

Problem A. 生产机器

Input file: **standard input**
Output file: **standard output**
Time limit: 5 seconds
Memory limit: 1024 megabytes

某型号的作业机器人可以生产蓝色和黄色两种充能球。根据接下来 n 小时的工作计划，该机器人在第 i 小时可以生产至多 l_i 个黄色充能球和 f_i 个蓝色充能球，也可以不生产任何一个充能球。

在接下来 n 个小时的工作结束之后，该机器人生产的充能球将按其被生产的顺序依次装入集装箱。具体而言，每小时内生产的充能球的顺序可以由机器人任意确定，而前一小时生产的所有充能球都排在后一小时生产的充能球之前。

现在，请你计算，最终集装箱中装入的充能球所构成的序列共有多少种可能。两个序列不同当且仅当其包含的充能球总数不同，或者存在一个 i ，使得两个序列中生产的第 i 个充能球颜色不同。由于答案可能很大，你只需要输出答案对 $10^9 + 7$ 取模的结果。

Input

第一行一个整数 $t(1 \leq t \leq 1000)$ ，表示测试数据的组数。

对于每组测试数据：

第一行一个整数 $n(1 \leq n \leq 10^5)$ ，表明机器人的工作时长。

接下来 n 行，每行两个整数 l_i 和 f_i ($0 \leq l_i, f_i \leq 10^6$)，表明该机器人第 i 小时最多可以生产的两种充能球的个数。

保证对于所有数据的 $\sum n \leq 2 \times 10^6$ 。

Output

对于每组数据输出一行一个整数，表明所求的答案。

Example

standard input	standard output
2	4
1	389
0 3	
2	
2 3	
4 1	

Problem B. 纠缠点对

Input file: standard input
Output file: standard output
Time limit: 15 seconds
Memory limit: 1024 megabytes

给定两棵由 n 个顶点构成的无根树 T_1 和 T_2 。
定义点对 (u, v) 和 (x, y) 相互纠缠，当且仅当，在 T_1 上和 T_2 上，从 u 到 v 和从 x 到 y 的最短路径都有至少一个交点。
现给定 m 对点对 $(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)$ ，计算：有多少对点对相互纠缠。即，存在多少对 $i < j$ ，使得点对 (x_i, y_i) 和点对 (x_j, y_j) 相互纠缠。

Input

第一行一个整数 t ($1 \leq t \leq 10$)，表示测试数据的组数。
对于每组数据：
第一行两个整数 n ($1 \leq n \leq 10^5$) 和 m ($1 \leq m \leq 10^5$)。
接下来 $n - 1$ 行，每行两个整数 u, v ，表示 T_1 上的一条边。
接下来 $n - 1$ 行，每行两个整数 u, v ，表示 T_2 上的一条边。
接下来 m 行，每行两个整数 x, y ，表示给定的一个点对。

Output

对每组数据输出一行一个整数，表示所求答案。

Example

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

Problem C. 自动人偶

Input file: **standard input**
Output file: **standard output**
Time limit: 15 seconds
Memory limit: 1024 megabytes

某科学家制作了 n 个机器人，将他们依次编号为 $1 \sim n$ ，并放置在可抽象为一数轴的某地上。其中，第 i 个机器人被放置在数轴的坐标 i 所对应的位置。

该科学家制作的机器人可执行两种移动指令：‘L’ 和 ‘R’。

- 当执行 ‘L’ 指令时，机器人将沿数轴负方向移动一个单位长度。特别地，数轴的位置 1 左侧有一座峭壁，如果机器人在坐标 1 处执行 ‘L’ 指令，则其由于被阻挡而无法完成移动，而停留在原地。
- 当执行 ‘R’ 指令时，机器人将沿数轴正方向移动一个单位长度。特别地，数轴的位置 n 右侧有一座悬崖，如果机器人在坐标 n 处执行 ‘R’ 指令，则其将从悬崖上坠落，并被损坏。
- 特别地，机器人中内置了自毁程序。若在执行完某条指令之后，其之前执行的指令序列中出现了自毁指令（某已知字符串 S ）作为子串，则该机器人将立即爆毁，并被损坏。

