

Problem A. Robin 的字符串压缩

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

在 Robin 的世界里, 过长的字符串会扰乱和谐. 为此, 人们发明了一种字符串压缩术 [RC] — 若字符串 s 的长度超过阈值 k , 会将其压缩成:

“首字符 + 中间字符数 + 尾字符”.

以此融入天地间的混沌美学.

例如, 字符串 “internationalization” 经操作 [RC] 后可能变为 “i18n”.

现给你 n 个字符串和阈值 k , 希望你对这些字符串完成操作 [RC].

Input

第一行输入两个整数 n ($1 \leq n \leq 100$) 和 k ($1 \leq k \leq 100$),

接下来 n 行, 每行输入一个字符串, 只包含小写字母, 长度在 1 到 100 之间.

Output

输出 n 行, 输出每个字符串经操作 [RC] 后得到的字符串.

Examples

standard input	standard output
5 8 beijing normal university artificial intelligence	beijing normal u8y a8l i10e
5 4 apple internationalization honorificabilitudinitatibus a icpc	a3e i18n h25s a icpc

Problem B. Invisible String

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

Time.
Wondrous time.
Gave me the blues and then purple pink skies.
And it's cool. Baby with me.
And isn't it just so pretty to think.
All along there was some.
Invisible string.
Tying you to me.

Taylor Swift, *invisible string* - folklore

小 x 有一个长度为 n 的只包含小写字母的字符串, 他想知道在这个字符串后面添加 m 个小写字母后得到的新字符串能包含多少个本质不相同的子序列, 包含本质不相同的子序列的数量最多的新字符串被称为 “invisible string”. 请你找出这个 “invisible string”, 并且给出它的本质不相同的子序列的数量, 答案对 998244353 取模.

注意, 答案不一定唯一, 如果有多组答案, 请任意输出一组即可.

子序列是从原字符串中删除零个或多个字符 (不改变剩余字符的相对顺序) 得到的非空序列. 例如, 对于字符串 “abc” 来说, “ac”, “ab”, “abc” 都是它的子序列, 但 “ca” 不是. 本质不相同的子序列是原字符串的表示不同字符串的子序列. 例如, 对于字符串 “abb” 来说, 删除中间的 “b” 与删除最右侧的 “b” 都会得到子序列 “ab”, 由于最终得到的子序列是相同的, 即使删除的字符不一样, 这两个子序列 “ab” 并不是本质不相同的.

Input

第一行输入两个数 n, m ($0 \leq n, m \leq 10^6$), 分别表示已有字符串的长度和需要添加字母的数量.

如果 $n = 0$, 则结束输入, 否则在第二行输入一个长度为 n 的字符串 s .

Output

第一行输出一个字符串, 表示找到的 “invisible string”.

第二行输出一个数, 表示找到的 “invisible string” 所包含的本质不相同的子序列的数量, 答案对 998244353 取模.

Examples

standard input	standard output
4 0 abca	abca 14
0 3	nzy 7

Problem C. Replay

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

众所周知, 魔塔是一款益智类小游戏. 玩家需要通过操控勇士移动, 最终击败魔王. 而 H5 魔塔网站上为玩家提供了录像回放功能, 即用玩家的操作序列推出最终结果. 作为一个不玩魔塔的人, 你需要为你的一位朋友实现这一功能.

魔塔的地图一般为一个 $n \times n$ 的矩形, 共有 m 层, 每一格均为道具, 墙, 怪物, 宝石, 门, 钥匙, 血瓶, 空地, 上楼或下楼之一.

当然, 为了简化问题, 我们只考虑单层塔, 即 m 等于 1 的情况, 此时不存在上楼与下楼, 且我们不考虑道具.

勇士拥有四个属性, 生命 H , 攻击 A , 防御 D 与魔防 M . 第 i 种怪物拥有三个属性, 生命 H_i , 攻击 A_i 与防御 D_i . 勇士首先攻击, 然后双方交替出手, 勇者每次造成 $\max(A - D_i, 0)$ 点伤害, 怪物每次造成 $\max(A_i - D, 0)$ 点伤害. 而魔防可以在战斗中抵御等量的伤害, 每场战斗单独计算. 更形式地, 若某场战斗中你受到的总伤害为 S (未计算魔防时), 你的魔防为 M , 则你受到的实际伤害为 $\max(S - M, 0)$.

