



西安交通大学
XI'AN JIAOTONG UNIVERSITY

2025 年西安交通大学程序设计校赛

赛题册

- A 纸牌大师
- B 全都登不上
- C 量子力学
- D 对称隔离：黑白之战
- E 公道杯
- F Beat Verdict: Precision Strike
- G Primal Core Optimization: Attribute Balance
- H 9-Nine
- I 奥日
- J 棱堡
- K 我永远喜欢希儿·芙乐艾
- L 罗斯飞鸽
- M 泰拉构史
- N 结束乐队
- O 离散对数

比赛日期：2025 年 5 月 18 日

比赛开始前，严禁翻阅！



题目 A. 纸牌大师

输入文件: standard input
输出文件: standard output
时间限制: 1 second
内存限制: 256 megabytes

CuteWhite 最近迷上了一款多人纸牌游戏。

这个游戏的规则为：每人抽取三张牌，将三张牌的牌面数字加起来作为本轮得分，得分大者获胜。特别地，如果三张牌牌面数字一致，则称为「炸弹」，得分在原基础上额外加 100 分。

但 CuteWhite 计算不太好。因此，他希望你编写一个程序，能快速计算他抽到的三张牌的得分。

输入格式

输入共一行三个整数 a_1, a_2 和 a_3 ($1 \leq a_1, a_2, a_3 \leq 13$)，用一个空格分隔，表示本轮游戏 CuteWhite 抓到的三张牌的牌面数字。

输出格式

输出一行，仅一个整数，表示 CuteWhite 的得分。

样例

standard input	standard output
6 6 6	118
1 1 4	6

提示

对于第一组样例，由于三张牌牌面数字一致，得分应在原基础上额外加 100 分，即总得分为 $6 + 6 + 6 + 100 = 118$ 。

对于第二组样例，三张牌牌面数字并不一致，故总得分为 $1 + 1 + 4 = 6$ 。



题目 B. 全都登不上

输入文件: standard input
输出文件: standard output
时间限制: 1 second
内存限制: 256 megabytes

第一章: 沉淀
第二章: 那场大雨毁了我的 OI 梦
第三章: 程序设计校赛一等奖也是一等
第四章: 什么叫密码错误?
第五章: 那场 1iIl 毁了我的一等梦
第六章: 沉淀, 备战 2026 年程序设计校赛

—— Shirot

J 学长是树王国著名的密码学大师。

树王国一年一度的程序设计校赛又要开始了, S 总理将维护参赛系统的重任交给了他。J 学长为 n 只参赛队伍生成了登录系统的账号和密码。起初, 他使用 `uuid4().hex` 生成十六进制字符串作为密码 (仅包含数字 0-9 和 a-f), 但他认为这种方式生成的密码强度略显不足。于是他改为随机生成字节序列, 并使用 `base64` 编码, 生成更复杂的高强度密码 (包含小写字母, 大写字母, 数字, 下划线 `_`, 减号 `-`)。

然而在热身赛中, 不少选手因难以分辨字符 1, i, I, l, 频繁输错密码, 无法顺利登录系统。S 总理对此非常生气, 要求 J 学长立即整改正式赛的密码, 删除所有容易混淆的字符。J 学长现在想知道正式赛的密码应该怎么改。

具体的, 给定每个队伍正式赛登录系统时的原密码, 你需要删除原密码中的所有 1 (数字 1), i (小写字母 i), I (大写字母 i), l (小写字母 L), 并输出修改后的密码。





输入格式

输入第一行为一个正整数 n ($1 \leq n \leq 5 \times 10^4$), 表示一共有 n 支队伍参加校赛。

接下来 n 行, 每行一个不为空的字符串 s_i (字符串的长度 $|s_i|$ 满足 $1 \leq |s_i| \leq 5 \times 10^4$; 输入字符仅包含小写字母, 大写字母, 数字, 下划线 `_`, 减号 `-`), 表示每支参赛队伍登录系统时的原密码。

保证 n 个队伍原密码的长度之和不超过 5×10^4 。

输出格式

输出一共 n 行, 第 i 行表示第 i 个队伍删除 `liIl` 后的密码, 若密码为空, 则输出空串。

样例

standard input	standard output
5 mCj_m3sYshA 1liil1ili Itboe1Vcig4 lsj5lFx1Nrs -j1ioFlIiI4	mCj_m3sYshA tboeVcg4 sj5FxNrs -joF4
6 Ave_mujica Doloris Mortis Timoris Amoris Oblivionis	Ave_mujca Doors Morts Tmors Amors Obvons



题目 C. 量子力学

输入文件: standard input
输出文件: standard output
时间限制: 1 second
内存限制: 256 megabytes

MCPlayer542 最近在学习《量子计算与量子信息》。他对量子态的表示方式非常感兴趣。

和经典比特不同, 一个量子比特不仅可以处于状态 $|0\rangle$ 或状态 $|1\rangle$, 还可以处于 $|0\rangle$ 和 $|1\rangle$ 之间的叠加态。一个由 n 量子比特组成的量子系统的状态通常由一个 2^n 维的复向量 $|\psi\rangle$ 表示, 其中当你在计算基上进行测量时, 第 i 个复数与其共轭之积代表这 n 个量子比特的状态转为二进制数后, 恰好为 i 的二进制表示的概率。

例如, 对两量子比特的系统 $|\phi\rangle = [\frac{1}{2}, \frac{1}{2}-\frac{i}{2}, 0, \frac{i}{2}]^T$ 表示的状态进行计算基上的测量后, 这个状态以 $\frac{1}{2} \times \frac{1}{2} = \frac{1}{4}$ 的概率处于状态 $|00\rangle$, 以 $(\frac{1}{2} - \frac{i}{2}) \times (\frac{1}{2} + \frac{i}{2}) = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}$ 的概率处于状态 $|01\rangle$, 以 $0 \times 0 = 0$ 的概率处于状态 $|10\rangle$, 以 $\frac{i}{2} \times (-\frac{i}{2}) = \frac{1}{4}$ 的概率处于状态 $|11\rangle$ 。