现该科学家从所有长度为 m 的指令序列中等概率随机地选取了一个序列 T ，将其烧录至所有机器人的指令芯片中。这些机器人将依次执行 T 中的所有指令。每当执行完毕所有指令，则回到 T 的开头，从第一条指令开始重新执行，直至机器人由于坠落或爆毁而损坏。

请你计算，在充分长的时间之后，期望会剩下多少台没有损坏的机器人。换言之，期望下有多少台机器人可以执行任意多条指令而不被损坏。为避免精度误差，答案对 $10^9 + 7$ 取模。

Input

第一行一个整数 $t(1 \leq t \leq 10)$ ，表示测试数据的组数。

对于每组数据：

第一行两个整数 $n, m(1 \leq n \leq 30, 1 \leq m \leq 40)$ – 机器人的个数，以及指令序列的长度。

接下来一行一个仅包含字符 ‘L’ 和 ‘R’ 的字符串 $S(1 \leq |S| \leq 30)$ – 机器人的自毁指令。

Output

对于每组数据输出一行一个整数，表示所求答案。

Example

standard input	standard output
2	750000006
3 2	433410649
LR	
6 15	
LRLRL	

Problem D. 战争游戏

Input file: **standard input**
Output file: **standard output**
Time limit: 5 seconds
Memory limit: 512 megabytes

坎格鲁斯普雷和袋鼠将军在游玩一款名叫“战争游戏”的游戏，在这款游戏中，坎格鲁斯普雷是进攻方，袋鼠将军是防守方。

游戏的地图可以抽象为一张有着 n 个节点的树。初始时，防守方的人物在 s 号节点。

游戏将会进行 10^{100} 回合，在每一回合中，游戏的流程如下：

首先，进攻方会选择一个节点 p ，作为轰炸中心，并对防守方进行“轰炸预告”，如果在回合结束时防守方所在的节点 t 与轰炸中心 p 的距离 **不超过** 轰炸半径 r_1 ，那么防守方的人物将会被炸死，此时游戏结束，进攻方获胜。之后 防守方可以操纵他的人物移动到与当前位置的距离 **不超过** r_2 的节点上，然后回合结束。如果防守方的人物在回合结束时没被炸死，那么接着进行下一轮游戏，直到游戏轮次耗尽。若游戏轮次耗尽的时候防守方操纵的人物仍未死亡，那么防守方获胜，游戏结束。

作为袋鼠中的精英，坎格鲁斯普雷和袋鼠将军都是绝顶聪明的（即他们做出的操作都是当前盘面下的最优操作），那么在游戏结束时，谁将获胜？

Input

输入第一行一个整数 T ，表示测试数据组数。（ $1 \leq T \leq 10^3$ ）

每组测试数据，第一行四个整数 n, s, r_1, r_2 。（ $1 \leq n \leq 10^5, 1 \leq s, r_1, r_2 \leq n$ ）

之后 $(n - 1)$ 行，每行两个整数 u_i, v_i ，表示一条存在于树内的边。（ $1 \leq u_i, v_i \leq n$ ）

数据保证 $\sum n \leq 2 \times 10^6$ 。

Output

对于每组测试数据，若坎格鲁斯普雷获胜，输出一行一个字符串 `Kangaroo_Splay`；否则，输出一行一个字符串 `General_Kangaroo`。每组测试数据的答案之间需换行。

Example

standard input	standard output
2	Kangaroo_Splay
5 3 2 3	General_Kangaroo
1 2	
2 3	
3 4	
3 5	
5 1 1 3	
1 2	
2 3	
3 4	
3 5	

Problem E. 华丽牧场

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 1024 megabytes

π 酱正在攀登一座尖塔, 通过尖塔的某层需要依次经过两个房间, 第一个房间会触发一次事件, 第二个房间会进行一场战斗. 刚进入尖塔的 π 酱初始拥有一套卡组, 卡组包含 n 张牌, 第 i 张牌上写有数字 $a_i (a_i \geq 0)$, 表示打出这张牌会从抽牌堆中抽取 a_i 张牌.

