时该黑恶势力瓦解，PIPI 美会抓住该黑恶势力首领并进行训话，接着把他放走，这样她们就能得到 w_i 的名望值。

2. 若组织没有瓦解，POP 子将继续进行打压，使用一击后黑恶势力喽啰数 p_i 相应减少 b ，如果此时黑恶势力瓦解，暴躁的 POP 子会逮住黑恶势力首领并直接做掉，这样她们将不能获得名望值。

若一个回合结束后该组织仍未瓦解，则进行下一轮回合的打压，直到该黑恶势力瓦解。

虽然 POP 子性情暴躁，但是她很听 PIPI 美的话，因此在某一回合中，本该轮到 POP 子进行打压时，PIPI 美可以让她住手，跳过 POP 子这回合的打压，而直接开始下一个回合，PIPI 美自己进行打压以获取名声。但是在所有的打压中，她总共只能喊停 POP 子 $k(1 \leq k \leq 10^3)$ 次（比如一共喊停 5 次，现在有 2 个黑恶势力组织，PIPI 美可以在打第 1 个黑恶势力时喊停 2 次，然后在打压第 2 个黑恶势力时喊停 2 次；但是不能在打压第 1 个黑恶势力时喊停 2 次然后在打压第 2 个黑恶势力时喊停 4 次，因为这样一共就喊停了 6 次，超过了 5 次）。

你的任务是计算她们能获得的最大名望值之和。

Input

第一行为四个非负整数 n 、 a 、 b 、 k ($1 \leq n \leq 2 \times 10^4, 1 \leq a, b, k \leq 10^3$)，分别代表黑恶势力的数量，PIPI 美的一击秒杀人数，POP 子的一击秒杀人数，以及 PIPI 美可以叫停 POP 子的总次数。

第二行为数组 p ，其中 p_i ($1 \leq i \leq n, 1 \leq p_i \leq 10^3$) 代表第 i 个黑恶势力的喽啰数。

第三行为数组 w ，其中 w_i ($1 \leq i \leq n, 1 \leq w_i \leq 10^3$) 代表打败第 i 个黑恶势力可获得的名望值。

Output

一个整数，代表在有限叫停次数 k 内她们俩能获得的最大名望值。

Examples

standard input	standard output
6 2 3 3 7 10 50 12 1 8 1 1 1 1 1 1	5

standard input	standard output
9 3 6 3 6 9 3 1 5 3 1 5 7 1 9 5 9 3 7 3 6 9	39

Explanation

第一个样例，可以证明以下操作可以让她们获得最大的名望值：

对于第一个黑帮，可以直接由 PIPI 美将其瓦解而不需要喊停；

对于第二个黑帮，PIPI 美放任 POP 子直接将其瓦解，不获得名望值；

对于第三个黑帮，PIPI 美喊停 POP 子两次，获得名望值；

对于第四个黑帮，可以直接由 PIPI 美将其瓦解而不需要喊停；

对于第五个黑帮，可以直接由 PIPI 美将其瓦解而不需要喊停；

对于第六个黑帮，PIPI 美喊停 POP 子一次，获得名望值。

Problem D. Shakugan no Shana

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 512 megabytes



在某个地方有一群红世使徒肆虐人间，在他们之中，互相友好的双方可以联合抗击火雾战士的攻击，多个互相友好的使徒可以组成一个使徒集团，集团中使徒两两之间都互相友好。朋友之间一定是互相友好的，并且如果使徒 B 是使徒 A 的朋友，使徒 C 是使徒 B 的朋友，则使徒 C 与使徒 A 也互相友好，即友好关系具有传递性。但如果使徒 B 被击杀了，那么使徒 A 与使徒 C 就不互相友好了，除非他们是朋友。

举三个例子：

1. 使徒 A 与使徒 B 是朋友，使徒 B 与使徒 C 是朋友，那么使徒 A 与使徒 C 互相友好。此时使徒 B 被击杀，则使徒 A 与使徒 C 不互相友好。
2. 使徒 A 与使徒 B 是朋友，使徒 B 与使徒 C 是朋友，使徒 A 与使徒 C 也是朋友。此时使徒 B 被击杀，则使徒 A 与使徒 C 仍互相友好。
3. 使徒 A 与使徒 B 是朋友，使徒 B 与使徒 C 是朋友，使徒 C 与使徒 D 是朋友，那么他们四个使徒之间都互相友好。此时使徒 C 被击杀，使徒 A 与使徒 B 仍互相友好，但是使徒 A 与使徒 D，使徒 B 与使徒 D 不互相友好。