血瓶分为四种, 红血瓶, 蓝血瓶, 黄血瓶, 绿血瓶, 均用于增加血量.

宝石分为三种, 红宝石, 蓝宝石, 绿宝石. 红宝石增加攻击, 蓝宝石增加防御, 绿宝石增加魔防.

门与钥匙均有三种, 分别为黄, 蓝, 红. 一把钥匙可以打开一把对应的门.

勇士有四种移动方式, 即向上移动, 向下移动, 向左移动, 向右移动.

勇士移动时, 若移动方向的下一格为空地, 则移动到这一格. 否则, 触发对应操作 (即拾取物品、开门或发生战斗) 后再移动到这一格, 移动到该格后, 该格即变为空地.

若勇士最终击败了魔王, 则输出 Win. 若勇士最终没有击败魔王, 则输出 Record ends. 若勇士在操作中执行了不合法操作, 则输出 Record error.

勇士初始拥有 100 点血量, 10 点攻击, 10 点防御, 0 点魔防.

击败魔王后即判断为胜利, 无论录像是否结束.

Input

第一行两个整数 n, k ($1 \leq n \leq 500, 1 \leq k \leq 99$), 分别为楼层大小, 楼层数目, 以及怪物种数. 保证 n 为奇数.

接下来 n 行, 每行 $2n$ 个字符. 为整个单层塔的地图. 每一格占两个字符. 格式如下:

1. ##: 代表墙;
2. ...: 代表空地;
3. st: 代表勇士的起始位置;
4. YK BK RK: 分别代表黄, 蓝, 红钥匙;
5. YD BD RD: 分别代表黄, 蓝, 红门;
6. Rx Bx Yx Gx: 分别代表红, 蓝, 黄, 绿血瓶, x 为血瓶的等级, 分别可以回复 $100x, 250x, 500x, 1000x$ 点血量;

7. **rx bx gx**: 分别代表红, 蓝, 绿宝石, x 为宝石的等级, 红宝石可以增加 x 点攻击力, 蓝宝石可增加 x 点防御力, 绿宝石可增加 $3x$ 点魔防;

8. **xx**: 这是一个两位整数, 代表怪物, 其中魔王为编号为 k 的怪物, 输入保证怪物的编号从 01 开始, 至 k 结束, 且编号为 k 的怪物有且仅有一个.

接下来 k 行, 第 i 行每行 3 个整数, 为 H_i, A_i, D_i ($0 \leq A_i, H_i, D_i \leq 10^5$).

接下来一行一个字符串, 为勇士的操作序列, 每个字符代表一个操作. **w s a d** 分别代表向上, 下, 左, 右移动, 保证操作序列的长度不超过 3×10^5 .

操作方式中未提到的行为即为不合法, 包括但不限于没有钥匙时开门, 向墙的位置移动, 勇士无法击败怪物或者击败后生命小于等于 0 等.

数据保证怪物的编号从 01 开始, 至 k 结束, 且编号为 k 的怪物有且仅有一个.

Output

一行一个字符串, 为 **Record ends**, **Win**, **Record error** 之一.

Examples

standard input	standard output
3(3) 010203 02#### 0101st 10 100000 0 100000 10 0 101 20 0 aawwdd	Record error
5 4 04030203r9 0302RD##03 0201..##b9 03##R101RD b90302RKst 20 300 0 20 50 0 20 120 0 20 20 0 aawwwa	Record ends

Note

对于样例 1, 勇士在最后一步操作中, 与魔王战斗, 战斗后血量小于 0, 故不合法;

对于样例 2, 勇士执行所有操作后停止, 未能击败魔王.