一次事件的描述如下:

- 一张脸出现在墙壁上, 说道:"有所改变, 我就让你看见新的道路."于是 π 酱将牌组中的第 x 张牌上的数字变化为 y .

一场战斗的进行过程如下:

- 初始时卡组的 n 张牌按 $1 \sim n$ 的顺序排列在抽牌堆中. 手牌和弃牌堆初始为空.
- 一次**抽牌**指将抽牌堆顶的一张牌放入手牌. 若进行一次抽牌时抽牌堆为空, 则进行一次**洗牌**, 即将弃牌堆中所有牌按 $1 \sim n$ 的顺序排列在抽牌堆中, 之后再从抽牌堆顶的一张牌放入手牌. 特别地, 若此时弃牌堆中也没有牌, 则认为此次抽牌不生效 (但仍然触发了洗牌).
- 战斗过程进行若干个回合, 每一回合如下进行:
 - 我方回合开始, 进行 m 次抽牌.
 - 打出若干张手牌, 每打出一张牌, 先执行该牌的效果 (进行 a_i 次抽牌), 再将该牌弃入弃牌堆.
 - 我方回合结束, 将所有手牌弃入弃牌堆. (不考虑敌方回合).
- π 酱拥有技能"华丽牧场"将在每次触发洗牌时对敌方造成大量伤害. 一场战斗在敌方死亡时结束.

π 酱计划爬 q 层塔, 即交替进行 q 次换牌事件和 q 场战斗. 由于每场战斗敌方的血量很高, π 酱希望在某回合可以无限使用华丽牧场技能, 从而无限出伤消灭敌方, 且由于敌方每回合都会对他造成伤害, 他希望这个无限出伤的回合尽量早.

你的任务是帮助 π 酱计算并回答在这 q 场战斗中, 他最早能达到无限出伤的回合数分别是多少, 或告诉他这不可能(即使不可能无限出伤战斗也会正常结束, 不会影响接下来的战斗).

Input

每个测试点包含多组测试数据, 第一行包含一个正整数 $t (t \leq 10^3)$, 表示测试数据的组数, 下面是每组测试数据的描述:

第一行包含两个正整数 $n (n \leq 10^5)$, $m (m \leq n)$, 分别表示卡组大小和每回合开始时的抽牌数.

第二行包括 n 个非负整数, 第 i 个数 $a_i (a_i \leq n)$ 表示卡组第 i 张牌上的数字.

第三行包含一个正整数 $q (q \leq 10^5)$, 表示 π 酱计划爬 q 层塔.

接下来 q 行每行两个整数 $x (1 \leq x \leq n)$, $y (0 \leq y \leq n)$ 表示每层房间中变牌事件中变化的牌编号和变化后的数字.

保证 $\sum n \leq 5 \times 10^5$, $\sum q \leq 5 \times 10^5$, 最大的 n, q 满足 $n, q \leq 10^5$.

Output

q 行, 第 i 行表示第 i 个房间中进行的战斗最小的能够无限出伤的回合数, 如果没有这样的回合输出-1.

Example

standard input	standard output
2	-1
10 5	2
0 0 0 0 3 0 0 0 0 3	1
3	-1
10 2	1
5 4	
1 1	
10 5	
0 0 0 0 0 0 0 0 0 0	
2	
1 10	
2 1	

Problem F. 怯战蜥蜴 II

Input file: standard input
Output file: standard output
Time limit: 15 seconds
Memory limit: 1024 megabytes

坎格鲁斯普雷正在和袋鼠将军麾下的爪牙们进行决斗。

袋鼠将军麾下一共有 n 名爪牙，第 i 名爪牙的实力为 a_i ，坎格鲁斯普雷的实力为 c 。

坎格鲁斯普雷在和爪牙们决斗前，先会评估爪牙们的实力，坎格鲁斯普雷评估爪牙的实力时会存在至多 k 的误差。设坎格鲁斯普雷评估第 i 名爪牙的实力为 b_i ，则 b_i 满足 $a_i - k \leq b_i \leq a_i + k$ 。由于坎格鲁斯普雷喜欢随机化算法，所以 b_i 是在 $[a_i - k, a_i + k]$ 这个范围内均匀随机选取的一个整数。

由于坎格鲁斯普雷是怯战蜥蜴，所以坎格鲁斯普雷与爪牙们的决斗过程如下：

坎格鲁斯普雷会不断地从所有它评估的实力小于等于 c 且还没有跟他决斗过的爪牙中随机挑选一名爪牙进行决斗，直到坎格鲁斯普雷的获胜的决斗场数达到了 m ，输掉了某场决斗或者目前没有满足以上条件的爪牙能和坎格鲁斯普雷进行决斗，那么此时坎格鲁斯普雷就会化身怯战蜥蜴，停止决斗并落荒而逃。若坎格鲁斯普雷挑选的这名爪牙的实力小于等于 c ，那么坎格鲁斯普雷会在这场决斗中获胜；否则坎格鲁斯普雷会输掉这场决斗。

现在，作为坎格鲁斯普雷的死对头的袋鼠将军想问问你，坎格鲁斯普雷的期望胜利场数是多少，可以证明，答案总为一个有理数，你需要输出它对 998244353 取模后的结果。

Input

输入第一行一个整数 T ，表示测试数据组数。（ $1 \leq T \leq 100$ ）

对于每组测试数据，第一行四个整数 n, m, c, k 。（ $1 \leq n \leq 2 \times 10^5, 1 \leq m \leq 20, 1 \leq c, k \leq 10^8$ ）

第二行 n 个整数，第 i 个整数表示 a_i 。（ $1 \leq a_i \leq 10^8$ ）

数据保证 $\sum n \leq 10^6, \sum m \leq 100$ 。

Output

对于每组测试数据，输出一行一个整数表示答案，不同测试数据的答案之间需换行。

Example

standard input	standard output
3	314262112
3 1 2 1	558102885
1 2 3	146453175
5 3 2139 200	
1743 2142 2127 1753 2005	
8 4 1600 600	
800 1200 1400 1600 1900 2100 2300 2400	

Note

请注意常数因子对程序效率的影响。

Problem G. 创作乐曲

Input file: standard input
Output file: standard output
Time limit: 8 seconds
Memory limit: 512 megabytes

众所周知，坎格鲁斯普雷喜欢创作乐曲。

根据《袋鼠韵律法则》，一首乐曲应由一定量的音符组成，且音符的音高应为 $1 \sim m$ 的整数；一首乐曲被认为是“美妙动听”的，当且仅当这首乐曲相邻的音符的音高差的绝对值不超过 k 。

现在，坎格鲁斯普雷创作了一首有 n 个音符的乐曲，其中第 i 个音符的音高为 a_i 。这时候，数据结构带师袋鼠将军出现了，它对你提出了 q 个询问，每个询问形如：对于第 l_i 个音符到第 r_i 个音符组成的子乐曲，至少删除多少个音符才能使这个子乐曲是“美妙动听”的。

虽然你很不情愿，但你还是接受了袋鼠将军的挑战，不然你就会被袋鼠将军当作怯战蜥蜴的。

Input

输入第一行一个整数 T ，表示测试数据组数。（ $1 \leq T \leq 10^3$ ）

对于每组测试数据，第一行三个整数 n, m, k 。（ $1 \leq n \leq 10^5, 1 \leq m, k \leq 10^{18}$ ）

第二行 n 个整数 a_i ，表示第 i 个音符的音高。（ $1 \leq a_i \leq m$ ）

第三行一个整数 q ，表示接下来有 q 个询问。（ $1 \leq q \leq 500$ ）

接下来的 q 行，每行两个整数 l_i, r_i ，表示一次询问。（ $1 \leq l_i \leq r_i \leq n$ ）

数据保证 $\sum n \leq 4 \times 10^5, \sum q \leq 2 \times 10^3$ 。

Output

对于每组测试数据的每个询问，输出一行一个整数，表示答案。每个询问的答案之间需换行，不同测试数据之间的答案也需换行。

Example

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

Problem H. 循环图

Input file: standard input
Output file: standard output
Time limit: 15 seconds
Memory limit: 512 megabytes

坎格鲁斯普雷被困在了一张循环图里，这张循环图有无数个节点，初始时坎格鲁斯普雷在 1 号节点。
循环图是边存在着一定循环关系的图，循环图里面的边可以用循环周期 n 和 m 对三元组 (u_i, v_i, w_i) ($1 \leq u_i \leq n, u_i + 1 \leq v_i \leq 2 \times n, 1 \leq w_i \leq 10^9$) 表示。每对三元组 (u_i, v_i, w_i) 表示，对于循环图内所有的满足 $s = u_i + k \times n$, $t = v_i + k \times n$ ($k \in N$) 的点对 (s, t) , 都存在有 w_i 条从点 s 通往点 t 的边。
现在，坎格鲁斯普雷知道了这张循环图的第 L 个节点到第 R 个节点各存在着一个出口，坎格鲁斯普雷需要到达这些节点中的任意一个才能逃出循环图（到达有出口存在的节点后不一定要立刻逃出）。坎格鲁斯普雷想请你帮他算算，他有多少种逃出这张循环图的方式。由于答案可能很大，你需要输出答案对 $10^9 + 7$ 取模后的结果。

Input

输入第一行一个整数 T , 表示测试数据组数。 ($1 \leq T \leq 10$)
对于每组测试数据，第一行四个整数 n , m , L , R 。 ($1 \leq n \leq 100, m \geq 1, 1 \leq L \leq R \leq 10^{18}$)
接下来 m 行，每行三个整数 u_i , v_i , w_i 。 ($1 \leq u_i \leq n, u_i + 1 \leq v_i \leq 2 \times n, 1 \leq w_i \leq 10^9$)
数据保证在同一组测试数据中，若 $i \neq j$, 那么 $u_i = u_j$ 和 $v_i = v_j$ 不同时成立。

Output

对于每组测试数据，输出一行一个整数表示答案。每两组测试数据之间的答案需换行。

Example

standard input	standard output
2 3 4 5 6 1 2 1 1 3 1 3 4 1 2 5 1 5 8 998244353 1000000007 1 2 114514 1 4 1919810 2 3 999999999 3 5 111111111 4 5 1000000000 1 10 123456789 5 6 987654321 3 9 888888888	3 18719743

Note

对于样例 #1:
在第一组测试数据中，坎格鲁斯普雷的 3 条逃跑路线分别为：
 $1 \rightarrow 2 \rightarrow 5$ 、 $1 \rightarrow 3 \rightarrow 4 \rightarrow 5$ 、 $1 \rightarrow 3 \rightarrow 4 \rightarrow 6$ 。
请注意常数因子对程序效率的影响。

Problem I. Hold'em Shark

Input file: standard input
 Output file: standard output
 Time limit: 15 seconds
 Memory limit: 256 megabytes

小Y 养了一群可爱的 Capoo，这些 Capoo 曾经热衷于给算法竞赛出题，然而，它们有一天突然发现，它们需要在两个月内以喵均 6.25 题的 KPI 出完 25 道题。于是它们决定让自己休息一会，先开始一轮 4 只 Capoo 的扑克游戏。

这种扑克游戏以德州扑克为原型，但请仔细阅读下方的题目描述并留意它们与通常规则的区别

扑克牌的点数有 13 种大小，从大到小分别为 A, K, Q, J, T, 9, 8, 7, 6, 5, 4, 3 和 2。其中 A, K, Q, J, T 依次对应 14, 13, 12, 11, 10。此外，扑克牌还有 4 种花色，分别为 红心 (Heart)，方块 (Diamond)，梅花 (Club) 和黑桃 (Spade)，这四种花色我们利用首字母，分别记为 H, D, C, S。因此，我们在一局游戏中，一共使用 $13 \times 4 = 52$ 张牌。

一手牌是 5 张牌所构成的集合。每手牌都有一个牌型，而每种牌型也具有一个大小。对于两手牌，如果它们的牌型不同，那么**牌型较大**的一方始终更大。如果两手牌的牌型相同，为了方便比较，我们将一手牌表达为序列 $(r_1, r_2, r_3, r_4, r_5)$ ，其中 r_i 为第 i 张牌的点数，一手的 5 张牌排列的顺序取决于具体牌型种的描述，随后，我们认为按上述规则表达为序列后**较大**的一方更大。对于两个序列，我们依次比较每一位的点数，直到某一位它们的点数不同，此时该位点数较大的序列较大，如果比较完最后一位之后仍然相同，则认为两个序列一样大。如果两手牌的牌型和序列均相同，我们认为它们是一样大的。

接下来我们从高到低给出 10 种牌型，如果一手牌满足多种牌型的模式，它属于大小较大的牌型。

- 皇家同花顺：由 T, J, Q, K, A 构成的同花顺
- 同花顺：手牌中包含五张同一花色且点数连续的牌，对应序列满足 $r_1 > r_2 > r_3 > r_4 > r_5$ ，请注意：同一花色的 A, 2, 3, 4, 5 构成同花顺，此时 A 的点数被认为比 2 小
- 四条：手牌中包含四张点数相同的牌，对应序列满足 $r_1 = r_2 = r_3 = r_4$
- 葫芦：手牌中包含三张点数相同的牌和另外两张点数相同的牌，对应序列满足 $r_1 = r_2 = r_3, r_4 = r_5$
- 同花：手牌中包含五张花色相同的牌，对应序列满足 $r_1 > r_2 > r_3 > r_4 > r_5$
- 顺子：手牌中包含五张点数连续的牌，对应序列满足 $r_1 > r_2 > r_3 > r_4 > r_5$ ，请注意：A, 2, 3, 4, 5 构成顺子，此时 A 的点数被认为比 2 小
- 三条：手牌中包含三张点数相同的牌，对应序列满足 $r_1 = r_2 = r_3, r_4 > r_5$
- 两对：手牌中包含两张点数相同的牌以及另外两张点数相同的牌，对应序列满足 $r_1 = r_2 > r_3 = r_4$
- 对子：手牌中包含两张点数相同的牌，对应序列满足 $r_1 = r_2, r_3 > r_4 > r_5$
- 散牌：不满足以上任一情况的一手牌，对应序列满足 $r_1 > r_2 > r_3 > r_4 > r_5$

在一轮游戏开始时，每只 Capoo 都会被发到 2 张私有牌。由于丰富的算法竞赛命题经验，Capoo 们在仰卧起坐中锻炼出了很高的心理承受能力，因此每只 Capoo 都会等到全部 5 张公共牌都被翻开后进行结算，结算时，每只 Capoo 的手牌大小为它的 2 张私有牌以及桌上的 5 张公共牌所能构成的最大的手牌的大小，手牌大小最大的 Capoo 均为本轮游戏的胜者。

但事情并没有这么简单。小Y 决定给 Capoo 们的游戏增加一些趣味性，他准备对 4 只 Capoo 的共计 8 张私有牌以及桌上的 3 张公共牌（即在翻牌圈 (Flop) 之后）进行至多 1 次换牌操作，即从这 11 张牌中选择两张牌，并交换它们。

而你的任务则是，根据当前的局势判断，他如何操作可以使得指定位置的 Capoo 胜率最高。

Input

第 1 行为 1 个整数 T ，满足 $1 \leq T \leq 5$ ，为数据的组数。

接下来是 T 组数据，对于每一组数据：

第 1 行为 3 个以空格分隔的字符串，每个字符串的长度均为 2，格式为 点数 + 花色，表示桌面上公共牌的大小及花色。其中，第 i 张牌的序号为 $i - 1$ 。例如，

AH KD 3C 代表桌面上的 3 张公共牌分别为 红心A，方块K 以及 梅花3，它们的序号依次为 0, 1, 2。

接下来 4 行，每行 2 个以空格分隔的字符串，每个字符串的长度均为 2，格式为 大小 + 花色，第 i 行的两个字符串代表序号为 $i - 2$ 的 Capoo 的两张私有牌的大小和花色。其中，第 i 张牌的序号为 $i - 1$ 。例如，如果第 3 行为 2S 3S，则说明序号为 1 的 Capoo 的两张私有牌为 黑桃2 和 黑桃3，它们的序号依次为 0, 1。

接下来 1 行，为一个值域是 $[0, 3]$ 的整数，表示 小Y 希望最大化胜率的 Capoo 编号。

Output

如果 小Y 使得指定位置的 Capoo 胜率最高不需要进行任何操作，输出一行一个字符串 **None**。

否则，输出一行 2 个以空格分隔的字符串，每个字符串的长度均为 2，格式为 点数 + 花色，描述需要交换的两张牌的点数和花色。请按照字典序升序排序输出需要交换的两张牌。

如果有多种换牌方案都能使得指定的 Capoo 胜率最高，那么优先选择不需要进行任何操作的方案，其次将每种方案按照字典序升序对需要交换的两张牌排序并拼接，取字典序最小的方案。

Example

standard input	standard output
1 8H QS 8C 4D 9D 2C AH QC QH 7S AD 3	7S QC

Problem J. 故障机器人想活下去

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 512 megabytes

故障机器人共有 x 点生命值，他在高塔中遭遇了 n 个敌人，在和第 i 个敌人战斗后故障机器人会损失 a_i 点生命值，在战斗结束后生命值降低到 0 或以下时故障机器人会死亡。同时，故障机器人手上有 k 个烟雾弹，在战斗开始时，他可以选择消耗 1 个烟雾弹来逃离这场战斗，如此战斗会直接结束，故障机器人在这场战斗中不会损失生命值。

故障机器人将依次与 n 个敌人进行战斗，他想知道在最优策略下，最多能存活到第几场战斗结束，你能帮帮故障机器人吗？

Input

一行一个正整数 $t(1 \leq t \leq 30)$ ，表示有 t 组数据。

每组数据的第一行有三个整数 $n, x, k(1 \leq n \leq 10^5, 1 \leq x \leq 10^9, 0 \leq k \leq 10^5)$ ，分别表示敌人的数量，故障机器人的初始生命值和烟雾弹的数量。

每组数据的第二行有 n 个正整数，其中第 i 个正整数 $a_i(1 \leq a_i \leq 10^5)$ 表示故障机器人与第 i 个敌人战斗后损失的生命值。

Output

对于每组数据输出一行

每行一个整数，表示故障机器人最多能存活到第几场战斗结束。如果故障机器人不能在任何一场战斗结束时存活，输出 0。

Example

standard input	standard output
3	4
5 10 1	2
3 4 5 2 7	8
3 7 0	
2 3 4	
10 20 3	
6 12 9 4 10 1 3 4 2 1	

Note

对第一组样例的解释：故障机器人共有 10 点生命值与 1 个烟雾弹，在与第 1 个敌人战斗后损失了 3 点生命值，在与第 2 个敌人战斗后损失了 4 点生命值，使用烟雾弹结束与第 3 个敌人的战斗，在与第 4 个敌人战斗后损失了 2 点生命值，在与第 5 个敌人的战斗中故障机器人的生命值不足 7 点，死亡，因此故障机器人最多能存活到第 4 场战斗结束，可以证明不存在比 4 更大的答案。

Problem K. 关于 agKc 实在不喜欢自动化于是啥都自己合成这件事

Input file: standard input
Output file: standard output
Time limit: 15 seconds
Memory limit: 512 megabytes

在游戏 minecraft 中，合成是一件非常重要的事，其基本内容是：由若干种不同的物品，合成新的物品。玩家可以通过实现流水线自动化来大幅缩减自己的工作量，但 agKc 实在不喜欢这样玩，于是选择全靠自己合成。

现一共有 n 种物品，依次编号为 $1 \sim n$ 。为了更好的描述问题，我们将物品分为两类：

1. 原材料：可以直接获取的物品，获取 1 个第 i 种原材料需要花费 t_i 时间。
2. 合成产物：通过合成获得的物品，可以根据其合成配方，通过消耗若干个其它物品合成。合成不需要花费时间。

具体而言，若第 b 种物品为一合成产物，则其合成配方可表示为 $x_1a_1 + x_2a_2 + \cdots + x_ka_k \rightarrow b$ ，即，可以消耗 x_1 个第 a_1 种物品， x_2 个第 a_2 种物品， \dots ，以及 x_k 个第 a_k 种物品，来合成出一个第 b 种物品。

保证一种物品只会作为一个配方的原料，以及一种配方的产物。

以合成一个铁砧举例，所需要的配方如下：

- $1 \times \text{粗铁矿} \rightarrow 1 \times \text{铁锭}$
- $9 \times \text{铁锭} \rightarrow 1 \times \text{铁块}$
- $3 \times \text{铁块} + 4 \times \text{铁锭} \rightarrow 1 \times \text{铁砧}$

其中粗铁矿为原材料，铁锭，铁块和铁砧为合成产物。

现在 agKc 打算爆肝合成一件物品，并告诉你所有需要的配方，同时 agKc 的爆肝行为感动了服务器的管理员，管理员决定送给他除了最终产物外的无限的任意 1 种物品，请你计算出，agKc 需求哪一种物品，能使合成出最终产物的耗时最短，并回答出最短耗时。

最终耗时为生产所需原材料的耗时之总和，合成过程不需要花费时间。保证一定可以在有限的时间内生产出最终产物。但由于 agKc 的时间有限，若最短耗时超过了 10^9 ，请输出 ‘Impossible’。

Input

第一行一个整数 t ($1 \leq t \leq 20$)

第一行两个整数 n ($1 \leq n \leq 10^5$) 代表物品的总数量， k ($1 \leq k \leq n$) 代表 agKc 需要合成的物品 接下来 n 行,第 i 行给出第 i 个物品的合成配方，格式如下：首先一个正整数 p ($0 \leq p \leq 1$) 代表物品的种类

- $p = 0$ 代表该物品为原材料，则后面一个正整数 t ($1 \leq t \leq 10^5$) 代表获取一个该物品需要花费的时间。
- $p = 1$ 代表该物品为合成产物，则后面一个正整数 t ($1 \leq t \leq n$) 代表该配方中原料的数量，接着给出 t 组 (x_i, a_i) 数对,代表第 i 个原料的数量与种类。

Output

共 t 行 每行一个整数，表示 agKc 合成这件物品的最短耗时，同时若最短耗时大于 10^9 ，请输出 ‘Impossible’。

Example

standard input	standard output
3	6
5 1	0
1 2 3 2 2 5	13
1 2 2 3 1 4	
0 1	
0 2	
0 3	
5 4	
1 1 2 5	
1 1 2 4	
0 2	
1 1 1 3	
1 1 2 2	
5 5	
1 1 5 4	
0 3	
0 4	
0 3	
1 3 3 2 1 3 5 1	

Problem L. 蛋糕上的草莓是蛋糕的灵魂

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes

草莓蛋糕的灵魂是蛋糕上的草莓。

现在有 x 块大小相同的草莓和 y 块大小相同的蛋糕。Mobiusp 可以选择把所有草莓堆放在一起并且切 $n(n > 0)$ 刀，将每个草莓切成大小相同的 $2n$ 块，总共得到 $2nx$ 块大小相同的草莓，每块草莓的大小是原来的 $\frac{1}{2n}$ 。当然，也可以一刀也不切，保留原来的 x 块草莓。

同理，可以选择把所有蛋糕堆放在一起切 m 刀，得到大小相同的 $2my$ 块蛋糕，或者一刀也不切。

现在，Mobiusp 需要为蛋糕分配草莓，并满足以下条件：

- 每块蛋糕上至少有一块草莓
- 每块蛋糕上的草莓块数相同

请你设计一种切开草莓和蛋糕，并将草莓分配给每块蛋糕的方案，**优先**保证每个草莓的大小最大，再保证每个蛋糕的大小最大，输出最终蛋糕的个数与每个蛋糕上的草莓的个数。草莓和蛋糕必须用尽，不能丢弃。

Input

第一行一个整数 t ，表示有 t 组数据 ($1 \leq t \leq 10^5$)

接下来 t 行，每行两个整数 x 和 y ($1 \leq x, y \leq 10^{12}$)，分别表示草莓的个数与蛋糕的个数

Output

对于每组数据输出一行

每行两个整数用空格隔开，分别表示蛋糕的个数与每个蛋糕上草莓的个数

Example

standard input	standard output
2	8 1
2 8	5 2
10 5	