现在他有一个 n 量子比特的量子态 $|\psi\rangle$, 请你帮他对每个量子比特都求出进行计算基上的量子测量后其为 $|0\rangle$ 和 $|1\rangle$ 的概率分别是多少。

输入格式

第一行输入一个正整数 n ($1 \leq n \leq 8$), 表示系统的量子比特数。

接下来 2^n 行, 第 i 行输入两个由空格分隔的实数 a_i 和 b_i , 分别表示量子系统向量第 i 个分量的实部和虚部。

数据保证量子系统处于所有态的概率求和与 1 的绝对或相对误差不超过 10^{-6} 。

输出格式

输出 n 行, 每行两个实数 p_0 和 p_1 , 用一个空格分隔, 分别表示每个量子比特在测量后处于状态 $|0\rangle$ 和处于状态 $|1\rangle$ 的概率。

你的输出被认为是正确的当且仅当其与正确答案的绝对或相对误差不超过 10^{-6} 。

样例

standard input	standard output
2	0.250000000 0.750000000
0.5 0	0.750000000 0.250000000
0.5 -0.5	
0 0	
0 0.5	



题目 D. 对称隔离：黑白之战

输入文件: standard input
输出文件: standard output
时间限制: 1 second
内存限制: 512 megabytes

你在玩一款叫做「你的克拉夫特」的游戏。在这个游戏里，你可以在一个充满着方块的三维空间中自由地创造和破坏不同种类的方块。

现在你脚下有一块由 $N \times M$ 个格子组成的网格。一开始，你在这些格子上铺满了白色地毯，然后你就去睡觉了。

但是不幸的是，在你睡觉的时候，出题人将一些白色地毯偷偷换成了黑色地毯。

当你醒来再次打开游戏，你发现你的杰作被污染了，所以你非常气愤！所以你决定将错就错，希望通过保持现有黑色地毯不变，将部分白色地毯染黑，来满足下面两个条件：

- 任意两个黑色地毯所在的格子不能共享公共边，即满足：
 - 若第 i 行第 j 列格子中为黑色地毯，则第 $i-1$ 行第 j 列、第 $i+1$ 行第 j 列、第 i 行第 $j-1$ 列、第 i 行第 $j+1$ 列格子中如果存在地毯，则不能为黑色地毯。
- 最终形成的图案至少存在一条水平或者竖直的对称轴，即下面两个命题中至少满足一个：
 - 对于任意的 $i \in [1, N]$ 和 $j \in [1, M]$ ，第 i 行第 j 列格子中的地毯颜色与第 $N-i+1$ 行第 j 列格子中的地毯颜色相同；
 - 对于任意的 $i \in [1, N]$ 和 $j \in [1, M]$ ，第 i 行第 j 列格子中的地毯颜色与第 i 行第 $M-j+1$ 列格子中的地毯颜色相同。

你需要判断你是否可以达成这两个条件。

注意：你不能破坏已有的黑色地毯。

输入格式

第一行，两个整数 N 和 M ($1 \leq N, M \leq 100$)，用一个空格分隔，表示网格的大小。

接下来 N 行，每行一个长度为 M 的字符串 S 。 S 中某一位为 W 表示这个位置是一块白色地毯，为 B 表示这个位置是一块黑色地毯。字符串中不会出现除此以外的字符。

输出格式

如果你可以达成题目中给出的条件，输出 **Yes**；否则，输出 **No**。

答案对大小写不敏感。例如 **yEs**, **Yes**, **yes** 和 **YES** 都可以表示你可以达成条件。

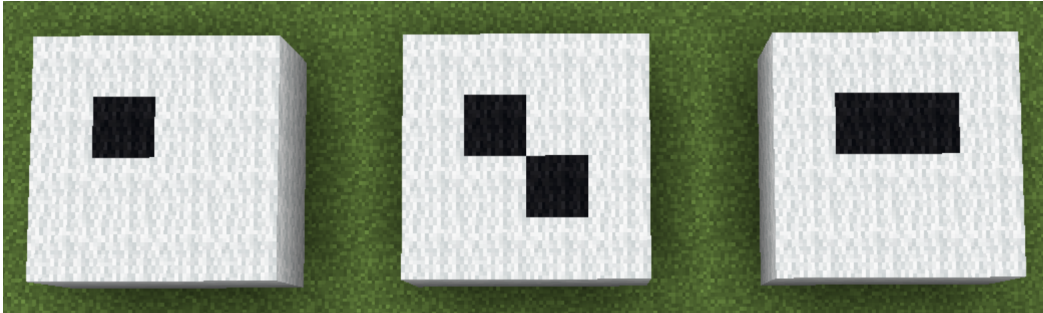


样例

standard input	standard output
4 4 WWW WBWW WWW WWW	No
4 4 BWBB WWW WWW WWBB	Yes
4 4 BWWW WBWW WWBW WWW	No
5 5 BWBWB WBWBW BWBWB WBWBW BWBWB	Yes
2 2 BB WW	No
2 2 WW WW	Yes
8 6 WWBWWW BWWWWW WBWWW WWWBWB BWWWWW WWBWWW WBWWW BWWWWW	No

提示

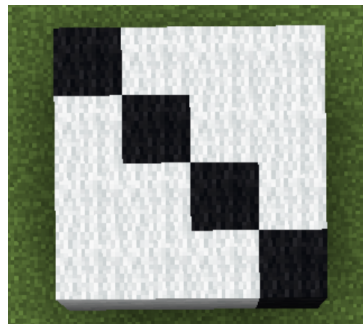
对于第一组样例：



最左边的图是你醒来时看到的情况，你无法通过把白色地毯换成黑色地毯以达成条件。中间和最右边的图，显示了两种可能的更换地毯的方式，但并不能满足题目中的两个要求（中间的图不存在水平或者竖直的对称轴；最右边的图黑色地毯共享公共边）。可以证明，任何更换地毯的方式均不符合要求，所以输出 No。