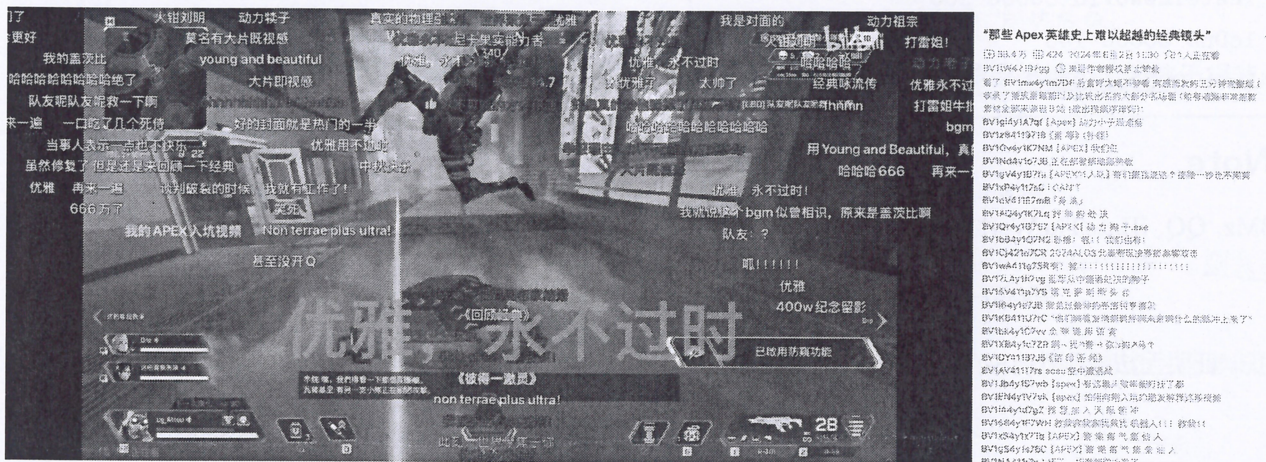
Problem D. Apex Legends

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

失败意味着你努力过了, 亲爱的.

Horizon

作为世界上最好玩的大逃杀游戏, Apex Legends 以其独特的魅力赢得了玩家们的青睐.



Apex Legends 优雅永不过时

在策划的谜之操作下, 日活跃用户数量显著回升. 备受玩家们关注的不仅是“战斗爽”的回归, 还有四支 cn 战队于大年初一远赴日本札幌市参加的备受瞩目的冠军赛. 其中, 由 Kaaasa 率领的 VKG 战队脱颖而出, 夺得了世界第九的佳绩. 然而, 并非所有战队都能一帆风顺, 由 3Mz 带领的 DF 战队却未能延续其在曼海姆季后赛和星空杯中的出色表现, 由于配合上的混乱, 他们在第一轮淘汰赛中遗憾出局.

此次经历让人们深刻意识到, 在国际赛场上, 团队的综合实力远非一两个选手的个人能力所能支撑, 团队成员之间的默契配合才是取得胜利的关键. 现在, 我们有 n 名 pro 选手, 每位选手都拥有两个关键属性: 个人能力值 a 和配合值 b , 这两个属性均为正整数. 当一名选手加入战队时, 他的个人能力值将直接提升战队的实力. 不仅如此, 当两名不同的选手同时在队中时, 他们之间产生的配合效果还将额外为战队贡献等同于他们配合值乘积的实力值.

你的任务是从这 n 名职业选手中精心挑选出三人, 组成实力最强的三人小队, 代表 cn 战队征战 Year 5 的常规赛和季后赛. 请输出这三名选手的 id, 以空格分隔. 保证选手的 id 两两不同.

注意, 答案可能不唯一, 输出任意一种即可.

Input

第一行输入一个正整数 n ($3 \leq n \leq 100$) 表示有 n 位 pro 选手.

接着输入 n 行, 第 i 行包含一个字符串 s_i 表示选手的 id, 以及两个数 a_i, b_i ($1 \leq a_i \leq 10^{18}, 1 \leq b_i \leq 10^9$) 分别表示选手的个人能力值和配合值, 均用空格隔开. 保证每一个字符串均由数字, 大小写英文字母组成, 且满足 $2 \leq |s_i| \leq 20$.

Output

输出一行, 包含三个字符串, 用空格隔开.

如果存在多种组合, 输出任意一种即可.