现在已知有 $n(2 \leq n \leq 2 \times 10^5)$ 个使徒标号为 $[1, n]$ ，以及他们间的朋友关系。每个使徒都有一个战斗力 $a_i(1 \leq i \leq n, 1 \leq a_i \leq 10^4)$ 。但他们惊恐地发现他们即将面对的火雾战士是夏娜！夏娜非常强大以至于 n 个使徒再怎么联合也是徒劳无功，所以夏娜会按照数组 $k(|k| = n, k \text{ 为 } [1, n] \text{ 的排列})$ 的顺序将使徒挨个击杀。

但是无聊的悠二想要计算一下每击杀一个使徒后，当前所剩下的使徒集团中，战斗力最大的使徒集团的战斗力是多少（使徒集团的战斗力是集团内所有使徒战斗力的总和）。

Input

第一行为一个整数 $n(2 \leq n \leq 2 \times 10^5)$ 代表使徒的数量。

第二行为数组 a ，其中 $a_i(1 \leq i \leq n, 1 \leq a_i \leq 10^4)$ 代表第 i 个使徒的战斗力。

第三行为数组 k ，其中 $k_i(1 \leq i \leq n, 1 \leq k_i \leq n)$ 代表夏娜第 i 个击杀的使徒标号。

第四行为一个非负整数 $m(1 \leq m \leq \min(\frac{n(n-1)}{2}, 2 \times 10^5))$ 代表朋友关系数量。

接下来 m 行，每行两个数字 $x, y(x \neq y, 1 \leq x, y \leq n)$ 代表标号为 x 和 y 的使徒是朋友。

Output

输出 n 行，每行一个非负整数 s_i ，代表夏娜击杀第 i 个使徒后，剩余使徒集团中战斗力最大的一个集团的战斗力。

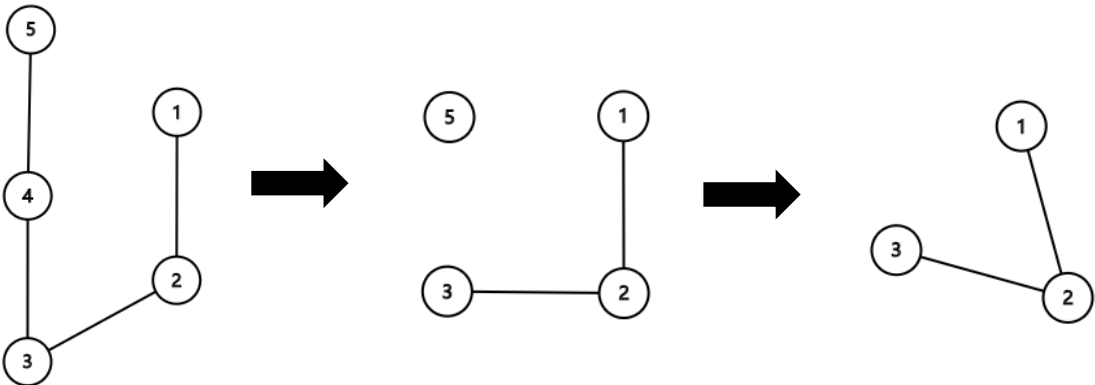
Examples

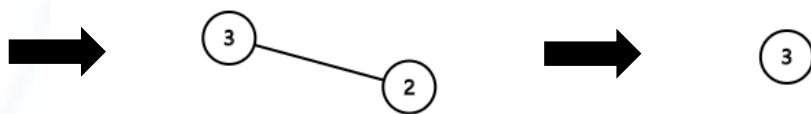
standard input	standard output
5	6
1 2 3 4 5	6
4 5 1 2 3	5
4	3
1 2	0
2 3	
3 4	
4 5	

standard input	standard output
2	2
1 2	0
1 2	
1	
1 2	

Explanation

第一个样例的使徒存活状况如下图所示：





夏娜的击杀顺序为 4、5、1、2、3，战斗力最大的集团由 $1-2-3-4-5$ 变为 $1-2-3$ ，接下来分别是 $1-2-3$ 、 $2-3$ 、 3 。

This page is intentionally left blank.