(a) 样例二图



(b) 样例三图



(c) 样例四图

对于第二组样例：如图显示了一种合法的更换地毯的方式。你可以把左下角的白色地毯换成黑色地毯，这样符合题目中的两个要求。因此，你应该输出 Yes。

对于第三组样例：如图显示了一种可能的更换地毯的方式，但是这个方式并不是合法的。请记住：只有存在水平或者竖直的对称轴才被视作合法的方案。

对于第四组样例：如图，你醒来的时候看到的情况本身已经是合法的了，所以你不需要进行操作。

对于第五组样例：你醒来的时候已经有两个共享公共边的黑色地毯了，所以你无法更换白色地毯以符合要求。

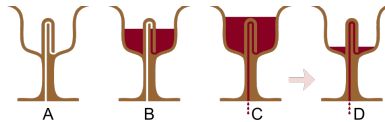
对于第六组样例：全是白色地毯的图案也被视作是合法的，因为它有水平或者竖直的对称轴，同时不存在两个共享公共边的黑色地毯。



题目 E. 公道杯

输入文件: standard input
输出文件: standard output
时间限制: 1 second
内存限制: 256 megabytes

Alice 和 Bob 在玩一个游戏, 在他们面前摆着 n 个公道杯 (在欧洲称为毕达哥拉斯杯)。用公道杯饮酒之时, 只要不斟满, 则与常杯无异。若斟满, 则酒会从底孔完全流光。公道杯的结构如下图。



在初始阶段, 向 n 个空公道杯依次倒入酒, 可能倒入如下三种分量的酒: 不倒酒, 用大写英文字母 E 表示; 半杯酒, 用大写英文字母 H 表示; 一整杯酒, 用大写英文字母 F 表示。

然后, 每一次 Alice 和 Bob 可以选择一个非空的公道杯, 将它之中的所有酒倒入它右边的公道杯中。若选择最右边的公道杯, 则可以直接将其倒掉。

在公道杯的任意时刻, 当酒满了的时候, 会立刻流光变成空杯。若一个公道杯是半满的状态, 向其中再倒半杯酒, 则会因为满了而流空。

Alice 和 Bob 依次进行操作, Alice 先进行第一次操作。当 Alice 或 Bob 在场上没有可以选取的公道杯时, 则无法选取公道杯的人输掉游戏。假设 Alice 和 Bob 都很聪明, 采取最优做法, 问 Alice 和 Bob 谁获胜。

输入格式

输入共一行一个字符串 S ($1 \leq |S| \leq 10^6$)。字符串的长度 $|S|$ 就是公道杯的个数 n 。

S_i 为大写英文字母 EHF 中的一个, 表示第 i 个公道杯在最开始要倒多少酒。不倒酒, 用大写英文字母 E 表示; 半杯酒, 用大写英文字母 H 表示; 一整杯酒, 用大写英文字母 F 表示。

输出格式

输出共一行一个字符串。若 Alice 获胜输出 Alice; 若 Bob 获胜输出 Bob。

样例

standard input	standard output
HEFH	Alice
HEHEFHFE	Bob

提示

对于第一组样例: 最开始时, 对第 1 个公道杯倒入半杯酒; 第 2 个公道杯不倒酒; 第 3 个公道杯倒入一整杯酒, 随即立刻流空; 第 4 个公道杯倒入半杯酒。即开始后, 4 个公道杯的状态为半满、空、空、半满。

Alice 开始时可以选择第 4 个公道杯, 直接将其倒掉; 随后, 只有第 1 个公道杯是半满的, Bob 只能选择第 1 个公道杯, 倒入第 2 个公道杯中; 同样的, 此时只有第 2 个公道杯是半满的, Alice 只能选择第 2 个公道杯, 倒入第 3 个公道杯中; Bob 只能选择第 3 个公道杯, 倒入第 4 个公道杯中; Alice 只能选择第 4 个公道杯, 直接将其倒掉。此时, Bob 无法再选择公道杯, 输掉了游戏。



题目 F. Beat Verdict: Precision Strike

输入文件: standard input
输出文件: standard output
时间限制: 1 second
内存限制: 256 megabytes

这是一道交互题。

你正在玩一款名为「乌蒙滴叉」的音乐游戏。在专家模式中，需要精确校准音符的击打时机参数，该参数是一个 $[1, n]$ 内的神秘正整数 x 。

为了确定这个参数，你可以进行至多 4 次校准测试。每次测试时，你选择一个正整数测试值 y ($1 \leq y \leq n$)，系统将告知你 $y > x$ 是否成立。最终你需要给出一个正整数参数估计值 x' ($1 \leq x' \leq n$)，且满足 $x' \in [0.5x, 2x]$ 。

注意：最终给出的估计值 x' 不计入上述至多 4 次测试次数。

输入格式

输入包含多个测试用例。数据的第一行包含一个整数 t ($1 \leq t \leq 500$) 表示测试用例数。每个测试用例的交互流程在下文中描述。

Interaction Protocol

在每个测试用例中，输入的第一行包含一个正整数 n ($1 \leq n \leq 5 \times 10^9$)，表示参数的可能范围。

若你想进行测试，输出 `? y` ($1 \leq y \leq n$)。然后你读入对该测试的响应，为一个整数 a ($a \in \{0, 1\}$)，1 表示 $y > x$ 为真，0 表示 $y > x$ 为假。

若你想提交参数 x' ，输出 `! x'` ($1 \leq x' \leq n$)。然后你立即结束本个测试用例的交互，并准备下一个测试用例的交互。这次交互不计入上述至多 4 次测试次数。

注意在你的程序每轮输出结束时（即，每一次输出 `? y` 或 `! x'` 后），需要输出回车并刷新输出缓冲区，否则你将会得到 `Time Limit`。

你可以使用：

- C 的 `fflush(stdout)`;
- C++ 的 `cout.flush()`;
- Java 的 `System.out.flush()`;
- Python 的 `stdout.flush()`;