$n=100$

Example

standard input	standard output
6 3Mz 25000 200 Pite 100 10 QQ 45000 180 iHateTheWorld 50000 100 lqDuD 30000 135 Lvvy 20000 125	3Mz QQ iHateTheWorld

Note

3Mz, QQ, iHateTheWorld 三人所组成的队伍实力值为:

$$25000 + 45000 + 50000 + 200 \times 180 + 200 \times 100 + 180 \times 100 = 194000,$$

可以证明无法找到比 194000 更高的组合方式.

Problem E. 灾难! fufu 大营救

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

fufu 国的结构是一棵无向有根树，根节点处为首都，编号为 1。除根节点外，树上的每个节点代表一个城邦，每个城邦都有一位 fufu 城主，并且各城邦的节点编号互不相同，每个节点都跟其他若干个节点相连。根节点有一个传送阵，通过传送阵能瞬间离开 fufu 国。

在公元 2024 年 12 月 19 日，黑魔法势力大规模进攻 fufu 国。fufu 国依靠魔法屏障苦苦支撑，在三个月的时间里，将所有城邦的平民全部成功撤离，同时部分 fufu 城主也跟随平民一起撤离了。现在，黑魔法势力突然攻破了魔法屏障，他们将不断吞噬 fufu 国的领土，而此时仍有一些 fufu 城主留在城邦内，局势危急！

危急时刻，国王 slld 派遣传奇魔法使 CZR 去营救这些 fufu 城主，目标是尽可能多地将她们带回位于根节点的传送阵。

- 每天早晨，CZR 可以调动一次自身魔力，传送到一个与他当前所在节点相连的节点。
- 每天夜晚，在当前未被吞噬的节点中，离根节点最远的节点将会被吞噬（若这样的节点存在多个，则会同时被吞噬）。
- CZR 在传送时可以携带最多一位 fufu 城主一起传送，如果超过一位，他将由于身体孱弱无法施展传送魔法。
- 由于 fufu 城主行动速度非常缓慢，她们只能留在各自所在的节点中等待救援。
- CZR 从根节点立即出发，出发时间为早晨。

Input

第一行输入一个整数 n ($1 \leq n \leq 10^5$)，表示这棵树有 n 个节点，其中根节点编号为 1。

接下来 $n-1$ 行，第 i 行输入两个整数 u_i 和 v_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i$)，表示节点 u_i 和节点 v_i 之间存在一条无向边，边权均为 1。

第 $n+1$ 行输入一个整数 m ($0 \leq m < n$)。

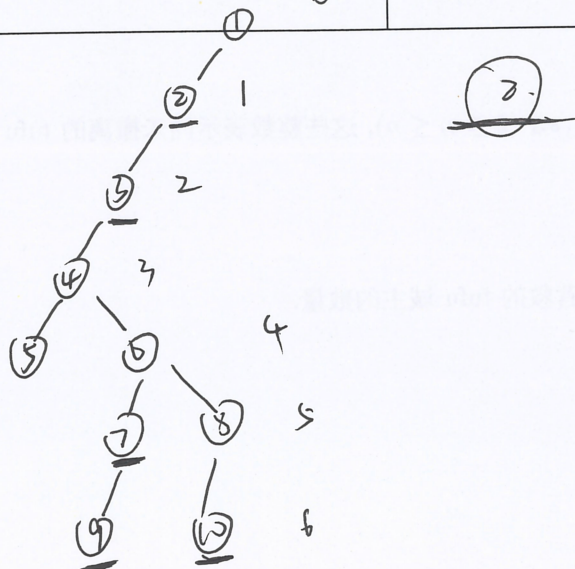
接下来一行输入 m 个不同的整数 $s_1, s_2, \dots, s_m$ ($2 \leq s_i \leq n$)，这些整数表示尚未撤离的 fufu 城主所在的节点编号。