此页有意留为空白。

Problem E. daiko!!

Input file: standard input
 Output file: standard output
 Time limit: 3 seconds
 Memory limit: 512 megabytes



胖哥哥最近编程了一个音乐游戏 daiko!!，每次游戏初始得分是 1，每个圆圈上都有一个整数 x ($2 \leq x \leq 10^{114514}$)。打中圆圈会根据节拍难易度将你的得分乘以圆圈上的整数 x ，在游戏的最后会根据你的综合表现，将你的得分乘以一个实数 y ($1 \leq y \leq 10^{1919810}$)，由此你就得到了这次游戏你的总得分 z ($z \in \mathbb{Z}, 2 \leq z \leq 5 \times 10^4$)。

胖哥哥编完以后就开始试玩，他发挥稳定，能连续多天打出同一总分。但是对于一个总分，我们不知道这次游戏胖哥哥打了多少个圆圈，所以我们就当作胖哥哥打出了该总分所能打出的最大的圆圈数。

现在胖哥哥想知道他在某段时间里，他打出的某个圆圈数的总次数。

Input

第一行包含一个正整数 m ($1 \leq m \leq 2 \times 10^5$) 表示记录和询问的总条数。

接下来 m 行每行包含 4 个整数，表示记录或者询问，具体如下：

1. $p \ z \ l \ r$: 胖哥哥在 $[l, r]$ 天每天都打出了一次总得分 z 。 ($z \in \mathbb{Z}, 2 \leq z \leq 5 \times 10^4, 1 \leq l \leq r \leq 5 \times 10^4$)
2. $q \ c \ l \ r$: 查询胖哥哥在 $[l, r]$ 天一共打出了多少次圆圈数 c 。 ($c \in \mathbb{Z}, 1 \leq c \leq 5 \times 10^4, 1 \leq l \leq r \leq 5 \times 10^4$)

Output

对于每一次询问，输出一个非负整数，代表胖哥哥在所询问的天数内打出的该圆圈数次数总数。

Examples

standard input	standard output
5 p 2 1 2 p 2 2 3 p 2 3 4 p 3 4 5 q 1 1 5	8

standard input	standard output
2 p 18 1 10 q 18 1 10	0

Explanation

第二个样例，胖哥哥在第 1 天到第 10 天每天都打出了一次总分 18，但是该总分不可能是打了 18 个圆圈而得来的，因此输出 0。

Problem F. :)’s necklace

Input file: standard input
 Output file: standard output
 Time limit: 1 second
 Memory limit: 256 megabytes



:) 月底总是变得很郁闷，他昨天检查了自己的信用卡，没有任何惊喜，只剩下 66.6 元。他很绝望，想着该如何度过这个月剩下的日子。受创新创业课程的影响，他决定卖掉一些东西赚钱。当然，这没有那么容易。

经过思考，他发现出售项链是个不错的选择，众所周知，女孩喜欢五颜六色的装饰。现在他有一串带有 $n(1 \leq n \leq 2 \times 10^5)$ 个珠子的项链，珠子被标号为 $[1, n]$ 。注意，项链是环形的，因此 n 号珠子后是 1 号珠子。每个珠子都有自己的形状，我们将第 i 种形状用整数 $t_i(1 \leq t_i \leq 2 \times 10^5)$ 表示。

他想把每个珠子都涂上颜色，同时他认为如果项链上相邻的两个珠子形状不同但颜色相同，这串项链就会很廉价。因为颜料也需要钱，为了节省成本，你的任务是使用尽可能少的颜色给珠子上色使得这串项链让 :) 觉得不廉价。

Input

输入包含 $q(1 \leq q \leq 10^4)$ 组测试样例。

接下来 q 组测试样例，每组测试样例第一行为一个整数 $n(3 \leq n \leq 2 \times 10^5)$ 代表该串项链珠子的总数，第二行为数组 t ，其中 $t_i(1 \leq t_i \leq 2 \times 10^5)$ 。