来刷新输出缓冲区。

请注意：交互库自适应，即每个测试用例中的正整数 x ($1 \leq x \leq n$) 可能会随着你的输入而变化，但始终满足所有已经发生过的询问。

如果你最后输出的答案正确，你会得到 `Correct`；

如果你给出的询问不符合题目范围要求，或最后输出的答案不正确，你会得到 `Wrong Answer`；



此外，其他的评测结果仍会在评测过程中根据通常情况返回。

样例

standard input	standard output
2	
1	! 1
8	
	? 6
1	
	? 4
1	
	? 2
0	
	? 3
0	
	! 3

提示

在第一组测试用例中，可以唯一确定 $x = 1$ ，因此我们直接提交 $x' = 1$ 。

在第二组测试用例中，隐藏的参数 $x = 3$ ，交互过程如下：

1. 测试 $y = 6$ ，响应为 $y > x$ 为真；
2. 测试 $y = 4$ ，响应为 $y > x$ 为真；
3. 测试 $y = 2$ ，响应为 $y > x$ 为假；
4. 测试 $y = 3$ ，响应为 $y > x$ 为假；
5. 提交 $x' = 3$ 。

请注意，此示例仅用于演示交互格式。不能保证提供的查询是最优的或唯一确定答案。



题目 G. Primal Core Optimization: Attribute Balance

输入文件: standard input
输出文件: standard output
时间限制: 9 seconds
内存限制: 512 megabytes

你正在玩一款被称为「韵律原罪」的游戏。在这个游戏里，你需要操纵你的搭档们，在「世界」中探索。在游戏中，每一个搭档都有三种属性：Stop (S)，Flag (F) 和 Ever (E)。当所有搭档的属性完全一致时，她们就能成功融合。

现在你想要把所有搭档全部融合，换句话说就是让每一对搭档的属性全部相等。为了达成这一目的，你不得不对她们的属性进行修改。一次修改分为以下步骤：

1. 你任选一位搭档作为修改的目标（你只能选一位搭档）；
2. 你任选她的若干个属性；
3. 将她的所选属性的数值全部增加 1 或全部减少 1（必须同时执行同一种操作，即全部加或全部减）。

由于你的魔力不强，所以每次修改都需要花费 1 源石。现在你想知道，你至少需要花费多少源石，才能达到你的目标。

输入格式

第一行，一个整数 N ($1 \leq N \leq 10^5$) 表示你拥有的搭档个数。

接下来 N 行，每行三个整数 S_i, F_i 和 E_i ($1 \leq S_i, F_i, E_i \leq 10^5$)，用一个空格分隔，表示第 i 个搭档的三个属性的值。

输出格式

仅一个整数表示答案。



样例

standard input	standard output
3 2 2 2 2 2 2 1 2 3	2
3 2 3 1 3 5 3 3 2 1	4
1 201 502 10	0
4 1 2 3 1 2 3 4 5 6 4 5 6	6

提示

对于第一组样例, 存在以下操作方式, 使得总修改次数是 2 次:

1. 选择搭档 3 和她的第一个属性, 将她的属性修改为 2, 2, 3;
2. 选择搭档 3 和她的第三个属性, 将她的属性修改为 2, 2, 2。

同时可以证明, 不能通过少于 2 次操作达成目标。所以, 你应该输出 2。

对于第二组样例, 存在以下操作方式, 使得总修改次数是 4 次:

1. 选择搭档 1 和她的第一、第三属性, 将她的属性修改为 3, 3, 2;
2. 选择搭档 2 和她的第二、第三属性, 将她的属性修改为 3, 4, 2;
3. 选择搭档 2 和她的第二个属性, 将她的属性修改为 3, 3, 2;
4. 选择搭档 3 和她的第二、第三属性, 将她的属性修改为 3, 3, 2。

同时可以证明, 不能通过少于 4 次操作达成目标。所以, 你应该输出 4。

对于第三组样例, 因为只有一位搭档, 所以不需要进行操作。所以, 你应该输出 0。



题目 H. 9-Nine

输入文件: standard input
输出文件: standard output
时间限制: 1 second
内存限制: 256 megabytes

『九次九日九重色，
天色天歌天籁音，
春色春恋春熙风，
雪色雪花雪余痕』

— 《9-nine-》

工艺品 (Artifact)，白已津川市自行制作的动画《轮回转生的莫比乌斯环》(虽然收视率极低) 中的道具居然来到了现实！拥有它的人都会获得独有的特殊能力，这一切的一切都要从那场地震开始。

「千年之前，本世界和异世界是互通的，而工艺品正是异世界的魔法道具。由于种种原因，出现了一个叫「生命之树」的组织，将所有的工艺品回收后设下了法阵，阻止了两个世界的流通。但是由于地震，法阵被破坏，于是工艺品便流向了这个世界。」——来自异世界的玩偶索菲提娅如是说道。

在索菲的帮助下，我们的主角新海翔带领四位女主与反派进行激烈的斗争，最终成功阻止其使用工艺品犯下不可饶恕的错误。不过，为了根除这个问题，新海翔和索菲决定断绝两个世界的联系，重新打开封印。

封印的法阵可以看做两个 9×9 的黑白方阵，新海翔和索菲分别可以控制一个方阵，他们可以做如下动作：

1. 新海翔将自己面前的方阵向左或向右旋转 90 度；
2. 索菲将自己面前的方阵向左或向右旋转 90 度（两个方阵不联动）；
3. 新海翔和索菲选择方阵的同一列，进行交换；

当新海翔一侧的方阵全是白色，同时索菲一侧的方阵全是黑色，那么封印就完成了，法阵将会开启，断开两个世界的联系，大家可以回归正常生活。

时间有限，请你——我们的「第九人 (Nine)」，帮助他们在 999 步内完成这个封印。

话虽如此，但是出题人被警告题出太难了，所以决定缩小数据范围！

具体的，将方阵的边长减小至 3，同时步数减小至 $9 \times 9 = 81$ ！

形式化地说，有两个 3×3 的 01 矩阵 A 和 B ，你可以做如下操作：

- 操作 1：选择矩阵 A 或 B ，将其向左或向右旋转 90 度；
- 操作 2：选择一个整数 k ($1 \leq k \leq 3$)，将两矩阵的第 k 列交换。