Output

输出一个整数，该整数代表 CZR 最多能够营救的 fufu 城主的数量。

Examples

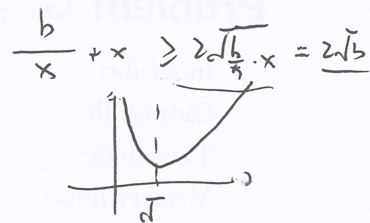
standard input	standard output
5 1 2 1 3 4 1 3 5 1 3	1
6 1 2 1 3 4 1 3 5 6 5 1 5	0
10 1 2 2 3 3 4 5 4 6 4 7 6 8 6 9 7 10 8 5 2 3 7 9 10	2



$$a+b \geq 2\sqrt{ab}$$

Problem F. MathMagic

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



小北是一名数学爱好者, 最近他迷上了一个奇怪的数学式子:

$$\frac{a_i}{a_j} + a_j$$

$$\frac{a_i}{a_j} + a_j \quad (i \neq j).$$

他手中有一个长度为 n 的正整数数组 a , 但是这些数字排列得十分混乱. 小北想知道, 在所有满足 $i \neq j$ 的情况下, 这个式子的最小值是多少?

你能帮帮他吗?

$$\frac{b}{x} = x$$

$$x = \sqrt{b}$$

Input

第一行输入一个整数 n ($2 \leq n \leq 10^5$), 表示正整数数组 a 的长度;

接下来输入 n 个整数, 表示 $a_0, a_1, \dots, a_{n-1}$ ($1 \leq a_i \leq 10^6$).

Output

输出一个浮点数 x , 表示最小的 $\frac{a_i}{a_j} + a_j$ ($i \neq j$).

设真实结果为 ans , 则您的输出结果 x 视为正确的判定条件为: $\min(|x - ans|, \frac{|x - ans|}{ans}) \leq 10^{-6}$.

Example

standard input	standard output
5 1 2 3 4 5	2.5000000000

$$\frac{1}{2} + 2 = 2.5$$

$$\frac{b}{x} = x \quad \boxed{x = \sqrt{b}}$$

$$\frac{2}{3} + 3 = \boxed{\frac{19}{3}}$$

Problem G. 星盾计划

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 MB

在星际联邦与反物质军团的持久战中, 科学家发现了一种关键战略资源——量子结晶矿. 每种矿石 (编号为 $1 \sim n$) 在自然状态下具有固定的储量, 记为 $a_1, a_2, \dots, a_n$. 矿石分为两类, 分别为单态矿石和双态矿石 (在输入中给出编号, 其余都为单态矿石), 这些矿石是制造能量护盾的核心材料. 能量核心包是构建量子共振屏障的基本单元, 用于生成防御敌方反物质侵蚀的护盾网络节点. 联邦需通过开采量子结晶矿来组装满足特定条件的能量核心包. 具体的, 在战争持续的 q 日中, 由于每日每种矿石的储量都会恢复至初始状态, 每天你需要处理一个指令 (l, r, k) , 计算当日最多可装配的能量核心包数量. 每个能量核心包需满足:

- 仅使用编号在 $[l, r]$ 区间内的矿石种类;
- 每种双态矿石在单个包中的使用量 ≤ 2 , 每种普通矿石使用量 ≤ 1 ;
- 包内矿石总数严格等于 k .

Input

第一行包含三个整数 n, m, q ($1 \leq n, q \leq 10^5, 0 \leq m \leq n$).

第二行有 n 个整数, 分别表示 $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^5$).

第三行是 m 个整数, 它们表示双态矿石的编号, 保证这些编号保证互不相同.

接下来的 q 行中, 每行有三个整数 l, r, k ($1 \leq l \leq r \leq n, 1 \leq k \leq 10^8$).

Output

输出 q 行, 每行对应一次查询的结果.

Example

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

Note

对于第一天查询, 能够组成 4 个包裹, 一种可行组合为 $(2, 4, 4), (2, 4, 4), (2, 3, 4), (2, 3, 4)$, 其中 2 号和 4 号双态晶石可在一个包中出现 2 次, 3 号普通矿石只能出现一次.

对于第二天查询, 无法组成合法包, 因 4 号矿石在一个包中最多使用 2 次, 当前情况不满足条件.

Problem H. Betty

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

Betty, one time,
I was riding on my skateboard when I passed your house.
It's like I couldn't breathe.