所有测试点保证 $\sum_{i=1}^q n \leq 2 \times 10^5$ 。

Output

对于每个测试样例，第一行输出上色该串项链所需的最少颜色种类，第二行输出具体的上色方案，你可以用任意正整数 $c(1 \leq c \leq 2 \times 10^5)$ 代表一种新颜色，只要他之前没被你使用过。

如果有多组符合条件的输出，你只需输出其中一种即可。

Examples

standard input	standard output
2	2
5	1 2 1 2 2
1 2 1 2 2	2
6	2 1 2 1 2 1
1 2 2 1 2 2	

standard input	standard output
2	3
5	2 3 2 3 1
1 2 1 2 3	1
3	1 1 1
10 10 10	

Explanation

第二个样例，第一组测试数据，1 形状与 2 形状，2 形状与 3 形状都是不同的形状，因此不能涂相同的颜色，同时注意首尾颜色也不能相同。第二组测试数据形状都相同，因此可以涂相同的颜色。

Problem G. Overfly

Input file: standard input
 Output file: standard output
 Time limit: 3 seconds
 Memory limit: 512 megabytes



以有着“精灵国度”意义的 ALfheim 大陆为舞台的次世代飞行型 MMO 发售了，胖哥哥和 Jyo 在发售之际就连夜排队抢到了游戏软体，加入了以飞行速度与听力见长的风精灵阵营。

ALfheim 中央耸立着所有玩家的最终目的地“世界树”，而世界树根部则有大陆最大的都市：央都“阿伦”。世界树”上面据说有传说中的空中都市，而如果有种族能够谒见住在那里面的妖精王“奥伯龙”，就可以转生为高等种“光之精灵”。

现在风精灵领主朔夜在经过与猫精灵领主的会谈后，紧急通知胖哥哥和 Jyo 迅速将物资从风精灵首都“司伊鲁班”集结到“阿伦”，准备攻打“世界树”，这可把胖哥哥和 Jyo 激动坏了。

朔夜塞给了胖哥哥和 Jyo 一张 ALfheim 的地图，上面记载了 ALfheim 的所有 $n(1 \leq n \leq 10^4)$ 个城市，标号为 $[1, n]$ ，其中“司伊鲁班”标号为 1，“阿伦”标号为 n ，以及 $m(1 \leq m \leq 10^5)$ 条单向路和道路限制：携带物资通过这条路的最大物资运载量 $a(1 \leq a \leq 100)$ 及单位物资运输费用 $b(1 \leq b \leq 100)$ 。不过这张地图是游戏初期的地图了，现在已知如果两个城市能互相到达，则就会归属于同一属性阵营，若不能互相到达则不属于同一阵营。

要将物资从“司伊鲁班”运送至“阿伦”，途中就要经过不少其他城市，若通过单向路从一个城市将物资运送至另一个城市且这两个城市同属一个属性阵营，运送就没有任何限制，也就是最大物资运载量为无穷大，单位物资运输费用为 0，但是跨阵营之间的道路限制依然存在。

朔夜积攒的物资有很多，你可以视为无穷大，但是由于道路限制只能运送一部分，且因为时间紧迫只能允许运输一次，也就是出发后就不能返回“司伊鲁班”，另外你可以在运送途中将物资和运输部队进行划分，让大家分头运输，最后在“阿伦”集结即可。朔夜要求在将尽可能多的物资运送到“阿伦”的前提下，所花费用最小。虽说朔夜在会谈后带回一大笔钱，但添置装备已经花费不少，运送的钱就需要尽可能节省。

Input

第一行是两个正整数 n 和 m ，代表地图上大陆的城市总数和单向路总数。 $(2 \leq n \leq 10^4, 1 \leq m \leq \min(\frac{n(n-1)}{2}, 10^5))$

接下来 m 行，每行有四个整数 x 、 y 、 a 、 b ，代表 x 到 y 存在一条单向路，最多可以携带 a 个单位的物资通过，每单位物资需花费 b ，注意，这些道路限制都是地图上的内容，而现在受到阵营影响，每条路具体的道路限制可能发生变化。（ $1 \leq x, y \leq 10^4$ ， $1 \leq a, b \leq 100$ ）

题目同时保证阵营数不会超过 5000，不同阵营之间的单向路数量不会超过 10000，且“阿伦”所在阵营不存在其他城市，至少有一条路线可从“司伊鲁班”到达“阿伦”。

