

Problem A. 莫卡与 MCPC

输入文件: standard input
输出文件: standard output
时间限制: 1 second
空间限制: 2048 megabytes

作为全球领先的人工智能软件公司, 商汤科技以“坚持原创, 让 AI 引领人类进步”为使命, 以“人工智能实现物理世界和数字世界的连接, 促进社会生产力可持续发展, 并为人们带来更好的虚实结合生活体验”为愿景, 旨在持续引领人工智能前沿研究, 持续打造更具拓展性更普惠的人工智能软件平台, 推动经济、社会和人类的发展, 并持续吸引及培养顶尖人才, 共同塑造未来。

莫卡正在为商汤科技筹划一场竞赛! 她将这场竞赛命名为 MCPC (Mocha Cable Protection Contest)。这场竞赛需要大量服务器来评测选手给出的方案。可是作为一个魔法少女, 莫卡显然不需要租用服务器, 她借助魔法改造了一筐土豆并将它们作为 MCPC 的服务器。

准备好题目后, 莫卡邀请 wolf war 队来测试题目。但是这次, 莫卡的魔法出现了 bug, 土豆的性能甚至不足以支撑一支队伍的正常使用。wolf war 队每次访问都会得到一个状态码和一条消息, 当且仅当状态码既是质数又是偶数时, wolf war 队可以成功访问竞赛网站并得到一条内容为“OK”的消息, 否则 wolf war 队访问失败并得到一条内容为“An invalid response was received from the upstream server”的消息。

现在给出 wolf war 队每次访问得到的状态码, 请你判断他们每次访问得到的消息内容。

输入格式

第一行包含一个整数 n ($1 \leq n \leq 1000$), 表示 wolf war 队的访问次数。

接下来 n 行, 第 i 行包含一个正整数 x ($1 \leq x \leq 10^6$), 表示 wolf war 队第 i 次访问得到的状态码。

输出格式

输出 n 行, 第 i 行包含一个字符串, 若第 i 次访问成功, 输出 OK, 否则输出 An invalid response was received from the upstream server。注意不要输出多余空格。

样例

standard input	standard output
3	OK
2	An invalid response was received from
3	the upstream server
4	An invalid response was received from
	the upstream server

提示

第一次访问时, 状态码为 2, 满足要求, 因此输出 OK。

第二次访问时, 状态码为 3, 是质数但不是偶数, 因此输出 An invalid response was received from the upstream server。

为了美观，样例输出在 `from` 和 `the` 之间添加了换行符。你不应输出这个换行符。你可从 DOMjudge 中下载真实输出。

Problem B. bb 去食堂

输入文件: standard input

输出文件: standard output

时间限制: 1 second

空间限制: 2048 megabytes

bb 是北京航空航天大学的新生。

bb 踏进校门的那一刻，就被眼前亚洲最大的单体教学楼——新主楼吸引了眼球。震撼于它的庞大和精巧，bb 决定一有空闲时间就去新主楼自习。

在自习后，bb 会选择到一个离新主楼最近的食堂吃饭。具体地，他会将地图抽象为一个平面直角坐标系，并将新主楼看作一个四条边均与坐标轴平行的矩形，将各个食堂看作点。一个食堂到新主楼的最近距离定义为表示食堂的点与表示新主楼的矩形上所有点（包括四条边上的点和四个顶点）的欧几里得距离的最小值。其中，点 (x_1, y_1) 与点 (x_2, y_2) 的欧几里得距离为 $\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$ 。如果有多个距离新主楼最近的食堂，bb 会选择其中编号最小的一个。

刚开始，bb 只需要在学校的几个食堂中做出选择。但渐渐地，bb 发现除了学校的食堂外，南门的美食苑，东门的麦当劳，北门的烤鱼等地方，都是吃东西的好去处，因此他把这些地方也视为食堂。随着食堂数量增多，bb 不知道应该选择哪个了，你能帮帮他吗？

输入格式

第一行包含一个整数 n ($1 \leq n \leq 10^5$)，表示食堂的个数。

第二行包含四个整数 X_1, Y_1, X_2, Y_2 ($-100 \leq X_1 < X_2 \leq 100, -100 \leq Y_1 < Y_2 \leq 100$)， (X_1, Y_1) 表示新主楼左下角的坐标， (X_2, Y_2) 表示新主楼右上角的坐标。

接下来 n 行，第 i 行包含两个整数 x_i, y_i ($-100 \leq x_i, y_i \leq 100$)，表示编号为 i 的食堂的坐标为 (x_i, y_i) 。保证不会有食堂在新主楼的内部或边界上。