Taylor Swift, *betty - folklore*

James 无意中惹 Betty 生气了. 为了赢回 Betty 的欢心, 他试图通过在 Betty 家门口的鹅卵石小径上向院子里投掷石子来引起她的注意. 神奇的是, James 扔出石子的运动轨迹并不是抛物线, 而是一段圆弧!

现在, James 对这个圆弧轨迹的长度产生了浓厚的兴趣. 他想要计算出石子在空中飞行运动轨迹的长度. 更具体地, 在二维平面中给出圆弧两侧的端点坐标 (x_1, y_1) 与 (x_2, y_2) , 以及圆弧到这两个端点所在直线的最大垂直距离. 特别需要注意的是, 这两个端点都是整数坐标点, 并且该垂直距离也是一个整数值.

请你帮助 James 计算圆弧的长度.

Input

先输入一行, 包含两个数 x_1, y_1 , 分别表示圆弧一个端点的 x, y 坐标.

再输入一行, 包含两个数 x_2, y_2 , 分别表示圆弧另一个端点的 x, y 坐标.

最后再输入一行, 包含一个正整数 h , 表示圆弧最高点到这两个端点所在直线的距离.

保证 $-10^3 \leq x_1, x_2, y_1, y_2 \leq 10^3$, h 是 int 范围.

Output

输出一个浮点数, 表示圆弧的长度.

设真实结果为 ans , 则您的输出结果 x 正确的判定条件为: $\min(|x - ans|, \frac{|x - ans|}{ans}) \leq 10^{-6}$.

Example

standard input	standard output
-1 0 1 0 1	3.1415926536

Note

题目中的圆弧均指劣弧, 即圆心角不超过 180 度的圆弧.

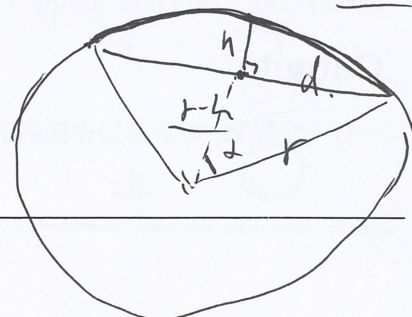
为使误差容易被控制, 数据保证圆弧所对应的圆心角不小于 10 度.

$$\begin{aligned} d^2 + (r-h)^2 &= r^2 \\ d^2 + r^2 - 2rh + h^2 &= r^2 \\ d^2 + h^2 &= 2rh \\ r &= \frac{d^2 + h^2}{2h} \end{aligned}$$

$$\sin \frac{\alpha}{2} = \frac{d}{r}$$

$$2r \cdot \frac{d}{2r}$$

$$dr$$



Problem I. Generals

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 1024 megabytes

generals.io 是一个网页游戏, 游戏地图是一个网格图, 你可以操纵我方的一个军队, 每秒可以移动, 你的军队有一个兵力数.

在这个题目中, 我们考虑一个经过简化的版本:

有一个 $n \times m$ 的地图, 地图上有以下这些元素:

- 障碍: 你的军队无法通过这个格子;
- 我方领地: 这个格子上有我方的兵力;
- 敌方领地: 这个格子上有敌方的兵力;
- 敌方防御塔: 这个格子有敌方的兵力, 且每秒兵力会增加 1.
- 敌方大本营: 这个格子是敌方的大本营, 其初始有一定的兵力, 而且每秒兵力会增加 1.

你扮演的将军操纵着一个军队, 军队有一个兵力值. 这位将军比较急躁, 急着冲到敌方大本营, 所以军队每秒必须向右或下移动一格, 不能不移动.

如果一个格子上有敌方的兵力, 设兵力为 c , 你操纵的军队移动到这个格子时, 兵力会减少 $c + 1$, 如果兵力减少之后小于零, 那么你就失败了.

如果一个格子上有我方的兵力, 设兵力为 c , 你操纵的军队移动到这个格子时, 兵力会增加 $c - 1$.

在本题中, (i, j) 表示第 i 行第 j 列的格子, 编号均从 1 开始. 我方军队的初始位置是 $(1, 1)$, 敌方大本营的位置是 (n, m) , 你需要移动军队, 目标是成功到达敌方大本营, 输出初始你操纵的兵力最少是多少才能达到目标. 注意, 每秒里, 大本营和防御塔的兵力先增长, 你再移动.