Output

输出两个数，第一个是所能运送的物资最大值，第二个是在该最大值下所花运输费用的最小值，中间用空格隔开。

Examples

standard input	standard output
5 6 1 2 1 1 2 3 1 1 3 4 6 9 4 2 8 10 1 5 2 1 4 5 1 1	3 4

standard input	standard output
2 1 1 2 39 39	39 1521

Explanation

第一个样例，城市 2、3、4 同属一个阵营。从“司伊鲁班”兵分两路，一路从城市 2、3、4 所属阵营穿过然后到达“阿伦”，另一路可通过一条单行路直接到达“阿伦”。

Problem H. Law of Cycles

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes



圆环之理究竟是什么？这是一个很好解答，也很不好解答的问题。

圆神许愿消灭过去、现在、与未来的魔女，改写了整个宇宙规则，魔法少女在能量耗尽的时候不会变成魔女，而是被圆神接到另一个更高层次的位面，在其他人的眼中魔法少女将死时会消失，这种现象称作圆环之理。

同样因为圆神的愿望，圆神将魔女的自己消灭，因此她也不存在于地球的空间中，她的存在被抹去，除了焰以外，没有人记得她，她变成了一个概念，在不同的时空中往返，迎接那些等待救赎的魔法少女，换句话说，她把自己变成了魔法少女的希望。

所以说，圆环之理是一种宇宙的法则，是一种自然的现象，是一种假定的，超越作为表象的世界（时间、空间、因果律）的宇宙秩序。我们可以从先哲的话语里找到此类说法，如毕达哥拉斯的灵魂不灭，柏拉图的伟大年（天文学中称柏拉图年），印度宗教中的涅槃等。尼采在他的散文诗《查拉图斯特拉如是说》中假托琐罗亚斯德教的创始人查拉图斯特拉之口也道出了类似的思想。原作中并没有对圆环之理做过多的解释，只能靠我们自己理解，这或许正是其作为艺术的迷人之处吧。

想要理解圆环之理，必须要先搞懂圆环，所以你的任务就是在平面直角坐标系中，对于给定的圆心与该圆上某个切点，求出该圆的周长。

Input

输入一共有 2 行，第一行为该圆的圆心 O 坐标 $(-5 \leq x_o, y_o \leq 5)$ ，第二行为该圆上一切点 P 的坐标 $(-5 \leq x_p, y_p \leq 5)$ ，输入保证圆心坐标不与切点坐标相同。

横纵坐标间用空格隔开，输入为两位小数。

Output

一个浮点数代表该圆周长，**结果保留两位小数**。

Examples

standard input	standard output
1.9 -1.09 0.54 0.15	11.56

standard input	standard output
1.00 0.00 0.00 0.00	6.28

Explanation

圆的周长公式为 $C = 2\pi r = \pi d$ ，其中 π 为圆周率， r 为圆的半径，而 d 为圆的直径。

Problem I. No repetition

等着禁言吧

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes



:) 最近发现群内存在有很多人经常复读 wtcl（我太菜了）的现象，为了杜绝这种不良现象，:) 决定给予复读者一些禁言处罚。

对于第一个发 wtcl 的人，:) 不对其进行禁言，但第一个复读的人，即第二个发 wtcl 的人，:) 会对其禁言 5 分钟；对于第二个复读的人，:) 会对其禁言 10 分钟，以此类推。

更正式的，对于第 i 个发 wtcl 的人，会被禁言 $(i - 1) * 5$ 分钟。

现在给出一段复读的聊天记录，:) 想知道最后一个复读者将会被禁言多少分钟。

Input

输入包含 $q(1 \leq q \leq 10)$ 组测试样例。

接下来 q 组测试样例，每组测试样例仅有一行，包含一串字符串 s 代表复读的聊天记录。（ $4 \leq |s| \leq 10^5$ ）

输入保证字符串 s 仅包含 w, t, c, l 四个小写字母且长度为 4 的倍数，另外不会出现不连续或复读不正确的情况，即不会出现诸如 $wtclw, wctlwt, wtcltwc, wtclwtlc$ 等情况。