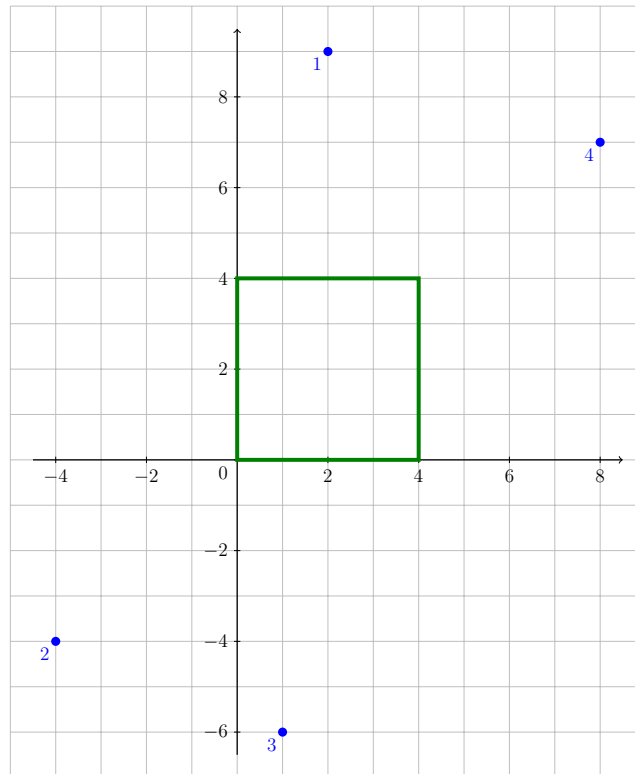
输出格式

输出一行一个整数，表示 bb 选择的食堂编号。

样例

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

提示



如图所示：

- 编号为 1 的食堂离新主楼的最近距离为 5。
- 编号为 2 的食堂离新主楼的最近距离为 $4\sqrt{2}$ 。
- 编号为 3 的食堂离新主楼的最近距离为 6。
- 编号为 4 的食堂离新主楼的最近距离为 5。

在两个最近距离为 5 的食堂中，bb 会选择编号较小的 1 号食堂。

Problem C. 抵御阿草

输入文件: standard input
输出文件: standard output
时间限制: 1 second
空间限制: 2048 megabytes

这天在沙滩上晒着太阳的 Eileen 发现阿草又一次来进攻枝江沙滩了, 然而这一次没有一个魂的帮助, 胆小好色的 Eileen 只能寻求队长 Bella 的帮助。

Bella 发现此次进攻枝江沙滩的阿草一共有 2^n 只, 它们站成了一排来进攻枝江沙滩, 第 i 只阿草初始有 $a_i = i$ 点美味度 ($i = 0, 1, \dots, 2^n - 1$)。Bella 会使用她的铁棒横扫 n 次, 每次横扫会将所有偶数位置 i 的阿草与位置 $i + 1$ 的阿草合并成一只新阿草。新阿草的美味度 $b_{i/2} = a_i \text{ op } a_{i+1}$, 其中 op 是与、或、异或中的一种运算。在同一次横扫中, 对于所有的 i , op 是同一种操作。显然 Bella 每次横扫之后阿草的数量就会减半。

当最后只剩下一只阿草的时候, 吃货 Diana 会突然出现将其吃掉, Diana 想知道最后吃掉的阿草美味度是多少。

由于阿草的美味度可能会很大, 因此只需要输出最终阿草美味度的二进制表示。

输入格式

第一行包含一个正整数 n ($0 \leq n \leq 10^5$)。

n 不为 0 时, 第二行包含一个长度为 n 的字符串, 每个位置为 0, 1, 2, 分别代表与、或、异或。

输出格式

输出一行一个字符串, 为最终美味度的二进制表示。输出不应包含前导 0。

样例

standard input	standard output
2 10	1
3 012	100

提示

对于样例一, 合并的过程为:

$$(0, 1, 2, 3) \rightarrow (1, 3) \rightarrow (1)$$

对于样例二, 合并的过程为:

$$(0, 1, 2, 3, 4, 5, 6, 7) \rightarrow (0, 2, 4, 6) \rightarrow (2, 6) \rightarrow (4)$$

题面中的与、或、异或分别是按位与、按位或、按位异或。

Problem D. 皮卡丘与宝可梦对战模拟器

输入文件: standard input

输出文件: standard output

时间限制: 5 seconds

空间限制: 2048 megabytes

皮卡丘最近迷上了宝可梦对战, 他希望编写一个对战用的模拟器。

影响宝可梦对战能力的重要因素之一是等级, 宝可梦等级越高, 对战能力也越强。

每只宝可梦有 6 项能力, 即 HP、攻击、防御、特攻、特防、速度。每项能力由三个因素决定, 即该项能力对应的种族值、个体值、努力值。它们分别代表该宝可梦所属种类固有的能力、该宝可梦个体先天的能力以及该宝可梦个体后天的能力。计算公式如下:

$$\begin{aligned} \text{HP 能力值} &= \left\lfloor \frac{\lfloor \text{种族值} \times 2 + \text{个体值} + \text{努力值}/4 \rfloor \times \text{等级}}{100} + 10 + \text{等级} \right\rfloor \\ \text{其它能力值} &= \left\lfloor \frac{\lfloor \text{种族值} \times 2 + \text{个体值} + \text{努力值}/4 \rfloor \times \text{等级}}{100} + 5 \right\rfloor \end{aligned}$$

每只宝可梦有 4 个招式, 每个招式有其固定的威力。招式分为物理招式和特殊招式, 物理招式的伤害使用攻击方的攻击和防御方的防御计算, 特殊招式的伤害使用攻击方的特攻和防御方的特防计算。伤害计算公式如下:

$$\text{伤害} = \left\lfloor \left(\frac{2 \times \text{等级} + 10}{250} \times \frac{\text{攻击}}{\text{防御}} \times \text{威力} + 2 \right) \times \text{随机数} \right\rfloor$$

其中等级为攻击方的等级, 随机数取值范围是 $[\frac{17}{20}, 1]$ 。显然在题目条件下, 每次使用招式至少造成 1 点伤害。

宝可梦对战采用回合制, 对战持续多个回合, 直到某只宝可梦输掉比赛。每个回合双方先任意选取自己的一个招式, 速度值大的宝可梦先使用招式, 速度值小的宝可梦后使用招式。若两只宝可梦速度值相同, 那么两只宝可梦各有 0.5 的概率先手。宝可梦开始对战时, 它的初始 HP 值为 HP 能力值, 每次受到招式伤害时, HP 值扣减相应的伤害。当 HP 值小于等于 0 时, 该宝可梦将失去战斗能力, 即输掉比赛。

现在, 有 n 只宝可梦参与对战, 皮卡丘希望对每两只宝可梦 A 和 B , 计算出 A 与 B 进行对战时, A 是否有可能取得胜利。这里, 有可能意味着 B 不一定选取最优的招式。请你帮它计算一下。

输入格式

第一行包含一个整数 n ($1 \leq n \leq 100$)。

接下来依次给出每只宝可梦的信息。

对于每只宝可梦, 第一行包含一个整数 Lv ($1 \leq Lv \leq 100$), 表示宝可梦的等级。

第二行给出 6 项种族值, 格式为 $SSs: SS1/SS2/SS3/SS4/SS5/SS6$, 其中 $SS_1, SS_2, \dots, SS_6$ 均为整数, $5 \leq SS_i \leq 255$, 分别表示该宝可梦 HP、攻击、防御、特攻、特防、速度的种族值。

第三行给出 6 项个体值, 格式为 $IVs: IV1/IV2/IV3/IV4/IV5/IV6$, 其中 $IV_1, IV_2, \dots, IV_6$ 均为整数, $0 \leq IV_i \leq 31$, 分别表示该宝可梦 HP、攻击、防御、特攻、特防、速度的个体值。

第四行给出 6 项努力值, 格式为 **EVs: EV1/EV2/EV3/EV4/EV5/EV6**, 其中 $EV_1, EV_2, \dots, EV_6$ 均为整数, $0 \leq EV_i \leq 252$, 分别表示该宝可梦 HP、攻击、防御、特攻、特防、速度的努力值。

第五行至第八行, 每行描述该宝可梦所拥有的一个招式, 格式为 **- Power/Type**。其中 Power 为整数, $10 \leq \text{Power} \leq 250$, 表示招式的威力。Type 可能为 **Physical** 或 **Special**, 分别表示物理招式和特殊招式。

请参考样例以进一步理解输入格式。

输出格式

输出 n 行, 第 i 行输出一个包含 n 个字符的字符串。若第 i 只宝可梦与第 j 只宝可梦对战时, 第 i 只宝可梦可能获胜, 那么第 i 行第 j 个字符输出 1, 否则第 i 行第 j 个字符输出 0。特别地, 由于宝可梦不能跟自己对战, 第 i 行第 i 个字符输出 **X**。

样例

standard input	standard output
2 100 SSs: 105/180/100/180/100/115 IVs: 31/31/31/31/31/31 EVs: 4/252/0/0/0/252 - 120/Physical - 150/Special - 110/Special - 90/Physical 1 SSs: 45/30/35/20/20/45 IVs: 0/0/0/0/0/0 EVs: 0/0/0/0/0/0 - 40/Physical - 60/Physical - 55/Special - 40/Special	X1 OX
This sample input is too long. Please download it from DOMjudge.	X101001010 OX00000000 11X1111111 110X000100 1111X11111 11111X1111 110100X110 1111001X11 11010011X0 110110111X

提示

请勿代入真实的宝可梦游戏。

$\lfloor x \rfloor$ 表示 x 向下取整。本题使用浮点数会产生精度误差，建议使用分数代替浮点数进行计算。若 a 为非负整数， b 为正整数，那么 $\lfloor \frac{a}{b} \rfloor$ 在 C/C++/Java 语言中可用 a / b 计算，在 Python 语言中可用 $a // b$ 计算。

样例一中，宝可梦 1 的能力值为 352/459/236/396/236/329，宝可梦 2 的能力值为 11/5/5/5/5/5。宝可梦 1 对宝可梦 2 每回合至多能造成 9981 伤害，宝可梦 2 对宝可梦 1 每回合至多能造成 2 伤害。因此显然宝可梦 1 可以战胜宝可梦 2，而宝可梦 2 不能战胜宝可梦 1。

Problem E. 圣莫卡造题的七天

输入文件: standard input
输出文件: standard output
时间限制: 1 second
空间限制: 2048 megabytes

“Santa Mocha, the light of my life, the fire of my desire. My sin, my soul.”

在惯例的默念之后, Bokazo31 开始今天的阅读。今天, Bokazo31 读到了《圣莫卡造题的七天》, 这是一本 LLJeo 撰写的记录圣莫卡事迹的书籍, LLJeo 这样写到:

七天过后, 圣莫卡造出了一道题——给定一个带有 n 个节点 m 条边的无向图, 节点 i 带有一个点权 w_i 。每次操作可以选择一条路径 $v_1 \rightarrow v_2 \rightarrow \cdots \rightarrow v_t$, 路径中任意相邻的两点之间有边。操作时将路径上第 i 个点的点权变为 $w_{v_i} \& w_{v_{t+1-i}}$, 其中 $\&$ 代表位运算中的与运算。每次操作点权的变化是**所有点同时**进行的。现在你可以执行任意次操作, 使得点权的最大值最小, 并告诉圣莫卡点权的最大值最小可以是多少。

虽然 Bokazo31 怀疑 LLJeo 可能修改了这道题的部分内容, 但是作为一个信徒, 他认为还是有做出这道题的必要。然而他发现自身能力并不足够, 因此他向你求助, 请你帮他计算出答案。

输入格式

第一行包含两个整数 n, m ($1 \leq n \leq 5 \times 10^5, 0 \leq m \leq \min(5 \times 10^5, \frac{n(n-1)}{2})$)。

第二行包含 n 个整数, 第 i 个整数表示 w_i ($0 \leq w_i \leq 10^9$)。

接下来 m 行, 第 i 行包含两个整数 u_i, v_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i$), 表示节点 u_i, v_i 之间存在一条边。保证不存在重边, 即对于 $\forall 1 \leq i < j \leq m$, 集合 $\{u_i, v_i\}$ 不等于集合 $\{u_j, v_j\}$ 。

输出格式

输出一个整数, 代表点权的最大值最小可以是多少。

样例

standard input	standard output
4 2 1 2 2 3 1 2 3 4	2
5 7 2 2 3 3 6 1 2 2 3 3 4 4 5 5 1 3 5 4 1	2

提示

第一个样例中, Bokazo31 可以选择路径 $1 \rightarrow 2$ 操作, 此时 1,2 权值均为 0。之后可以选择路径 $3 \rightarrow 4$ 操作, 此时 3,4 权值均为 2, 最大点权为 2。可以证明不能得到更小的最大点权。

第二个样例中, Bokazo31 可以选择路径 $3 \rightarrow 5$ 操作, 此时 3,5 权值均为 2。之后可以选择路径 $2 \rightarrow 3 \rightarrow 4$ 操作, 此时 2,3,4 权值均为 2, 最大点权为 2。可以证明不能得到更小的最大点权。

Problem F. PotasHub Copilot

输入文件: standard input

输出文件: standard output

时间限制: 1 second

空间限制: 2048 megabytes

在代码需求日益增长的当下, 自动补全甚至代码预测功能显得尤为重要。为了让编写代码更轻松, 小钾制造出了 PotasHub Copilot 替自己完成这项工作。

当然, PotasHub Copilot 的功能尚不完善, 只能从训练的代码数据中不太智能地截取一部分代码, 并利用该代码片段进行自动补全。具体地说, 假设训练数据是一段 n 行的代码, 那么 PotasHub Copilot 将会随机选择一段连续的行数区间 $[l, r]$ ($1 \leq l \leq r \leq n$), 将这些行对应的代码片段截取下来。

我们记 $\text{rank}(S, x)$ 为将集合 S 按从小到大的顺序排列后 x 的排名, 即 x 是 S 中第 $\text{rank}(S, x)$ 小的数。对于一段 n 行且每行长度分别为 $a_1, a_2, \dots, a_n$ 的代码, 小钾定义第 i 行代码的价值 $f(i)$ 为所有包含了这行代码的截取方案 $[l, r]$ 中, $\text{rank}(\{a_l, a_{l+1}, \dots, a_r\}, a_i)$ 之和, 即

$$f(i) = \sum_{l=1}^i \sum_{r=i}^n \text{rank}(\{a_l, a_{l+1}, \dots, a_r\}, a_i)$$

注意到 f 的值只与每行的相对代码长度有关, 为了方便计算, 小钾已经将绝对代码长度调整为相对代码长度, 即 $a_1, a_2, \dots, a_n$ 是一个 $1, 2, \dots, n$ 的排列。现在小钾希望你计算每一行代码的价值 $f(1), f(2), \dots, f(n)$ 。

输入格式

第一行包含一个整数 n ($1 \leq n \leq 5 \times 10^5$)。

第二行包含 n 个整数 $a_1, a_2, \dots, a_n$ 。保证 $a_1, a_2, \dots, a_n$ 是一个 $1, 2, \dots, n$ 的排列。

输出格式

输出 n 行, 第 i 行包含一个整数, 表示 $f(i)$ 。

样例

standard input	standard output
5 3 1 2 5 4	12 8 15 24 11
8 4 1 2 8 5 7 3 6	23 14 30 90 54 69 24 26

提示

在第一个样例中，每个数的计算方法如下：

- $a_1 = 3$ ，在 $[1, 1], [1, 2], [1, 3], [1, 4], [1, 5]$ 区间中的排名分别为 1, 2, 3, 3, 3，故 $f(1) = 1 + 2 + 3 + 3 + 3 = 12$ ；
- $a_2 = 1$ ，在 $[1, 2], [1, 3], [1, 4], [1, 5], [2, 2], [2, 3], [2, 4], [2, 5]$ 区间中的排名分别为 1, 1, 1, 1, 1, 1, 1, 1，故 $f(2) = 1 + 1 + 1 + 1 + 1 + 1 + 1 + 1 = 8$ ；
- $a_3 = 2$ ，在 $[1, 3], [1, 4], [1, 5], [2, 3], [2, 4], [2, 5], [3, 3], [3, 4], [3, 5]$ 区间中的排名分别为 2, 2, 2, 2, 2, 2, 1, 1, 1，故 $f(3) = 2 + 2 + 2 + 2 + 2 + 2 + 1 + 1 + 1 = 15$ ；
- $a_4 = 5$ ，在 $[1, 4], [1, 5], [2, 4], [2, 5], [3, 4], [3, 5], [4, 4], [4, 5]$ 区间中的排名分别为 4, 5, 3, 4, 2, 3, 1, 2，故 $f(4) = 4 + 5 + 3 + 4 + 2 + 3 + 1 + 2 = 24$ ；
- $a_5 = 4$ ，在 $[1, 5], [2, 5], [3, 5], [4, 5], [5, 5]$ 区间中的排名分别为 4, 3, 2, 1, 1，故 $f(5) = 4 + 3 + 2 + 1 + 1 = 11$ ；

Problem G. Serval 的字符串

输入文件: standard input
输出文件: standard output
时间限制: 1 second
空间限制: 2048 megabytes

Serval 是加帕里幼儿园的新生。

Serval 有一个仅包含小写英文字母的字符串 S ，它的长度记作 $|S|$ 。

有 q 个好奇的小朋友，第 i 个小朋友有一个仅包含小写英文字母的字符串 T_i 。第 i 位小朋友想知道，他的字符串 T_i 无限拼接得到的 $T_iT_iT_i\dots$ 截取前 $|S|$ 个字符得到的字符串，有多少个位置的字符与 S 不同。

输入格式

第一行包含一个字符串 S ($1 \leq |S| \leq 2 \times 10^5$)。 S 仅包含小写英文字母。

第二行包含一个整数 q ($1 \leq q \leq 2 \times 10^5$)。

接下来 q 行，第 i 行包含一个字符串 T_i ($1 \leq |T_i| \leq 2 \times 10^5$)。 T_i 仅包含小写英文字母。保证所有 T_i 长度之和 $\sum_{i=1}^q |T_i| \leq 2 \times 10^5$ 。

输出格式

输出 q 行，第 i 行包含一个整数，表示 T_i 无限拼接得到的 $T_iT_iT_i\dots$ 截取前 $|S|$ 个字符得到的字符串，有多少个位置的字符与 S 不同。

样例

standard input	standard output
abcbac	3
6	2
ac	4
abc	0
a	1
abcbacd	1
abcb	
accb	

提示

样例中， $S = \text{abcbac}$ ，第一个小朋友的字符串 $T_1 = \text{ac}$ 。 T_1 无限拼接得到 $\text{acacacac}\dots$ ，截取前 $|S|$ 个字符得到 acacac ， abcbac 与 acacac 有 3 个位置的字符不同。

Problem H. 狂乱

输入文件: standard input

输出文件: standard output

时间限制: 1 second

空间限制: 2048 megabytes

在炉石传说游戏里，有一张牌叫狂乱。



它的效果是：选择一个随从，令其随机地攻击其他随从，直至该随从死亡或者其他随从全部死亡。

现有 n 个随从在场上，第 i 个随从拥有攻击力 atk_i 和血量 hp_i 。不考虑随从的其它各种特效，每次攻击时，参与攻击的双方的血量各自减少对方的攻击力。当某个随从的血量小于或等于 0 时，则认为该随从死亡。

例如：一个 5 攻 2 血的随从攻击一个 1 攻 11 血的随从，攻击之后 5 攻 2 血的随从会变成 5 攻 1 血，而 1 攻 11 血的随从则会变成 1 攻 6 血。

如果在 10^{30} 次攻击后，依然有随从存活，则“狂乱”会自行停止。

请问是否有可能，对一个指定随从使用“狂乱”，在运气足够好的情况下，最终消灭所有随从（包括指定随从）？

输入格式

第一行包含一个整数 n ($1 \leq n \leq 10^5$)。

接下来 n 行，第 i 行包含两个整数 $\text{atk}_i, \text{hp}_i$ ($0 \leq \text{atk}_i \leq 10^6, 1 \leq \text{hp}_i \leq 10^6$)，表示第 i 个随从的攻击力和血量。

输出格式

输出一行一个字符串。如果最终能够消灭所有随从，则输出 YES，否则输出 NO（大小写不敏感）。

样例

standard input	standard output
2 2 6 3 3	YES
3 2 3 3 2 3 3	YES
3 2 2 2 2 2 2	NO
5 30 30 2 7 5 1 12 12 1 4	NO
2 1 10 2 5	YES

提示

对于第一个样例，无论选择哪个随从，两个随从都会交战两次并同时死亡，可以满足要求。

对于第二个样例，选择狂乱 3 攻 3 血的随从，并先攻击 2 攻 3 血的随从、后攻击 3 攻 2 血的随从，可以消灭所有的随从，可以满足要求。

对于第三个样例，无论如何都只能消灭其中的 2 个随从，不可能满足要求。

对于第四个样例，在狂乱 30 攻、30 血的随从后，它会消灭所有其他随从，但是自己依然存活，不可能满足要求。

Problem I. Serval 的相对论

输入文件: standard input

输出文件: standard output

时间限制: 2 seconds

空间限制: 2048 megabytes

本题为交互题。

Serval 是加帕里幼儿园的新生。

Serval 最近发明了一款风靡校园的小游戏：相对论。

这个游戏是玩家与 Serval 的小 bot 之间的“博弈”。小 bot 要来抓玩家，而玩家获胜的条件就是不被小 bot 抓到。

游戏在平面直角坐标系中有限的区域内进行，玩家与小 bot 只能在格点上移动，任意一方的坐标 (x, y) 需要满足 $|x| \leq 50, |y| \leq 50$ ，合法格点所在的这个正方形区域称为**游戏区域**。

当玩家位于 (x, y) ，小 bot 位于 (x', y') 时，我们定义玩家与 bot 之间的**距离**为 $d = |x - x'| + |y - y'|$ 。玩家被小 bot 抓到是指在某个时刻玩家与小 bot 位于相同位置。

初始时玩家位于 $(0, 0)$ ，而小 bot 可能位于任意格点 (x, y) 。玩家**不知道**小 bot 初始位置 (x, y) 的具体值，但知道小 bot 与玩家的**初始距离一定不小于 4**。

接下来玩家与小 bot 双方会进行多轮移动。在一轮移动中，玩家可以选定上下左右中的一个方向移动 1 单位距离，但必须保证移动后仍在游戏区域内。此外，**玩家还可以选择在这一轮中不移动**。小 bot 则会选择一个靠近玩家的方向移动 1 单位距离，且**不会选择不移动**。

更具体地说，在一轮移动中，玩家与小 bot 双方**依次**完成下述操作：

1. 玩家得到此时与小 bot 之间的距离。
2. 小 bot 根据玩家此时的坐标选定本轮小 bot 移动的方向。

设玩家位于 (x, y) ，小 bot 位于 (x', y') ，小 bot 会按如下方式选择移动的方向：

- 若 $x \neq x'$ ，则小 bot 会选择 $|x - x'|$ 减少的方向移动，而 y' 不变。
- 若 $x = x'$ ，则小 bot 会选择 $|y - y'|$ 减少的方向移动，而 x' 不变。

3. 玩家选择本轮是否移动，若选择移动还需要选定移动方向。
4. 小 bot 向选定的移动方向移动 1 单位距离。显然移动后小 bot 仍位于游戏区域内。
5. 若玩家选择移动，则玩家向选定的移动方向移动 1 单位距离。
6. 如果玩家被小 bot 抓到（即在某一时刻与小 bot 处于同一位置），或是移动到游戏区域外，则游戏结束，小 bot 获胜。

现在 Serval 向你发起了挑战。你作为玩家需要制定你的策略获得“相对论”小游戏的胜利。课间时间只够进行 5×10^4 轮移动，如果 5×10^4 轮移动后你仍然没有被小 bot 抓到，那么游戏结束，你获得胜利啦！

输入格式

本题无输入，与交互器的交互方式参见交互格式。

输出格式

对于游戏进行时的每一轮次，你需要选择是否移动，若选择移动还需要选定移动方向。具体来说，你需要输出一行，包含“UDLRX”其中一个字符，字符含义如下：

- “U”：本轮向上移动，即从 (x, y) 移动到 $(x, y + 1)$ 。
- “D”：本轮向下移动，即从 (x, y) 移动到 $(x, y - 1)$ 。
- “L”：本轮向左移动，即从 (x, y) 移动到 $(x - 1, y)$ 。
- “R”：本轮向右移动，即从 (x, y) 移动到 $(x + 1, y)$ 。
- “X”：本轮不移动。

交互格式

本题采用输入输出进行交互。交互器的输出将作为你的程序的输入，你的程序的输出将作为交互器的输入。更多细节请参考[此文档](#)。

每一轮移动开始时，交互器会向你返回一个整数 d ，含义如下：

- 若 d 为正整数则表示游戏尚未结束，此时 d 表示你与小 bot 之间的距离。
- 若 $d = 0$ 则表示经过 5×10^4 轮移动，游戏结束且你获胜。
- 若 $d = -1$ 则表示你已经被小 bot 抓到，或是移动到游戏区域外，游戏结束且小 bot 获胜。

若 d 为 0 或 -1 ，此时游戏结束，你需要立即终止程序运行，否则由于程序尝试从关闭的输入流获取输入等情况，你可能得到意料之外的评测结果（如 **Wrong Answer** 或 **Time Limit Exceeded**）。在程序正常结束的情况下，若 d 为 -1 ，表示小 bot 获胜，交互器将返回 **Wrong Answer** 评测结果。

若 d 为正整数，此时游戏尚未结束，你需要输出一行，包含“UDLRX”其中一个字符。行末需要换行，并且在输出之后需要刷新输出流，否则你可能得到意料之外的评测结果（如 **Time Limit Exceeded**）。你可以通过以下方式刷新输出流：

- 对于 C++ 语言，使用 `fflush(stdout)` 或者 `cout.flush()`。
- 对于 C 语言，使用 `fflush(stdout)`。
- 对于 Java 语言，使用 `System.out.flush()`。
- 对于 Python 语言，需要载入 `sys` 模块，使用 `sys.stdout.flush()`。
- 对于 Pascal 语言，使用 `flush(output)`。
- 对于其他语言，请参考有关文档。

游戏至多会进行 5×10^4 轮移动，这意味着你至多需要读取 $5 \times 10^4 + 1$ 个整数，至多需要输出 5×10^4 行，每行包含一个字符。

如果 5×10^4 轮移动后你仍然没有被小 bot 抓到，那么游戏结束且你获得胜利。

样例

standard input	standard output
6	U
4	L
4	L
4	D
4	X
3	D
3	R
0	
4	U
2	U
-1	

提示

样例仅为说明交互方式，不代表按照样例的方式进行交互可以通过相应测试点。

在第一个样例中，小 bot 的初始位置为 (3,3)。可以在[此处](#)观看每轮移动的可视化过程。限于题面长度，7 轮移动后交互器返回整数 0，表示游戏结束且玩家获胜。请注意正式评测时需要进行 5×10^4 轮移动且未被抓到才能获胜。

在第二个样例中，小 bot 的初始位置为 (0,4)。由于 2 轮移动之后玩家被抓到，交互器返回整数 -1，表示游戏结束且小 bot 获胜。这样的策略在正式评测中将被判为 Wrong Answer。

Problem J. 数据重命名

输入文件: standard input
输出文件: standard output
时间限制: 1 second
空间限制: 2048 megabytes

闲来无聊，小钾开始造数据。所有的数据都在某个文件夹下存放，每条数据由一个独立的文件存储，并且拥有独一无二的文件名。

在该文件夹下，小钾一共造了 n 条数据，这些数据文件以列表的形式排成 n 行，每行代表一个文件。同时，系统总是会自动按照文件名的字典序排列文件夹下的文件，排序后从上至下为升序。如果该文件夹下某些文件的文件名发生了变更，那么系统同样会自动为所有文件重新按字典序排序。

不巧的是，小钾发现一开始的 n 条数据命名存在问题，因此他决定为每条数据都手动重命名。具体来说，对于第 i 条数据，其初始字典序为 p_i ，重命名后的字典序为 q_i 。重命名后文件名的字典序总是比原来所有的文件名都小。即是说，如果把这 $2n$ 个重命名前后的文件名按字典序升序排列，那么前 n 个都是重命名后的，后 n 个都是重命名前的。

小钾的重命名过程如下：将光标用上方向键或下方向键移动至某个还未被重命名的文件上，按下 F2 将其重命名，接着系统自动为所有文件重新排序，光标将跟随被重命名的文件到达新的位置。这样的操作将重复 n 次，直到所有文件都重命名完成。初始时，光标停留在从上至下的第一个文件上。当按下一次上方向键或下方向键时，光标会移动到列表中的上一个或下一个文件上。光标停留在最顶上（或最底下）的文件上时，无法按上方向键（或下方向键）继续移动。

现在小钾想知道，为了重命名所有 n 个文件，需要按上方向键和下方向键的最少次数。你能告诉他吗？

输入格式

第一行包含一个整数 n ($1 \leq n \leq 5 \times 10^5$)。

第二行包含 n 个整数 $p_1, p_2, \dots, p_n$ ，含义如题所示。保证 $p_1, p_2, \dots, p_n$ 一定是一个 $n+1, n+2, \dots, 2n$ 的排列。

第三行包含 n 个整数 $q_1, q_2, \dots, q_n$ ，含义如题所示。保证 $q_1, q_2, \dots, q_n$ 一定是一个 $1, 2, \dots, n$ 的排列。

输出格式

输出一行一个整数，表示为了重命名所有 n 个文件，需要按上方向键和下方向键的最少次数。

样例

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

提示

在第一组数据中，小钾可以依次重命名第 1,2,3 条数据。用下划线表示光标的位置，则光标移动方案如下：

- 初始时文件名升序为 [4,5,6]；
- 将文件名 4 重命名为 2，文件名升序为 [2,5,6]；
- 按动 1 次下方向键将光标移动到文件名 5 上，文件名升序为 [2,5,6]；
- 将文件名 5 重命名为 1，文件名升序为 [1,2,6]；
- 按动 2 次下方向键将光标移动到文件名 6 上，文件名升序为 [1,2,6]；
- 将文件名 6 重命名为 3，文件名升序为 [1,2,3]。

最少次数为 $1 + 2 = 3$ 。方案可能不唯一，例如在该数据下小钾可以依次重命名第 1,3,2 条数据，最少次数同样为 3。

在第二组数据中，小钾可以依次重命名第 1,3,2,5,4 条数据，最少次数为 7。

Problem K. 皮卡丘与 Minimum Spanning Tree-I

输入文件: standard input

输出文件: standard output

时间限制: 2 seconds

空间限制: 2048 megabytes

皮卡丘在宝可梦学校学习到了最小生成树这个知识点, 维基百科对它的定义如下:

在一给定的无向图 $G = (V, E)$ 中, (u, v) 代表连接顶点 u 与顶点 v 的边 (即 $(u, v) \in E$), 而 $w(u, v)$ 代表此边的权重。若存在 T 为 E 的子集 (即 $T \subseteq E$) 且 (V, T) 为树, 使得

$$w(T) = \sum_{(u,v) \in T} w(u, v)$$

的 $w(T)$ 最小, 则此 T 为 G 的最小生成树。

自然, 皮卡丘今天的课后作业就是一道最小生成树题了。但是令他头疼的是, 题目中的图有些大。

题目中的图 G 是一个完全图, 即任意两点之间都有连边的图。点集 $V = \{1, 2, \dots, n\}$ 。每个点有一个点权, 点 i 的权为 a_i 。边集 $E = \{(u, v) | 1 \leq u < v \leq n\}$ 。其一个子集 E' (包括权值) 由输入给定, 对于其它的边 $(u, v) \in E - E'$, $w(u, v) = \min(a_u, a_v)$ 。

皮卡丘不太会做这道题, 请你帮他求出 G 的最小生成树, 即最小的 $w(T)$ 以及相应的 T 。

输入格式

第一行包含两个整数 n, m ($1 \leq n \leq 5 \times 10^5, 0 \leq m \leq \min(5 \times 10^5, \frac{n(n-1)}{2})$)。

接下来 m 行, 第 i 行包含三个整数 u_i, v_i 和 w_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i, 1 \leq w_i \leq 10^9$), 表示边 $(u_i, v_i) \in E'$, 其权值为 w_i 。保证无重边, 即对于 $\forall 1 \leq i < j \leq m$, 集合 $\{u_i, v_i\}$ 不等于集合 $\{u_j, v_j\}$ 。

接下来一行, 包含 n 个整数 $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$)。

输出格式

第一行输出一个整数, 表示最小的 $w(T)$ 。

接下来 $n - 1$ 行, 第 i 行输出两个整数 u_i, v_i , 表示最小的 $w(T)$ 对应的 T 中的一条边。

你所输出的 $n - 1$ 条边应当构成一棵树。你可以以任意顺序输出 T 中的边。你可以以任意顺序输出一条边中的两个点。

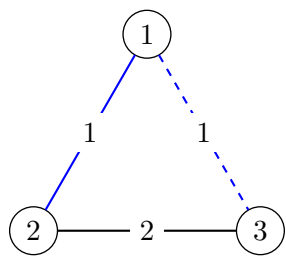
如果有多种可能的答案, 你可以输出任意一种。

样例

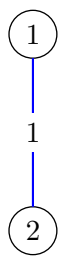
standard input	standard output
3 2 1 2 1 2 3 2 3 2 1	2 1 2 1 3
2 1 2 1 1 2 1	1 1 2
4 3 1 2 1 2 3 4 1 3 2 3 4 5 3	6 1 2 1 3 1 4

提示

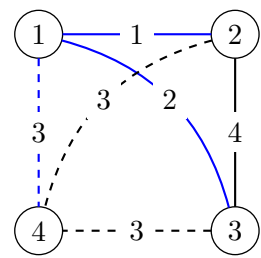
图中，实线表示输入的边，虚线表示非输入的边。蓝色边组成了最小生成树。



样例一



样例二



样例三

Problem L. 皮卡丘与 Minimum Spanning Tree-II

输入文件: standard input

输出文件: standard output

时间限制: 4 seconds

空间限制: 2048 megabytes

皮卡丘在宝可梦学校学习到了最小树形图这个知识点, 维基百科对它的定义如下:

树形图: 树形图是一个**有向图**; 并且对其中一个我们称呼作根的顶点 r , 以及任何其他顶点 u , 此图必然存在且只存在一条从 r 到 u 的路径。换句话说, 树形图是一个有向的, 有根的树, 并且所有的边都指离根的方向。

最小树形图: 在一给定的**有向图** $G = (V, E)$ 中, (u, v) 代表顶点 u 指向顶点 v 的边 (即 $(u, v) \in E$), 而 $w(u, v)$ 代表此边的权重。给定根 r , 若存在 T 为 E 的子集 (即 $T \subseteq E$) 且 (V, T) 为以 r 为根的树形图, 使得

$$w(T) = \sum_{(u,v) \in T} w(u, v)$$

的 $w(T)$ 最小, 则此 T 为 G 以 r 为根的最小树形图。

自然, 皮卡丘今天的课后作业就是一道最小树形图题了。但是令他头疼的是, 题目中的图有些大。

题目中的图 G 是一个完全有向图, 即任意两点之间都有一对相反方向边的图。点集 $V = \{1, 2, \dots, n\}$ 。每个点有一个点权, 点 i 的权为 a_i 。边集 $E = \{(u, v) | 1 \leq u, v \leq n, u \neq v\}$ 。其一个子集 E' (包括权值) 由输入给定, 对于其它的边 $(u, v) \in E - E'$, $w(u, v) = a_v$ 。

皮卡丘不太会做这道题, 请你帮他求出 G 分别以 $r = 1, 2, \dots, n$ 为根的最小树形图的权值, 即以每个点 i 为根最小的 $w_i(T_i)$ 。特别地, 对于 $r = r'$, 你要求出方案, 即 $w_{r'}(T_{r'})$ 对应的 $T_{r'}$ 。

输入格式

第一行包含三个整数 n, m, r' ($1 \leq n \leq 5 \times 10^5, 0 \leq m \leq \min(5 \times 10^5, n(n-1)), 1 \leq r' \leq n$)。

接下来 m 行, 第 i 行包含三个整数 u_i, v_i 和 w_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i, 1 \leq w_i \leq 10^9$), 表示边 $(u_i, v_i) \in E'$, 其权值为 w_i 。保证无重边, 即对于 $\forall 1 \leq i < j \leq m$, $u_i \neq u_j$ 或 $v_i \neq v_j$ 。

接下来一行, 包含 n 个整数 $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$)。

输出格式

第一行输出 n 个整数, 第 i 个整数表示以 i 为根的最小树形图的权值 $w_i(T_i)$ 。

接下来 $n-1$ 行, 第 i 行输出两个整数 u_i, v_i , 表示最小的 $w_{r'}(T_{r'})$ 对应的 $T_{r'}$ 中的一条边。

你所输出的 $n-1$ 条边应当构成一个以 r' 为根的树形图。你可以以任意顺序输出 $T_{r'}$ 中的边。

如果有多种可能的答案, 你可以输出任意一种。

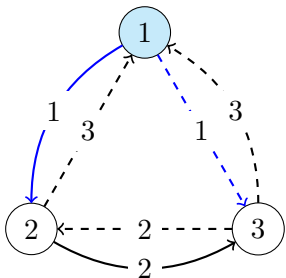
样例

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

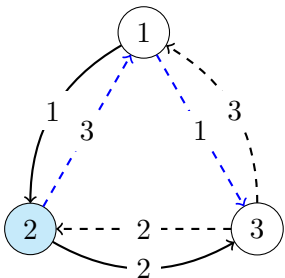
提示

图中，实线表示输入的边，虚线表示非输入的边。蓝色边组成了以浅蓝色点为根的最小树形图。

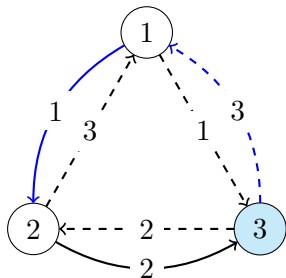
样例一：



以 1 为根

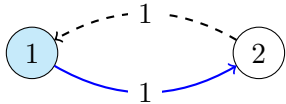


以 2 为根

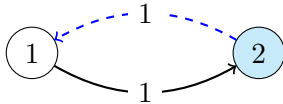


以 3 为根

样例二：



以 1 为根



以 2 为根