Input

第一行包括两个整数 n, m ($1 \leq n, m \leq 1000$), 表示地图的行数和列数.

接下来 n 行, 每行 m 个整数, 其中 $c_{i,j}$ ($-10^9 \leq c_{i,j} \leq 10^9$) 的绝对值表示坐标 (i, j) 的初始兵力数, 如果 $c_{i,j} = 0$, 那么这个格子是障碍. 如果 $c_{i,j} > 0$, 这个格子是我方领地, 如果 $c_{i,j} < 0$, 这个格子是敌方领地. 保证 $c_{1,1} = 1$.

接下来一行包括两个整数 q ($0 \leq q \leq nm$), 表示敌方防御塔的个数.

接下来 q 行, 每行两个正整数 x, y ($1 \leq x \leq n, 1 \leq y \leq m$) 表示一个敌方防御塔的坐标.

保证我方军队和敌方大本营可以相互到达. 保证每个防御塔的位置, $(1, 1), (n, m)$ 都不是障碍, 一个格子不会同时存在我方单位和敌方单位. 敌方防御塔一定在敌方领地, 敌方防御塔不会在 $(1, 1), (n, m)$.

Output

一行一个整数, 表示初始你操纵的兵力最少是多少才能达到目标.

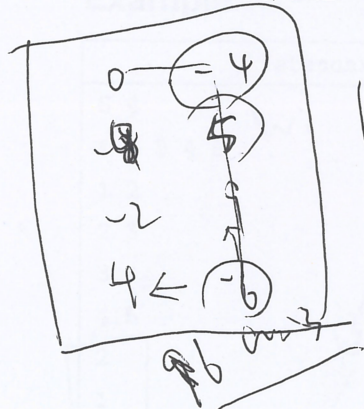
Examples

standard input	standard output
2 2 $\begin{array}{ c c } \hline 1 & 1 \\ \hline -3 & -1 \\ \hline \end{array}$ $\begin{array}{ c } \hline 1 \\ \hline \end{array}$ $\begin{array}{ c } \hline 2 & 1 \\ \hline \end{array}$	4
2 2 1 10 0 -1 0	0
4 2 $\begin{array}{ c c } \hline 1 & -3 \\ \hline -7 & 6 \\ \hline -1 & 10 \\ \hline 5 & -5 \\ \hline \end{array}$ 0	4
4 3 1 -3 1 0 3 -3 0 0 -1 0 3 -5 0	19

Note

对于第一个样例, 最优方案为先向右移动, 再向下移动. 第 0 秒的时候军队在 (1, 1), 兵力为 4; 第 1 秒的时候兵力军队在 (1, 2), 兵力为 4; 第 2 秒中, 在军队移动之前, 敌方大本营的兵力是 3 ($3 = 1 + 2$), 然后军队移动到 (2, 2), 兵力变为 $4 - (3 + 1) = 0$, 成功.

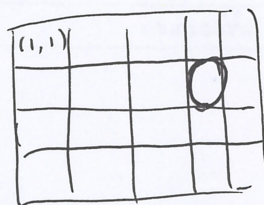
对于第二个样例, 注意一开始的兵力可以为 0.



④

-6-

18 - 4



(1, 1)
12, 4)
1 + 3 = 4 my 矣.

4
-8
5 張

2 張

○

Problem J. Sequence

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

给定一个正整数 N , 你需要构造一个长度为 $2N$ 的序列 $\{a_i\}$, 使得它满足以下条件:

- 对任意 $1 \leq i \leq 2N$ 都有 $1 \leq a_i \leq 3N$;
- 对任意 $1 \leq i, j \leq 2N$ 都有 $a_i \neq a_j$;
- 对任意 $1 \leq i \leq 2N - 1$ 都有 $a_i a_{i+1} + a_i + a_{i+1}$ 不是质数

Input

第一行一个正整数 T ($T \leq 200$), 代表数据组数.