请在 81 次操作内将 A 变为全 0 矩阵。数据保证此时 B 一定为全 1 矩阵。



输入格式

第 1 ~ 3 行, 每行是一个长度为 3 的 01 字符串, 表示矩阵 A 的初始状态;

第 4 ~ 6 行, 每行是一个长度为 3 的 01 字符串, 表示矩阵 B 的初始状态。

数据保证在有限次操作内, 可以将 A 变为全 0 矩阵, B 变为全 1 矩阵。

输出格式

第一行一个整数 N ($0 \leq N \leq 81$), 表示你一共需要进行 N 次操作。

接下来的 N 行表示你依次进行的操作。请按如下格式输出:

- 如果你选择使用操作 1, 请输出矩阵名与旋转方向。具体地:
 - 若你想将 A 矩阵向左旋转 90 度, 输出 AL;
 - 若你想将 A 矩阵向右旋转 90 度, 输出 AR;
 - 若你想将 B 矩阵向左旋转 90 度, 输出 BL;
 - 若你想将 B 矩阵向右旋转 90 度, 输出 BR。
- 如果你选择使用操作 2, 请输出 C 与列号。例如, 你要交换两个矩阵的第 2 列, 请输出 C2。

其中, 将一个矩阵向左旋转, 指逆时针旋转; 将一个矩阵向右旋转, 指顺时针旋转。

请注意: 不要输出多余的行末空格与文末回车。

样例

standard input	standard output
100	9
000	BL
000	BL
111	BL
111	AL
011	C1
	AL
	C1
	AR
	C1

提示

可以证明, 在数据合法的情况下, 原题在 999 步内一定存在解, 弱化版的题目在 81 步内一定存在解。

最重要的是还在分割商法的《9-nine》今年即将推出新作, 预计冬季上架某平台



题目 I. 奥日

输入文件: standard input
输出文件: standard output
时间限制: 1 second
内存限制: 256 megabytes

精灵古树是尼博尔山的生命之源, 其根系中流淌的流光维系着整片森林的生态平衡。古树的躯体由 n 枚光之核心构成, 这些璀璨如星的光点通过 $n-1$ 条能量枝干交织成无环的树状脉络。任意两枚核心之间, 都有一条唯一的能量枝干通道蜿蜒相连, 仿佛命运编织的丝线。

而奥日, 本是古树孕育的守护灵体, 却在雷霆撕裂苍穹的雨夜, 被狂暴的飓风卷入深渊。失去奥日的古树试图通过光之仪式呼唤他归来, 但失控的能量反噬让它陷入永夜, 原本澄澈的核心如今爬满黑暗的纹路。

如今, 归来的奥日必须激活所有被侵蚀的核心: 当他首次触碰某个核心时, 纯净能量会驱散黑暗; 但重复经过时, 紊乱的能量将累计形成过载波动。古树的法则严苛限定——整条路径中, 重复触发的波动总和不得超过 k 次。

此刻, 奥日悬浮在星网交织的虚空中。他可以从任意核心启程, 沿着能量枝干的轨迹穿梭。奥日需要在蜿蜒的能量枝干间规划路径, 在限制内点亮最多的核心。

唯有让尽可能多的光之核心重新共鸣, 才能唤醒古树真正的力量, 让治愈的流光再次奔涌在尼博尔山的每一片叶脉中! 尼博尔山将迎来破晓, 而黑暗, 终会在这片星网的共振中灰飞烟灭……

形式化的, 给定一个非负整数 k , 给出一颗无根树 $T = (V, E)$, $V = \{1, 2, \dots, n\}$ 。

定义一条由 m 个可重点构成的路径 $l = (u_1, u_2, u_3, \dots, u_m)$ 满足: 对于任意的 $1 \leq i \leq m-1$ 有 $(u_i, u_{i+1}) \in E$ 。如果存在 $1 \leq i < j \leq m$, 使得 $u_i = u_j$, 则仍然认为 l 是一条路径。

定义路径 l 上的本质不同点集 $V' = \{v \mid v = u_i, \exists i \in [1, m]\}$ 。记 $|V'|$ 为集合 V' 的大小。记路径 l 上的重复点数量为 s , 有 $s = m - |V'|$ 。

你需要找到一条路径 l , 在满足 $s \leq k$ 的情况下, 最大化 $|V'|$ 。

你需要输出这条路径。如果存在多组路径满足条件, 输出任意一条满足题意的路径, 就会被认为正确。

输入格式

输入包含多个测试用例。数据的第一行包含一个整数 t ($1 \leq t \leq 2 \times 10^3$) 表示测试用例数。接下来描述每个测试用例。

每个测试用例的第一行包含两个整数 n ($2 \leq n \leq 2 \times 10^5$) 和 k ($0 \leq k \leq n$), 用一个空格分隔, 分别表示树上点的个数和可重复点数量的最大值。注意, k 的取值可能为 0。

接下来 $n-1$ 行, 每行两个正整数 u 和 v ($1 \leq u, v \leq n$), 用一个空格分隔, 表示树上存在一条边 (u, v) 。保证该 $n-1$ 条边构成树。

保证所有测试用例的 n 之和不超过 2×10^5 。



输出格式

对于每个测试用例，输出两行。

第一行一个正整数 m ，表示路径 l 的可重点数量。路径 l 需要在满足重复点数量 $s \leq k$ 的情况下，最大化本质不同点集大小 $|V'|$ 。

第二行共 m 个正整数 $u_1, u_2, u_3, \dots, u_m$ ，每个正整数之间用一个空格分隔，表示路径 l 为 $(u_1, u_2, u_3, \dots, u_m)$ 。

如果存在多组路径满足条件，输出任意一条满足题意的路径，就会被认为正确。

可以被证明的，对于一个合法的路径， m 一定有 $m \leq n + k$ 。输出的 m 一定要满足 $m \leq n + k$ ，否则将直接返回 Wrong Answer。

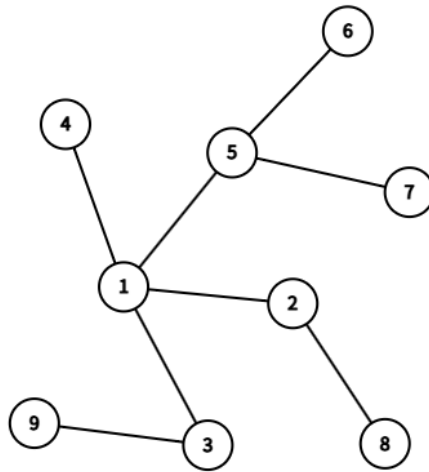
样例

standard input	standard output
2	11
9 3	9 3 1 5 7 5 6 5 1 2 8
1 3	5
2 1	3 2 1 2 4
7 5	
9 3	
2 8	
1 5	
4 1	
6 5	
4 4	
1 2	
2 3	
4 2	



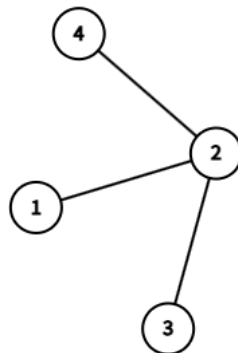
提示

对于第一组测试用例, $n = 9$ 且 $k = 3$, 组成的树如下:



一组可行的解为: $l = (9, 3, 1, 5, 7, 5, 6, 5, 1, 2, 8)$ 。可以被证明的, 此时本质不同点集大小 $|V'|$ 达到最大值 8, 且重复点数量 $s = 3 \leq k$ 。

对于第二组测试用例, $n = 4$ 且 $k = 4$, 组成的树如下:



一组可行的解为: $l = (3, 2, 1, 2, 4)$ 。可以被证明的, 此时本质不同点集大小 $|V'|$ 达到最大值 4, 且重复点数量 $s = 1 \leq k$ 。

由于本题输入输出数据规模较大, 建议使用较为快速的输入输出方式。



题目 J. 棱堡

输入文件: standard input
输出文件: standard output
时间限制: 1 second
内存限制: 256 megabytes

棱堡 (Bastion) 是第一种仅靠直射火力而不存在攻击死角的堡垒。

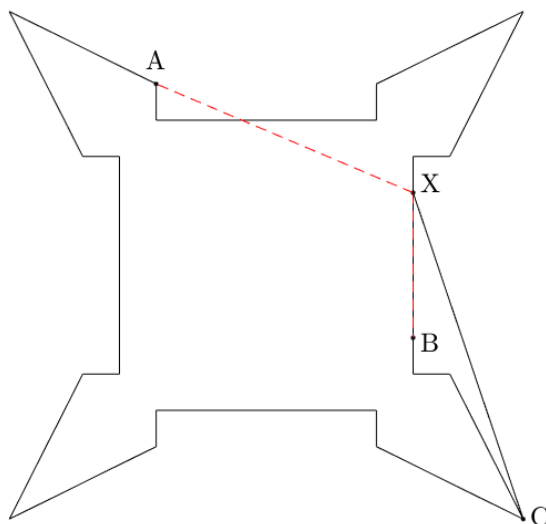


简单非退化多边形是由 n ($n \geq 3$) 个顶点序列组成的闭合多边形区域，满足以下条件：

- 顶点序列首尾相连，形成 n 条边，构成平面紧致闭集（含所有边界和内部点）；
- n 条边仅在顶点处相接且互不相交（相邻边在公共顶点外无其他交点）；
- n 个顶点互不重合，且不位于任何非相邻边的内部，且任意三个连续的顶点不共线。



棱堡可以被视为由 n 个点和 n 条边组成的简单非退化多边形。对于多边形边上互异两点 P 和 Q ，我们定义点 P 可以火力直射点 Q ，当且仅当线段 PQ 与多边形只交于 P 和 Q 两点。如下图所示，点 A 和 B 不能火力直射点 X ，但是点 C 可以。如果对于多边形边上一点 P ，不存在多边形边上另外一点 Q 使得点 Q 可以火力直射 P ，则我们称点 P 为该多边形的火力盲区。



我们称一个简单非退化多边形是棱堡，当且仅当其至多存在有限个点是火力盲区。给定一个简单非退化多边形，请判断其是否是一个棱堡。

形式化而言，给定一个简单非退化多边形，请判断其是否只有有限个位于多边形边上的点 P 满足，不存在多边形边上的另外一点 Q ，使得 PQ 线段与多边形的交集只包含 P 和 Q 。

输入格式

输入包含多组测试用例。数据的第一行包含一个整数 t ($1 \leq t \leq 10^4$) 表示测试用例数。接下来描述每个测试用例。

每个测试用例的第一行包含一个整数 n ($3 \leq n \leq 10^5$)，为多边形的边数。

接下来 n 行，每行包含两个整数 x_i 和 y_i ($-10^9 \leq x_i, y_i \leq 10^9$)，描述多边形的一个顶点。

输入数据保证构成简单非退化多边形。顶点以逆时针顺序给出，即按顺序走过所有顶点后恰好逆时针旋转 360° 。按顶点的输入顺序依次连边（第 i 个顶点连向第 $i+1$ 个顶点，第 n 个顶点连回第 1 个顶点）。

保证所有测试用例的 n 之和不超过 2×10^5 。

输出格式

对于每个测试用例，输出仅包含一行一个字符串，如果多边形是棱堡，输出 YES，否则输出 NO。答案对大小写不敏感。



样例

standard input	standard output
2	YES
20	NO
7 5	
9 5	
13 13	
5 9	
5 7	
-5 7	
-5 9	
-13 13	
-9 5	
-7 5	
-7 -5	
-9 -5	
-13 -13	
-5 -9	
-5 -7	
5 -7	
5 -9	
13 -13	
9 -5	
7 -5	
4	
1 1	
-1 1	
-1 -1	
1 -1	



题目 K. 我永远喜欢希儿·芙乐艾

输入文件: standard input
输出文件: standard output
时间限制: 3 seconds
内存限制: 32 megabytes

请注意本题不同寻常的空间限制。

给定操作序列 A 和一棵有根树 T , A 中的每个操作都形如「给 T 中 u 到 v 的简单路径上的所有点权增加 x 」。其中, u 到 v 的简单路径定义为: 从 u 开始到 v 结束的一条不重复访问任何节点和边的路径。

初始时 T 的根为 1, 所有点权皆为 0。

你需要执行三种操作:

1. 给定区间 $[l, r]$, 依次执行 $A_l, A_{l+1} \dots$ 到 A_r 的操作;
2. 查询 T 以 x 为根的子树和;
3. 换 T 的根到 y (若当前根为 y , 则不进行操作)。

输入格式

第一行包含三个正整数 n_1, n_2 和 m ($1 \leq n_1, n_2, m \leq 1 \times 10^5$), 用一个空格分隔, 分别表示 A 的长度、 T 的总点数、询问与操作的次数。

接下来 n_1 行, 第 i 行包含三个整数 u_i, v_i 和 x_i ($1 \leq u_i, v_i \leq n_2, 1 \leq x_i \leq 1 \times 10^2$), 用一个空格分隔, 表示操作 A_i 为「给 T 中 u_i 到 v_i 的简单路径上的所有点权增加 x_i 」。

接下来 $n_2 - 1$ 行, 每行包含两个正整数 u 和 v ($1 \leq u, v \leq n_2$), 用一个空格分隔, 表示树的一条边。保证该 $n_2 - 1$ 条边构成树。

接下来 m 行, 每行的第一个数 op ($op \in \{1, 2, 3\}$) 表示操作编号, 随后:

- 若 op 为 1, 则后面跟随两个正整数 l 和 r ($1 \leq l \leq r \leq n_1$), 用一个空格分隔, 表示依次执行 $A_l, A_{l+1} \dots A_r$ 的操作;
- 若 op 为 2, 则后面跟随一个正整数 x ($1 \leq x \leq n_2$), 表示查询 T 以 x 为根的子树和;
- 若 op 为 3, 则后面跟随一个正整数 y ($1 \leq y \leq n_2$), 表示换 T 的根到 y (若当前根为 y , 则不进行操作)。

输出格式

对于每个操作 2, 输出一行一个整数, 表示 T 以 x 为根的子树和。

数据保证答案在有符号超长整型所表达的范围内。



样例

standard input	standard output
4 5 7	29
3 5 2	25
5 4 3	63
3 1 2	
2 5 1	
2 1	
4 2	
1 5	
3 2	
1 1 4	
2 1	
1 2 3	
2 2	
3 3	
1 1 3	
2 2	



题目 L. 罗斯飞鸽

输入文件: standard input
输出文件: standard output
时间限制: 1 second
内存限制: 256 megabytes

awa 正在参加一个叫做罗斯飞鸽的真人音游活动。在活动中, awa 要在指定的时间出现在指定的位置。

具体来说, 罗斯飞鸽的活动场地是一条无限长的直线。罗斯飞鸽一共有 n 个判定点。第 i 个判定点要求 awa 在时刻 t_i 出现在位置 x_i 。

awa 的最大跑步速度是 v 个单位每时刻。在时刻 0, awa 可以任意选择一个位置, 随后开始游玩罗斯飞鸽。

awa 想知道, 自己最多能够达成多少个判定点。

输入格式

第一行一个正整数 T ($1 \leq T \leq 5 \times 10^5$), 表示 awa 一共进行了 T 次游戏。

对于每次游戏, 第一行两个正整数 n 和 v ($1 \leq n \leq 5 \times 10^5, 1 \leq v \leq 10^9$), 用一个空格分隔, 表示判定点的数量和 awa 的最大速度。

接下来 n 行, 每行两个整数 t_i 和 x_i ($0 \leq t_i, |x_i| \leq 10^9$), 用一个空格分隔, 描述一个判定点。保证一局游戏中没有两个完全相同的判定点。

保证 T 轮游戏中 n 的总和不超过 5×10^5 。

输出格式

一共 T 行, 每行一个整数, 表示对应游戏中 awa 最多能达成多少个判定点。



样例

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

提示

由于本题输入输出数据规模较大，建议使用较为快速的输入输出方式。例如 `scanf` 和 `printf`。



题目 M. 泰拉构史

输入文件: standard input
输出文件: standard output
时间限制: 2 seconds
内存限制: 1024 megabytes

「我相信我们之间的联系会超越时间和空间。」

「就算是海洋沸腾、大气消失，就算我们的卫星接连坠入重力的漩涡，就算我们的太阳凶恶地膨胀，无情地吃掉它的孩子直至万籁俱寂……我们也一样能再见面。」

「在那用黑暗和星点光芒装饰的文明尽头，我们也一样会再见面。一定。」

「我会等到那一天。我肯定会等到那一天。等我。你也要等我。」

「这里万籁俱寂……太安静了。别留下我。」

你是罗德岛的博士，是巴别塔的恶灵，但你更是普瑞赛斯的预言家。文明危在旦夕，但万幸的是，你在一切都来不及之前还是想起了她。去吧，将她自黑暗中找回，不要背叛了在时间中等待的她。



这是解开谜团的第一步，一道计数题。

有一个长度为 n 的序列 $a_1, a_2, \dots, a_n$ ，**序列中的元素互不相同**，一次操作定义如下：

- 选择一个下标 i ($1 \leq i \leq n-1$) 满足 $|a_i - a_{i+1}| = 1$ ，在序列中交换 a_i 与 a_{i+1} 。

你可以进行**任意次**操作。你能得到多少种不同的序列？由于答案较大，请输出答案对 998244353 取模的结果。

两个序列 $a_1, a_2, \dots, a_n$ 和 $b_1, b_2, \dots, b_n$ 不同当且仅当存在 $i \in [1, n]$ 使得 $a_i \neq b_i$ 。



输入格式

输入共两行。

第一行一个正整数 n ($1 \leq n \leq 10^6$)，表示序列的长度。

第二行 n 个正整数 a_i ($1 \leq a_i \leq 10^9$)，用一个空格分隔，表示初始的序列为 $a_1, a_2, \dots, a_n$ 。数据保证序列中的 a_i 互不相同。

输出格式

输出仅一个正整数，表示可能的序列的个数对 998244353 取模的结果。

样例

standard input	standard output
4 1 4 2 3	3
9 11 4 5 14 19 1 9 8 10	6
1 1	1
12 4 3 7 6 8 11 9 10 12 14 13 15	90

提示

对于第一组样例，初始时的序列为 1, 4, 2, 3。注意，初始时的序列也是一种可能的序列，需要计数。

对序列 1, 4, 2, 3，因为 $|a_3 - a_4| = 1$ ，此时可以交换 a_3, a_4 ，交换后序列为 1, 4, 3, 2。

对序列 1, 4, 3, 2，因为 $|a_2 - a_3| = 1$ ，此时可以交换 a_2, a_3 ，交换后序列为 1, 3, 4, 2。

可以证明，对于该序列，经过任意次操作不同的序列只有上述 3 种可能。

由于本题输入输出数据规模较大，建议使用较为快速的输入输出方式。



题目 N. 结束乐队

输入文件: standard input
输出文件: standard output
时间限制: 1 second
内存限制: 256 megabytes

「我是来结束乐队的」

某个少女乐队正在进行合照, 可摄影师不怀好意, 她打算对拍出的照片进行处理后再发布社交媒体, 引起粉丝们的不满, 这会导致乐队中某些成员人气的升降。她觉得这样就能结束这支乐队了。她的计划如下:

已知少女乐队有 n 个成员, 从左到右排成一排照相, 其中每个成员的初始人气排名并不相同, 分别从排名第 1 到第 n 。摄影师打算拍 k 张照片并在每次拍完之后公开到社交媒体, 她的每张照片只包含乐队成员排列中连续的一段, 这样这段照片中未出现成员中的人气排名最靠前的成员的粉丝就会红温脱粉, 导致这名成员的排名下降一名 (原低一名的成员排名会上升)。通过 k 次操作, 她成功打乱了乐队成员的排名, 她的目的就达到了。

但现在她遇到了一个问题, 因为乐队的成员太多了, 对于每次截取下来的某一段照片, 她不能判断哪个未出现的成员排名最靠前, 所以她求助于你, 希望你帮她解决问题。

形式化的, 给定一个从 1 到 n 的排列 $a_1, a_2, \dots, a_n$, 进行 k 次操作, 每次操作:

- 给定一个区间 $[l, r]$, 求当前排列中区间 $[l, r]$ 上未出现的最小正整数是多少, 并设此数为 x (即 $x = \min\{p \mid \forall i \in [l, r], p \neq a_i, p \geq 1\}$);
- 若 $x < n$, 则输出 x , 并交换当前排列 $a_1, a_2, \dots, a_n$ 中数字 x 与 $x+1$ 的位置;
- 若 $x \geq n$, 则输出字符串 `peace`。

其中, 从 1 到 n 的排列 $a_1, a_2, \dots, a_n$, 是指满足 $\{a_1, a_2, \dots, a_n\} = \{1, 2, \dots, n\}$ 的一组数。

输入格式

第一行包含一个正整数 n ($1 \leq n \leq 5 \times 10^5$), 表示排列的大小。

第二行包含 n 个正整数 a_i , 用一个空格分隔, 表示初始排列 $a_1, a_2, \dots, a_n$ ($\{a_1, a_2, \dots, a_n\} = \{1, 2, \dots, n\}$)。

第三行一个正整数 k ($1 \leq k \leq 10^5$), 表示操作的次数。

接下来 k 行, 每行两个正整数 l 和 r ($1 \leq l \leq r \leq n$), 用一个空格分隔, 表示操作的区间。

输出格式

对于每个测试数据中的每次操作, 给出答案, 并执行修改。

记排列 $a_1, a_2, \dots, a_n$ 在区间 $[l, r]$ 上未出现的最小正整数为 x :

- 若 $x < n$, 则输出 x , 并交换当前排列 $a_1, a_2, \dots, a_n$ 中数字 x 与 $x+1$ 的位置;
- 若 $x \geq n$, 则输出字符串 `peace`。



样例

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

提示

对于第一组样例：

第一次操作 [2, 4]，当前排列为 4, 3, 1, 2, 5。

区间 [2, 4] 内未出现的最小正整数为 4，输出 4 并交换排列中 4 和 5 的位置。

第二次操作 [2, 5]，当前排列为 5, 3, 1, 2, 4。

区间 [2, 5] 内未出现的最小正整数为 5，输出 peace，排列不变。

第三次操作 [1, 3]，当前排列为 5, 3, 1, 2, 4。

区间 [1, 3] 内未出现的最小正整数为 2，输出 2 并交换排列中 2 和 3 的位置。

第四次操作 [1, 3]，当前排列为 5, 2, 1, 3, 4。

区间 [1, 3] 内未出现的最小正整数为 3，输出 3 并交换排列中 3 和 4 的位置。

第五次操作 [1, 5]，当前排列为 5, 2, 1, 4, 3。

区间 [1, 5] 内未出现的最小正整数为 6，输出 peace，排列不变。



题目 O. 离散对数

输入文件: standard input
输出文件: standard output
时间限制: 1 second
内存限制: 256 megabytes

给定正整数 a, c, p , 保证 p 是素数, 求 b 使得:

$$a^b \equiv b^c \pmod{p}$$

我们称整数 A, B, C 有 $A \equiv B \pmod{C}$, 当且仅当存在整数 k 使得 $A - B = C \times k$ 。

输入格式

输入仅一行三个整数 a, c 和 p ($1 \leq a, c < p \leq 10^9$), 由一个空格隔开, 含义如题面所述。

数据保证 p 是素数。

输出格式

输出仅一个整数 b ($1 \leq b \leq 10^{18}$), 如果有多个合法答案, 你可以输出任意一个。

可以证明, 在范围内至少存在一个解。

样例

standard input	standard output
3 5 7	16
14530529 19260817 19491001	5660025

提示

对于第一组样例, 我们有:

$$3^{16} \equiv 16^5 \pmod{7}$$

因为:

$$3^{16} \bmod 7 = 43046721 \bmod 7 = 4$$

$$16^5 \bmod 7 = 1048576 \bmod 7 = 4$$