Output

对于每个测试样例，输出一个非负整数，代表最后一个复读者被禁言的时间。

Examples

standard input	standard output
1 wtcl	0

standard input	standard output
2 wtclwtcl wtclwtclwtcl	5 10

Explanation

第一个样例，该人没有复读，所以不需要禁言。

第二个样例，第一组测试数据，共有 1 人复读，因此复读的最后一个人禁言 5 分钟，第二组测试数据，共有2 人复读，因此复读的最后一个人禁言 10 分钟。

Problem J. Cirno's perfect math class

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes



琪露诺的完美算数课堂开课了！

琪露诺给大家出了一道找规律题：

21 36 43 __

但是大家认为“数字找规律”没有任何意义可言，因为有方法使一个给定数列的下一个数字是任何一个他想要的数字，比如⑨。例如，当 $f(x) = -5.5x^3 + 29x^2 - 33.5x + 31$ 时：

$$f(1) = 21, f(2) = 36, f(3) = 43, f(4) = 9$$

于是大家想要每次出规律题的时候都能随便填数字，比如都填⑨，但是当琪露诺问起来的时候，必须用一个多项式函数证明这么填是对的，大家不想每次都要计算这个多项式，请你帮大家做一个程序能直接根据大家输入的数列和大家想要的数字计算得出对应的多项式。

Input

第一行为一个正整数 $n(1 \leq n \leq 6)$ 代表数列长度。

第二行为 $n + 1$ 个整数，前 n 个数字为琪露诺给出的提问整数数列 a ，其中第 $i(1 \leq i \leq n)$ 个数字 $a_i(|a_i| \leq 100)$ ，而第 $n + 1$ 个整数 $b(|b| \leq 2 \times 10^6)$ 为大家想要填的数，中间用空格隔开。

Output

输出 $n + 1$ 行，第 $i(1 \leq i \leq n + 1)$ 行的小数代表多项式第 $n + 1 - i$ 次项的系数，输出**保留六位小数**。

Examples

standard input	standard output
3 21 36 43 9	-5.500000 29.000000 -33.500000 31.000000

standard input	standard output
4 1 3 5 7 114514	4771.041667 -47710.416667 166986.458333 -238550.083333 114504.000000

Explanation

第二个样例，对于

$$f(x) = 4771.041667x^4 - 47710.416667x^3 + 166986.458333x^2 - 238550.083333x + 114504.000000$$

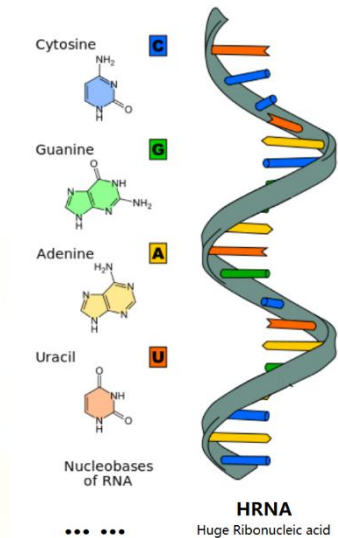
可以近似得到：

$$f(1) = 1, f(2) = 3, f(3) = 5, f(4) = 7, f(5) = 114514$$

Problem K. Searching for HRNA

Input file: standard input
 Output file: standard output
 Time limit: 0.5 second
 Memory limit: 512 megabytes

近期在艾勒比拉星球，爆发了一轮新型冠状病毒。生物学家发现该病毒与传统 RNA 病毒有很大不同，该病毒不仅有 A、U、C、G 四种核苷酸，还存在新发现的 B、D、E、F 等新型核苷酸，因此被称之为 HRNA（Huge RNA）病毒。



HRNA 病毒的核苷酸种类总量已查明，共有 26 种核苷酸，用大写字母 A 到 Z 表示，且其与其子代的基因序列可以用由大写字母组成的字符串 s ($1 \leq |s| \leq 1000$) 表示，我们将字符串的第一个字母称之为基因序列的**链首**。

HRNA 病毒和普通 RNA 病毒一样，会从**链首**开始按照基因序列对应的字符串的顺序复制自己的基因序列，然后创造出一个子代，该行为被称之为遗传。

HRNA 病毒及其子代在进行遗传时，可能会**遗漏后缀** t ($1 \leq |t| \leq |s| - 1$)，HRNA 病毒子代因为遗漏了遗传信息，因此**不能叫做** HRNA 病毒。

字符串的后缀是指字符串的任意尾部。例如对于字符串 *AUFODSJ*，有 *J*、*DSJ*、*AUFODSJ* 是其后缀，而 *FDJ*，*AUFJ* 则不是其后缀。

生物学家最近获得了一批病毒样本，该样本中包含了 HRNA 病毒及其子代的基因序列（也许没有子代的基因序列），你的任务是输出所有可能是 HRNA 病毒的病毒基因序列。

Input

输入第一行为一个正整数 n ($1 \leq n \leq 1000$)，表示这批样本中 HRNA 病毒及其子代的基因序列数量。

接下来 n 行是 n 个仅由大写字母组成的基因序列 s ($1 \leq |s| \leq 1000$)。

Output

按照字典序从小到大的顺序，每行输出一个 HRNA 病毒的病毒基因序列。注意，若有多组 HRNA 病毒的病毒基因序列相同，则只需输出一个。

Examples

standard input	standard output
1 CORONAVIRUSDISEASE	CORONAVIRUSDISEASE

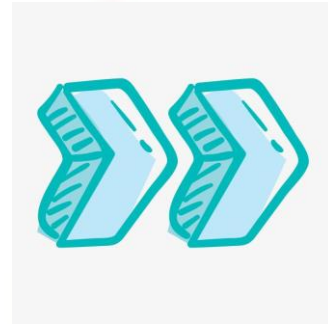
standard input	standard output
5 ABCDEFGHIJK BUBUBUU BUBUBUU ABCDE QWERTY	ABCDEFGHIJK BUBUBUU QWERTY

Explanation

第二个样例，*ABCDE* 明显是 *ABCDEFGHIJK* 遗漏了后缀 *FGHIJK* 而产生的后代，他一定不可能是 HRNA 病毒，而其他字符串代表的病毒都有可能是 HRNA 病毒，因为样本中没有其他病毒的基因序列可证明他们是子代。

Problem L. “<<” and “>>” operation

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes



那天本是一个 happy 的日子,:) 开心得在计算机组成原理课上刷着题。不幸的是老师刚好提问到他, 因为写的太认真,:) 在快下课前才看到。

老师很生气便留给他一道和位运算有关的题, 希望他编程解决, 如果答错平时分只有 60 分 (这意味着他必须考到 60 分才能及格, 这对他来说太难了)。题目是这样的:

给定你两个 64 位无符号整数 $x, y (0 \leq x, y \leq 10^{18})$, 问你能否仅通过对 y 进行 **逻辑左移** 与 **逻辑右移** 操作, 使得 $y = x$ 。

逻辑右移是一种移位运算, 对 y 进行逻辑右移的做法是将 y 转为二进制以后向右移动, 高位补 0。如对于 64 位无符号整数 11, 转化成二进制为 000...0001101, 则逻辑右移一位即为 000...0000110, 逻辑右移两位即为 000...0000011。逻辑左移同理, 将 64 位无符号整数 11 逻辑左移 61 位得到 101000...000。

这可难倒了 :), 所以他决定求助于聪明又善良的你, 相信你一定有办法。

Input

输入包含 $q (1 \leq q \leq 10^6)$ 组测试样例。

接下来 q 组测试样例, 每组测试样例仅有一行, 包含两个正整数 $x, y (1 \leq x, y \leq 10^{18})$ 。

Output

对于每个测试样例, 若 y 能通过**不限次数**的逻辑左移和逻辑右移变成 x , 则输出 YES, 否则输出 NO。

Examples

standard input	standard output
2	YES
2 3	YES
3 11	

standard input	standard output
5	YES
12 3	NO
12 5	YES
3 12	YES
10 5	YES
0 1	

Explanation

第一个样例，第一组测试数据，64 位无符号整数 3 的二进制表示为 $000 \dots 00011$ ，64 位无符号整数 2 的二进制表示为 $000 \dots 00010$ ，则 $y = 3$ 逻辑右移一位后变为 $000 \dots 0001$ ，再逻辑左移一位即变为 $x = 2$ 。第二组测试数据，11 的二进制表示为 $000 \dots 0001011$ ，则不断逻辑左移后能将最左侧的 1 溢出，然后不断进行逻辑右移即可得到 $x = 3$ 。