接下来 T 行, 每行一个正整数, 表示 N_i ($N_i \leq 10^5$).

对于所有数据, 满足 $\sum_{i=1}^T N_i \leq 10^5$.

Output

输出共有 T 行, 如果不存在满足条件的序列, 输出 "-1" (不包含引号), 否则, 输出序列 $\{a_i\}$.

如果序列 $\{a_i\}$ 不唯一, 输出一种满足条件的序列即可.

Example

standard input	standard output
3	-1
1	1 4 6 3
2	1 4 9 3 8 2
3	

Problem K. 霍格沃茨加密难题

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

在霍格沃茨魔法学校, 巫师们常开展法术实验和魔法比赛。最近, 学生们发明了一种新魔法, 用于对魔法物品的魔法强度进行加密, 以防止外部干扰。加密方式是利用魔法符号与每个魔法物品的强度进行异或 ($\oplus$) 运算。

在魔法课上, 教授给学生们布置了一项任务: 对给定的魔法物品进行加密。这些魔法物品之间通过魔法通道相连, 每个物品都有一个初始强度 w_i , 且相邻物品的初始强度互不相同。教授要求, 经过异或运算后, 每一对相邻物品的强度都必须不同, 这样才能保证魔法的安全性。

教授一共给出了 q 次指示, 每次指示会给出一个加密密码 x 。学生们需要统计, 在选取部分物品进行加密后, 仍能使整个物品集合满足相邻物品强度不同这一条件的情况数量。具体来说, 就是要判断对于给定的整数 x , 有多少种物品集合在经过异或处理后, 所有相邻物品的强度都不一样。

由于结果可能很大, 请将答案对 $10^9 + 7$ 取模。

Input

第一行输入两个整数 n ($1 \leq n \leq 10^5$) 和 m ($0 \leq m \leq \min(10^5, \frac{n(n+1)}{2})$), 分别表示魔法物品的数量和魔法通道的数量。

第二行输入 n 个整数 $w_1, w_2, \dots, w_n$ ($0 \leq w_i, x < 2^{60}$), 表示每个魔法物品的初始强度。

接下来 m 行, 每行输入两个整数 u ($1 \leq u \leq n$) 和 v ($1 \leq v \leq n$), 表示物品 u 和物品 v 之间有一条魔法通道相连, 保证 $u \neq v$ 。

再输入一个整数 q ($1 \leq q \leq 10^5$), 表示查询的次数。

最后 q 行, 每行输入一个整数 x ($0 \leq x < 2^{60}$), 表示一个查询的加密密码。

Output

对于每个加密密码 x , 输出对应的合法物品集合的数量, 答案对 $10^9 + 7$ 取模, 且每个答案占一行。

Example

standard input	standard output
5 4 1 2 3 4 5 1 2 2 3 3 4 4 5 2 1 3	8 16

Handwritten notes in the example table:

- Next to the first row of input: w_i
- A vertical chain of circles connected by lines, containing the numbers 1, 2, 3, 4, 5 from top to bottom, positioned to the right of the first five rows of input.
- A box around the last three rows of input (2, 1, 3).
- A circle containing the number 3 at the bottom of the chain.

Note

对于样例 1 中的第一个询问, 选取物品集合方案为:

$\{\}, \{1\}, \{2, 3\}, \{4, 5\}, \{1, 2, 3\}, \{1, 4, 5\}, \{2, 3, 4, 5\}, \{1, 2, 3, 4, 5\}$.

Problem L. 圆中圆

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

有一个圆心在 x 轴上的圆 C , 与 x 轴的交点分别是 $(a, 0)$ 和 $(b, 0)$. 请你求出一个圆心在 x 轴上一点 z 的圆 C' 的最小半径 r , 使得 C' 将 C 完全包含.

Input

输入一行包含两个整数 a, b ($-10^3 \leq a < b \leq 10^3$).

第二行输入两个整数 c, d ($-10^3 \leq c \leq 10^3, d = 0$), 表示点 z 的坐标.

Output

输出圆 C' 的最小半径 r .

Examples

standard input	standard output
-5 3 2 0	7
-3 3 0 0	3

Note

样例一示意图如下:

