

第二届环球杯



第 12 赛段：秦皇岛

2023 年 10 月 12-13 日

这套问题集应包含 13 个问题，共 20 页（有编号）。

根据



中国大学生程序设计竞赛（CCPC）

主办



筹备





问题 A. 气球机器人

时间限制 1 秒 内存限
制 64 兆字节

2017年中国大学生程序设计竞赛秦皇岛赛点来了！本次比赛将有 n 支队伍参赛，比赛将在一张巨大的圆桌上举行，圆桌周围有 m 个座位，从 1 到 m 顺时针依次排列。第 i 个参赛队将坐在第 s_i - 个座位上。

竞技编程爱好者 "包包" 在比赛前对比赛结果进行了 p 次预测。每个预测的形式都是 (a_i, b_i) ，这意味着 a_i -th 团队在 b_i -th 时间单位内解决了一个问题。

我们知道，当一个团队解决了一个问题，就会奖励该团队一个气球。如果气球几乎要花上几个世纪才能送到，参与者就会不高兴。如果一个团队在第 t_a -th 个时间单位内解决了一个问题，而气球是在第 t_b -th 个时间单位内送到他们手中的，那么该团队的不开心程度将增加 $(t_b - t_a)$ 。为了及时发放气球，比赛组织者购买了一个气球机器人。

比赛开始时（即第 1 个时间单位开始时），机器人将被放在第 k 个座位上，并开始桌面上移动。如果机器人走过的队伍在机器人最后一次走过之后为自己赢得了一些气球，那么机器人就会把自己应得的气球全部送给该队伍。在每个单位时间内，将依次发生以下事件：

1. 机器人移动到下一个座位也就是说，如果机器人目前在第 i 个 $(1 \leq i < m)$ 座位上，它将移动到第 $(i + 1)$ 个座位上；如果机器人目前在第 m 个座位上，它将移动到第 1 个座位上。
2. 参与者根据包包的预测解决一些问题。
3. 如有需要，机器人会向坐在其当前位置上的团队发放气球。

包包想让所有团队的总不快乐值最小。你的任务是选择机器人的起始位置 k ，并根据 "包包" 的预测计算出所有团队的总不愉快最小值。

输入

有多个测试用例。输入的第一行包含一个整数 T ，表示测试用例的数量。对于每个测试用例

第一行包含三个整数 n 、 m 和 p $(1 \leq n \leq 10^5, n \leq m \leq 10^9, 1 \leq p \leq 10^5)$ ，表示参赛队数、席位数和预测数。

第二行包含 n 个整数 s_1, s_2, \dots, s_n $(1 \leq s_i \leq m, \text{且对于所有 } i \neq j, s_i \neq s_j)$ ，表示每个小组的席位编号。



下面的 p 行分别包含两个整数 a_i 和 b_i ($1 \leq a_i \leq n$, $1 \leq b_i \leq 10^9$)，表明 a_i -th team solves a problem at time b_i 根据 BaoBao 的预测。

保证所有测试用例的 n 和 p 之和都不超过 5×10^5 。

输出

每个测试用例输出一个整数，表示根据包包的预测，所有团队的总不开心程度最小。



示例

标准输入	标准输出
4	1
2 3 3	4
1 2	5
1 1	50
2 1	
1 4	
2 3 5	
1 2	
1 1	
2 1	
1 2	
1 3	
1 4	
3 7 5	
3 5 7	
1 5	
2 1	
3 3	
1 5	
2 5	
2 100 2	
1 51	
1 500	
2 1000	

备注

对于第一个样本测试案例，如果我们选择的起始位置是第 1 个座位，则总的不快乐次数为 $(3 - 1) + (1 - 1) + (6 - 4) = 4$ 。如果我们选择第 2 个座位，则总的不快乐次数为 $(2 - 1) + (3 - 1) + (5 - 4) = 4$ 。如果我们选择第 3 个座位，则不开心的总次数为 $(1 - 1) + (2 - 1) + (4 - 4) = 1$ 。所以答案是 1。

在第二个抽样测试案例中，如果我们选择的起始位置是第 1 个座位，那么总的不开心次数为 $(3 - 1) + (1 - 1) + (3 - 2) + (3 - 3) + (6 - 4) = 5$ 。如果我们选择第 2 个座位，则总的不开心次数为 $(2 - 1) + (3 - 1) + (2 - 2) + (5 - 3) + (5 - 4) = 6$ 。如果我们选择第 3 个座位，则总的不开心次数为 $(1 - 1) + (2 - 1) + (4 - 2) + (4 - 3) + (4 - 4) = 4$ 。所以答案是 4。



问题 B. 预计等待时间

时间限制 1 秒 内存限制
512 兆字节

DreamGrid 是一位著名的竞技编程教练。他为人和善，很多选手都愿意向他请教。

DreamGrid 每天都会在办公室里遇到 n 个竞争对手，因此会发生 n 个 "到达" 事件和 n 个 "选择" 事件。在 t 时刻发生的到达事件表示在 t 时刻有一位竞争者到达办公室的候见室，而在 t 时刻发生的选择事件表示 DreamGrid 从候见室随机选择（概率相等）一位竞争者与之交谈。当然，如果选择事件发生，等候室一定不是空的。谈话结束后，竞争者离开办公室，再也不会回来。

几天后，DreamGrid 开始对每位选手在所有有效情况下的总预期等待时间的平均值感到好奇。选手的等待时间是他被 DreamGrid 选中的时间减去他到达候选室的时间。一个案例是一个长度为 $2n$ 的序列，由 n 个到达事件和 n 个选中事件组成，其中第 i 个事件将在时间 a_i 发生。对于一个有效案例，必须满足当选中事件发生时，候选室必须不是空的。

例如，我们把到达事件记为 "A"，把选择事件记为 "S"。如果 $n = 2$ ， $a_1 = 1$ ， $a_2 = 2$ ， $a_3 = 3$ ， $a_4 = 4$ ，那么序列 "AASS" 有效，但序列 "ASSA" 无效，因为在 $a_3 = 3$ 时刻发生的 "选择" 事件无效。

由于答案可能不是整数，您应该计算 $ab^{-1} \bmod p$ ，其中 a （ a 和 b 是同素）是答案， $p > 2n$ ，且 p 是素数，而 b^{-1} 是 b 关于模 p 的模乘逆。很容易证明， b 的素因子永远不会大于 $2n$ 。

输入

有多个测试用例。第一行输入包含一个整数 T ($1 \leq T \leq 10^4$)，表示测试用例的数量。对于每个测试用例

第一行包含五个整数 n, p, b_0, A, B ($1 \leq n \leq 10^6$, $2n < p \leq 2 \times 10^9$, $0 \leq b_0, A, B < p$)，其中 p 是质数。 n 和 p 的含义如上所述。其余为 a 的生成器，其中 $a_0 = 0$ ， $a_i = a_{i-1} + b_i + 1$ ， $b_i = (A \cdot b_{i-1} + B) \bmod p$ ，对于所有 $1 \leq i \leq 2n$ 。

保证所有测试用例中 n 的总和不超过 10^7 。

输出

对于每个测试用例，用一行输出一个表示答案的整数。

示例

标准输入	标准输出
5	1



1 1000000007 0 1 0	12
2 1000000007 0 1 1	1
2 7 5 2 3	21
3 31 15 6 24	879705565
20 1000000007 0 1 0	

备注

我们把到达事件记为 "A"，把选择事件记为 "S"。

在第一个测试案例中， $s_1 = 1$ ， $s_2 = 2$ ，只有一个有效序列 "AS"。唯一竞争者的等待时间为 $2 - s_1 = 1$ ，因此答案为 1，我们需要输出 $1 \bmod 1000000007 = 1$ 。



在第二个测试案例中，我们有 $t_1 = 2$ 、 $t_2 = 5$ 、 $t_3 = 9$ 和 $t_4 = 14$ 。有两个有效序列 "ASAS "和 "AASS"。
。

对于第一个序列，第一个到达的竞争者的预期等待时间是 $5 - 2 = 3$ ，第二个到达的竞争者的预期等待时间是 $14 - 9 = 5$ 。

对于第二个序列，第一个到达的竞争者的预期等待时间是 $((9 - 2) + (14 - 2))/2 = 9.5$ ，第二个到达的竞争者的预期等待时间是 $((9 - 5) + (14 - 5))/2 = 6.5$ 。因此，答案是 $((3 + 5) + (9.5 + 6.5))/2 = 12$ ，我们需要输出 $12 \bmod 1000000007 = 12$ 。

在第三个 *测试用例*中， $t_1 = 7$ ， $t_2 = 9$ ， $t_3 = 15$ ， $t_4 = 22$ 。与第二个测试用例的分析一样，序列 "ASAS "的总预期等待时间为 $(9 - 7) + (22 - 15) = 9$ ，序列 "AASS "的总预期等待时间为 $((15 - 7) + (22 - 7))/2 + ((15 - 9) + (22 - 9))/2 = 21$ 。因此答案为 $(9 + 21)/2 = 15$ ，我们需要输出 $15 \bmod 7 = 1$ 。



问题 c. 十字军任务

时间限制 1 秒 内存限
制 64 兆字节

《十字军任务》是一款有趣的手机游戏。一个神秘的女巫给游戏世界带来了巨大的黑暗，你的王国唯一的希望就是拯救女神们，让她们释放出自己的力量来对抗女巫。



为了拯救游戏世界，您需要选择三名英雄为胜利而战，并明智地使用他们的技能。屏幕下方会出现三种不同类型的九个技能方块（每种类型三个方块）。如果连续出现 k 个 ($k \geq 1$) 相同类型的图块，您就可以点击它们并消除它们，从而触发它们所代表的强大技能。消除后，左边的方块会与右边的方块相连。此外，如果连续消除 $k = 3$ 个相同类型的方块，它们释放的强大技能将升级为超级技能，也就是最强大的技能。

DreamGrid 是游戏新手，他想尽可能多地触发超级技能。给定 9 个符合上述描述的技能块，请帮助 DreamGrid 计算他能触发超级技能的最大次数。

输入

有多个测试用例。第一行输入包含一个整数 T (约 50)，表示测试用例的数量。对于每个测试用例

第一行包含一个字符串 s ($|s| = 9$)，由三个 "g"、三个 "a" 和三个 "o" 组成，代表三种不同类型的九个技能块。每种类型的字符代表一种类型的技能块。

输出



对于每个测试用例，输出一个整数，表示 DreamGrid 能触发超级技能的最大次数。



示例

标准输入	标准输出
7	3
gggaaaooo	3
aaoggogoa	2
googgaaa	1
aggaoaag	3
googgaaa	2
gogogoaaa	1
gaogaogao	

备注

在第一个测试案例中，DreamGrid 可以首先消除 "aaa"（触发一个超级技能），从而将技能块变为 "ggooo"。然后，他可以消除 "ggg"（触发另一个超级技能），最后消除 "ooo"（触发第三个超级技能）。所以答案是 3。

在第二个测试案例中，DreamGrid 可以首先消除 "ggg"（触发一个超级技能），从而将技能块变为 "aaooooa"。然后，他可以消除 "ooo"（触发另一个超级技能），最后消除 "aaa"（触发第三个超级技能）。因此答案也是 3。

在第三个测试案例中，DreamGrid 可以首先消除 "aaa"（触发了一个超级技能），从而将技能块变为 "googgo"。然后，他可以消除 "oo"，得到 "ggggo"，再消除 "gggg"（触发另一个超级技能），得到 "o"。因此，答案是 2。很容易证明，在这种技能块排列方式下，他无法三次触发超级技能。



问题 D. 图形生成器

时间限制 1 秒 内存限制 256 兆字节

DreamGrid 制作了一个图形生成器。生成器接收一个整数 n ，然后生成一个有 n 个顶点的无向图。
更具体地说，生成器将生成一个空图 G 和 $\{1, 2, \dots, n\}$ 之后，生成器将进行 n 次以下操作，并在第 q 次操作期间进行以下操作：

- 1. 找出 G 中所有相连的分量 C_1, C_2, \dots, C_m 在 G 中。
- 2. 选择一个子集 $S_q = \bigcup_{i=1}^m C_i$ ，使得 S_q 中没有两个顶点属于同一个连通部分。
- 3. 在 G 中创建一个索引为 p_q 的新顶点，并在 p_q 和 C_{bel_i} 中的每个顶点 v 之间添加一条边、
其中， bel_i 是 i 所属连通组件的索引。

鉴于最终生成的图形，DreamGrid 希望知道所有 p_q 和 S_{o_q}

输入

有多个测试用例。第一行输入包含一个整数 T ，表示测试用例的数量。对于每个测试用例
第一行包含两个整数 n 和 m ($1 \leq n \leq 10^5, 0 \leq m \leq \min(10^5, \frac{n(n-1)}{2})$) - 的数量
顶点和边的数量。
接下来的 m 行中，每一行都包含两个整数 u_i 和 v_i ($1 \leq u_i, v_i \leq n$)。不会出现自循环或多条边。
保证所有 n 的总和和所有 m 的总和都不超过 2×10^6 。

输出

对于每个测试用例，如果生成器无法生成图形，则在第一行输出 "否"。否则，在第一行输出 "是"。然后在下面第 i 行输出两个整数 q_i 和 s_i ($1 \leq q_i \leq n$)，接着输出 s_i 整数： a_1, a_2, \dots, a_{s_i} ($1 \leq a_j \leq n$) - 第 i 次操作中新建顶点和子集的索引。如果有多个解决方案，则打印其中任何一个。

示例

标准输入	标准输出
3	是
3 0	1 0
4 4	2 0
1 2	3 0
2 3	是
3 4	1 0



2 4	4 0
5 5	3 1 4
1 2	2 2 1 3
2 3	没有
3 4	
4 5	
2 4	



问题 E. CCPC 的字符串

时间限制 1 秒 内存限制
制 64 兆字节

小宝刚刚在口袋里发现了一个长度为 n 的字符串 s ，由 "C "和 "P "组成。作为中国大学生程序设计竞赛的忠实粉丝，包包认为，当且仅当 $s_i = s_{i+1} = s_{i+3} = \text{'C'}$ ，且 $s_{i+2} = \text{'P'}$ ，其中 s_i 表示字符串 s 中的第 i 个字符时， s 的子串 $s s_{ii+1i+2i+3}$ 是 "好 "的。 s 的值是 s 中不同 "好 "子串的个数。两个 "好 "子串 $s s_{ii+1i+2i+3}$ 和 $s s_{jj+1j+2j+3}$ 是不同的，当且仅当 $i \neq j$ 时。

为了让这个字符串更有价值，小宝决定从字符商店购买一些字符。每次他都可以从商店里买一个 "C "或一个 "P"，并把字符插入 s 的任意位置。如果小宝第 i 次购买一个字符，他需要花费 $(i - 1)$ 个单位的价值。

包包获得的最终价值是 s 的最终价值减去从商店购买的所有字符的总成本。请帮助包包最大化最终价值。

输入

有多个测试用例。输入的第一行包含一个整数 T ，表示测试用例的数量。对于每个测试用例

第一行包含整数 n ($1 \leq n \leq 2 \times 10^5$)，表示字符串 s 的长度。第二行包含由 "C "和 "P "组成的字符串 s ($|s| = n$)。

保证所有测试用例的 n 之和不超过 10^6 。

输出

每个测试用例输出一行，其中包含一个整数，表示 BaoBao 可以获得的最大最终值。

示例

标准输入	标准输出
3	1
3	1
CCC	1
5	
CCCCP	
4	
CPCP	

备注

在第一个示例测试用例中，余额宝可以购买一个 "P"（成本值为 0），并将 s 更改为 "CCPC"。因此



，最终值为 $1 - 0 = 1$ 。

在第二个示例测试用例中，包包可以购买一个 "C "和一个 "P"（成本 $0 + 1 = 1$ 值），并更改的值为 "CCPCCPC"。因此，最终值为 $2 - 1 = 1$ 。

在第三个示例测试用例中，余额宝可以购买一个 "C"（成本值为 0），并将 s 更改为 "CCPCP"。因此，最终值为 $1 - 0 = 1$ 。

很容易证明，在样本测试用例中，任何购买和插入字符的策略都无法取得更好的结果。



问题 F. 迷路

时间限制 1 秒 内存限制
512 兆字节

小宝在一个无限大的二维平面上迷路了！从 (a_x, a_y) 出发，小宝必须找到位于 (b_x, b_y) 的家。但这对他来说并不容易，因为平面上散布着 n 个圆形障碍物。第 i 个障碍物的半径为 r_i ，其中心位于 (x_i, y_i) 处。小宝可以在平面上自由移动，但他的路线不能与任何障碍物相交（但可以与任何障碍物相切）。

当且仅当小宝的最终位置与 (b_x, b_y) 之间的距离不大于 R ，且连接小宝最终位置与 (b_x, b_y) 的线段不与任何障碍物相交（但可以与任何障碍物相切）时，我们说小宝成功到达了他家。

小宝急着回家，你的任务是帮助小宝找到从以下地点出发的最短路线 (a_x, a_y) 并导致他成功抵达自己的家。

输入

有多个测试用例。输入的第一行包含一个整数 T （约 60），表示测试用例的数量。对于每个测试用例

第一行包含一个整数 n ($0 \leq n \leq 2$)，表示平面上的障碍物数量。

第二行包含两个整数 a_x 和 a_y ($-10^4 \leq a_x, a_y \leq 10^4$)，表示宝博的起点。

第三行包含三个整数 b_x, b_y ($-10^4 \leq b_x, b_y \leq 10^4$) 和 R ($0 \leq R \leq 10^4$)，表示描述中提到的包包家的位置和距离。

下面的 n 行分别包含两个整数 x_i, y_i ($-10^4 \leq x_i, y_i \leq 10^4$) 和 r_i ($0 < r_i \leq 10^4$)，表示障碍物的中心和半径。

保证 (a_x, a_y) 和 (b_x, b_y) 都在障碍物的外部或边界上，且障碍物之间互不包含、互不相交或相切。

输出

为每个测试用例输出一行，指出小宝到达家的最短距离。当且仅当您的答案的绝对误差或相对误差小于 10^{-6} 时，您的答案才会被认为是正确的。

示例

标准输入	标准输出
2	8.209191463668802
1	5.000000000000000
10 0	



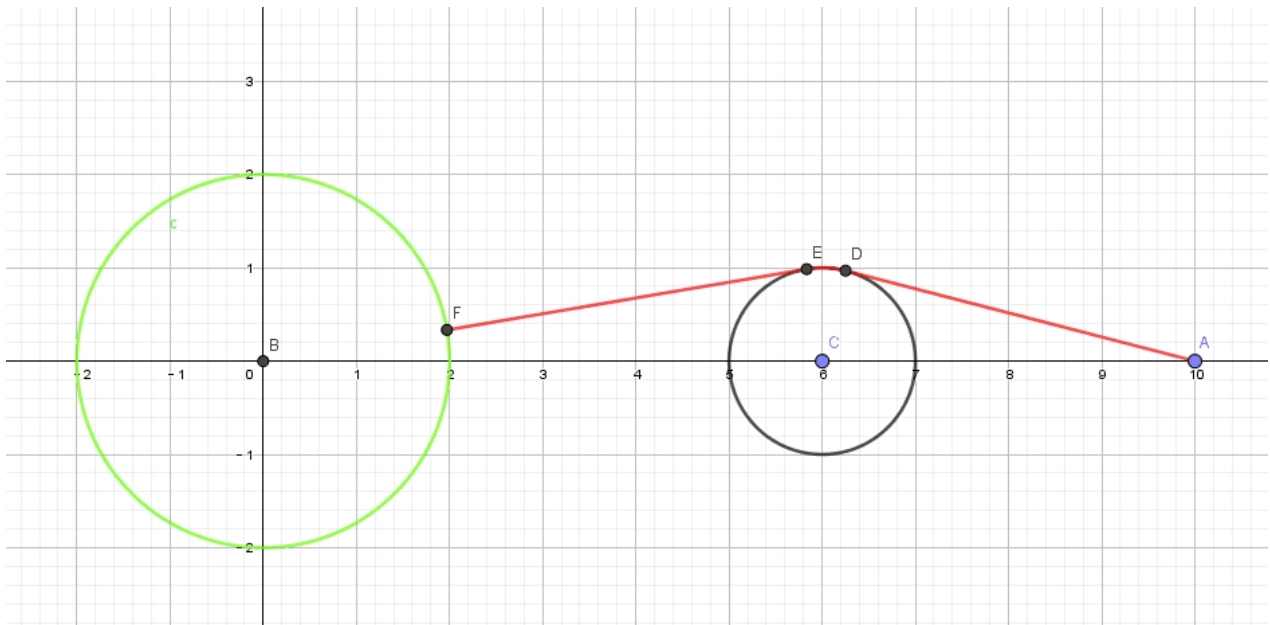
第三届环球杯

第 12 站：秦皇岛，2024 年 10 月 12-13 日

0 0 2	
6 0 1	
0	
0 10	
0 0 5	



备注



上图展示了第一个示例测试用例，其中

$$|ad| = 3.8729833462 \quad |\overline{de}| = 0.4201283344 \quad |ef| = 3.9160797831$$

所以答案是 $|AD| + |\overline{DE}| + |EF| = 8.2091914637$ 。



问题 G. 数字

时间限制 1 秒 内存限制
256 兆字节

DreamGrid 有一个非负整数 n 。他想把 n 分成 m 个非负整数 a_1, a_2, \dots ，并把它们的位或最小化（即 $n = a_1 + a_2 + \dots + a_m$ 和 $a_1 \text{ OR } a_2 \text{ OR } \dots \text{ OR } a_m$ ，并将它们的位或最小化（即 $n = a_1 + a_2 + \dots + a_m$ 和 $a_1 \text{ OR } a_2 \text{ OR } \dots \text{ OR } a_m$ 应尽可能小）。

输入

有多个测试用例。第一行输入包含一个整数 T ，表示测试用例的数量。对于每个测试用例第一行包含两个整数 n 和 m ($0 \leq n < 10^{1000}$, $1 \leq m < 10^{100}$)。保证 n 的长度之和不超过 2×10^4 。

输出

对于每个测试用例，输出一个整数，表示其位或的最小值。

示例

标准输入	标准输出
5	3
3 1	3
3 2	1
3 3	2000
10000 5	125
1244 10	



问题 H. 质集

时间限制 1 秒 内存限
制 64 兆字节

给定一个由 n 个整数组成的数组 a_1, a_2, \dots, a_n ，如果 $i \neq j$ ，我们就说集合 $\{i, j\}$ 是给定数组的质数集。

而一个 $i +$ 一个 j 是质数。

小宝刚刚在口袋里找到了一个由 n 个整数组成的数组 a_1, a_2, \dots, a_n 。他想最多选择 k 该数组的质集，以最大化所选集合的大小。也就是说，最大化

$\sum_{i=1}^m p_i$ 通过仔细选择 m 和 p_1, p_2, \dots, p_m ，其中 $m \leq k$ 且 p_i 是给定数组的素数集。

请帮助 BaoBao 计算联合集的最大大小。

输入

有多个测试用例。输入的第一行是整数 T ，表示测试用例的数量。对于每个测试用例

第一行包含两个整数 n 和 k ($1 \leq n \leq 3 \times 10^3, 0 \leq k \leq \frac{n(n-1)}{2}$)，它们的含义是如上所述。

第二行包含 n 个整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^6$)，表示给定的数组。保证所

有测试用例的 n 之和不超过 10^4 。

输出

对每个测试用例输出一行，其中包含一个整数，表示给定数组中最多 k 个素数集的最大联合大小。

示例

标准输入	标准输出
4	4
4 2	3
2 3 4 5	6
5 3	0
3 4 12 3 6	
6 3	
1 3 6 8 1 1	
1 0	
1	

备注

第一个样本测试用例有 3 个质数集： $\{1, 2\}$ 、 $\{1, 4\}$ 和 $\{2, 3\}$ 。由于 $k = 2$ ，我们可以选择 $\{1, 4\}$ 和 $\{2, 3\}$ ，得到最大的联合集 $\{1, 2, 3, 4\}$ ，大小为 4。



对于第二个样本测试用例，只有两个质集： $\{1, 2\}$ 和 $\{2, 4\}$ 。由于 $k = 3$ ，我们可以选择这两个集合，从而得到大小为 3 的最大联合集 $\{1, 2, 4\}$ 。

第三个样本测试用例有 7 个质集： $\{1, 3\}$ ， $\{1, 5\}$ ， $\{1, 6\}$ ， $\{2, 4\}$ ， $\{3, 5\}$ ， $\{3, 6\}$ 和 $\{5, 6\}$ 。由于 $k = 3$ ，我们可以选择 $\{1, 3\}$ 、 $\{2, 4\}$ 和 $\{5, 6\}$ ，从而得到最大的联合集 $\{1, 2, 3, 4, 5, 6\}$ ，其大小为 6。



问题 1. 三角测量

时间限制

1 秒 内存限制

256 兆字节

DreamGrid 有一个由 n 个点组成的点集 P 。点的标号从 1 到 n 。

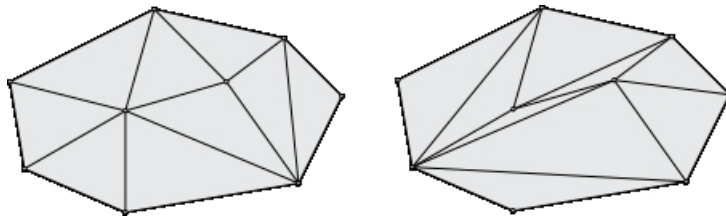
他想要在几对点之间绘制一些线段，使最终结果形成一个三角形。在点 u 和点 v 之间绘制线段的成本为 $w_{u,v}$ 。

DreamGrid 希望知道总成本的最小值，以及能达到最小总成本的三角剖分的数量。

点集 P 的三角剖分是三角形集合 T ，使得

1. $\text{conv}(P) = \bigcup_{T \in T} T$ ，其中 $\text{conv}(P)$ 是 P 的凸壳。
2. $P = \bigcup_{T \in T} V(T)$ ，其中 $V(T)$ 是三角形 T 的三个顶点的集合。
3. 对于每一对不同的 $T, U \in T$ ， $T \cap U$ 要么是一个公共顶点，要么是一条公共边，要么是空的。

例如，下面是同一组 9 个点的两种不同的三角剖分。



来自维基百科。 https://en.wikipedia.org/wiki/Point_set_triangulation

输入

有多个测试用例。第一行输入包含一个整数 T （约 70），表示测试用例的数量。对于每个测试用例

第一行包含一个整数 n ($3 \leq n \leq 18$) -- 点数。

接下来的 n 行中，每一行都包含两个整数 x_i 和 y_i ($0 \leq x_i, y_i \leq 10^6$)，表示第 i 个点的坐标。没有三个点位于同一条直线上。

接下来 n 行中的第 i 行包含 n 个整数 $w_{i,1}, w_{i,2}, \dots, w_{i,n}$ ($0 \leq w_{i,j} \leq 10^6, w_{i,i} = 0, w_{i,j} = w_{j,i}$)，表示绘制线段的成本。

输出

对于每个测试用例，输出两个整数，分别表示最小成本和三角剖分的次数。



示例

标准输入	标准输出
2	5 2
4	6 1
0 0	
1 1	
1 0	
0 1	
0 1 1 1	
1 0 1 1	
1 1 0 1	
1 1 1 0	
4	
0 0	
3 0	
1 3	
1 1	
0 1 1 1	
1 0 1 1	
1 1 0 1	
1 1 1 0	



问题 J. 树方程

时间限制

1 秒 内存限制

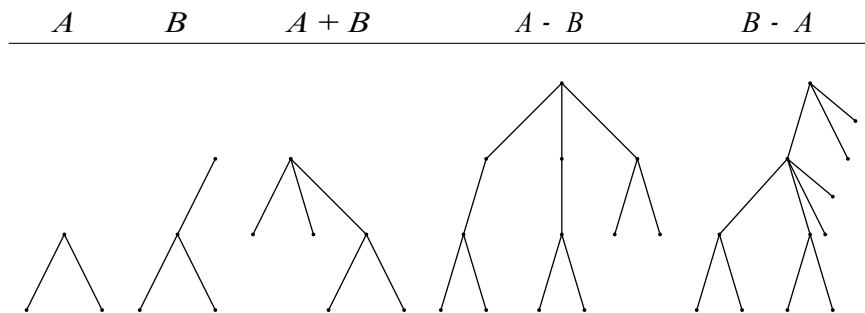
256 兆字节

厌倦了求解数学公式，DreamGrid 开始求解与有根树有关的方程。

假设 A 和 B 是两棵任意有根的树， $r(T)$ 表示 T 的根。DreamGrid 定义了两种基本操作：

- **加法。** $T = A + B$ 是通过将两个根 $r(A)$ 、 $r(B)$ 合并成一个新根 $r(T)$ 而建立的。也就是说， A 和 B 的子树（如果有的话）变成了 $r(T)$ 的子树。
- **乘法。** $T = A \cdot B$ 是通过将 $r(B)$ 与每个顶点 $x \in A$ 合并而建立的，这样 $r(B)$ 的所有子树都变成了 x 的新子树。

下图可能有助于您理解这些操作。



给定三棵有根树 A 、 B 和 C ，DreamGrid 希望找到两棵有根树 X 和 Y ，使得

$$a \cdot x + b \cdot y = c。$$

输入

有多个测试用例。第一行输入包含一个整数 T ，表示测试用例的数量。对于每个测试用例

第一行包含三个整数 n_a 、 n_b 和 n_c ($2 \leq n_a, n_b \leq n_c \leq 10^5$) --分别是有根树 A 、 B 和 C 的顶点数。

第二行包含 n 个整数 a_1, a_2, \dots, a_{n_a} ($0 \leq a_i < i$) --其中， a_i 是树 A 中第 i 个顶点的父节点。

第三行包含 n 个整数 b_1, b_2, \dots, b_{n_b} ($0 \leq b_i < i$) --其中 b_i 是树 B 中第 i 个顶点的父顶点。

第四行包含 n 个整数 c_1, c_2, \dots, c_{n_c} ($0 \leq c_i < i$) --其中 c_i 是树 C 中第 i 个顶点的父顶点。

请注意，如果 $a_i = 0$ ($b_i = 0$ 或 $c_i = 0$)，那么第 i 个顶点就是树 A (B 或 C) 的根。



保证所有 n 个的总和不超过 2×10^6 。

输出

对于每个测试用例，如果找不到解决方案，就在第一行输出 "不可能"（不带引号）。

否则，在第一行输出两个整数 n_x 和 n_y ($1 \leq n_x, n_y \leq 10^5$)，表示有根树 X 和 Y 的顶点数。

然后在第二行中，输出 n 个整数 x_1, x_2, \dots, x_{n_x} ($0 \leq x_i < i$) --其中 x_i 是父节点。
树 X 中的第 i 个顶点。



然后在第三行输出 n 个整数 y_1, y_2, \dots, y_{ny} ($0 \leq y_i < i$) --其中 y_i 是树 Y 中第 i 个顶点的父顶点。

如果有多个解决方案，请打印其中任何一个。

实例

标准输入	标准输出
2 2 3 10 0 1 0 1 2 0 1 1 3 4 3 6 3 1 9 4 3 10 0 1 2 2 0 1 2 0 1 1 3 4 3 6 3 1 9	不可能 2 1 0 1 0
1 5 5 49 0 1 1 3 1 0 1 2 1 2 0 1 2 3 4 1 6 7 8 9 1 11 12 13 14 11 16 17 18 19 1 21 22 23 24 1 26 26 1 1 30 31 31 30 30 35 36 36 35 30 40 41 41 40 1 45 46 46 45	5 5 0 1 2 3 4 0 1 1 3 3



问题 K. 多样性和差异

时间限制

1 秒 内存限制

256 兆字节

Helianthuswolf Co., Ltd. 是一家跨国 "Intestnet" 公司。其来自全球市场的收入逐年增长。Helianthuswolf 公司的文化是其成功的最重要关键。

Helianthuswolf 的企业文化是什么？Helianthuswolf 重视创新、激情、质量、合作和多样性。随着 Helianthuswolf 公司成长为一家成功的跨国巨头公司，其全球办事处的数量也在不断增加，现在它最看重的就是多样性。

无论如何，这艘巨轮并非一帆风顺。Helianthuswolf 最近遭遇了金融危机。为了度过危机，公司决定裁员。每个部门都要裁掉一定数量的程序员。为了消除年龄歧视，尊重多元化文化，Helianthuswolf 决定在裁员后保持年龄多元化。

但《Helianthuswolf》中的招聘人员略显夸张和刻板。他们需要一个量化的公式来衡量多样性。读完数学书后，他们认为方差是衡量多样性的最佳标准。因此，他们希望最大限度地提高裁员后剩余程序员年龄的方差。

在概率论和统计学中，方差是对随机偏差平方的期望值。

变量的方差。 $\{x_1, x_2, \dots, x_k\}$ 的方差是 $\frac{1}{k} \sum_{i=1}^k (x_i - \frac{1}{k} \sum_{j=1}^k x_j)^2$ 。

请帮助招聘人员编写一个程序，以决定哪些程序员将被解雇。

输入

有多个测试用例。第一行输入包含一个整数 T ，表示测试用例的数量。对于每个测试用例

第一行包含两个整数 n 和 m ($1 \leq n \leq 10^5$, $0 \leq m < n$)，表示该团队的程序员人数和公司将裁掉的程序员人数。

第二行包含 n 个整数 a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^4$)，表示每个程序员的年龄。保证所有情况下的 n 之和小于 2×10^6 。

输出

对于每个测试用例，按递增顺序在一行中输出 $n - m$ 个整数。 $(n - m)$ 个整数是其余程序员的指数。如果存在多个方差最大化的解决方案，请选择词典中最小的一个。请勿在每行末尾输出额外的空格。

◆◆◆◆ $\{x_1, x_2, \dots, x_p\}$ ◆◆◆◆◆ 法上小于序列 $\{y_1, y_2, \dots, y_q\}$ 的词性小于序列 $\{y_i, y, \dots,$



x_i 的词性大于序列 $\{y_i, y\}$ 的词性 $\diamond\diamond$ 于序列 $\{y, y, \dots, x\}$ $\diamond\diamond$ 词性小于序列 $\{y, y\}$ 的词性。



示例

标准输入	标准输出
9	1
3 2	1 4
34 35 36	1 3 5
4 2	1 2 5
20 35 41 74	1 2 5
5 2	4 5
40 30 50 20 10	1
5 2	2 3 4 5
50 40 30 20 10	1 2 3 4 5
5 2	
10 20 30 40 50	
5 3	
30 40 20 10 50	
5 4	
10 20 30 40 50	
5 1	
30 50 40 20 10	
5 0	
30 50 40 20 10	

备注

在第一种情况下，解雇任何两名程序员都不会改变方差，因此我们有三个候选答案 "1"、"2 "和 "3"。在这三个答案中，"1 "的词序最小。因此，我们输出 "1"（不带引号）。

在第二种情况下，解雇一名 35 岁（第 2 位）和一名 41 岁（第 3 位）的程序员可使方差最大化。由于 "1 4 "在词法上小于 "4 1"，因此我们输出 "1 4"（不带引号）。

在第三种情况中，我们有 12 个候选答案 "1 3 5"、"1 5 3"、"3 1 5"、"3 5 1"、"5 1 3"、"5 3 1"、"3 4 5"、"3 5 4"、"4 3 5"、"4 5 3"、"5 3 4 "和 "5 4 3"。由于 "1 3 5 "在词典中最小，因此我们输出 "1 3 5"（不带引号）。



问题 L.一维迷宫

时间限制 1 秒 内存限
制 64 兆字节

小宝被困在一个一维迷宫中，迷宫由 n 个排成一行的网格组成！网格从左到右编号为 1 到 n ，第 i 个网格上标有字符 s_i ，其中 s_i 为 "L" 或 "R"。

从第 m 个网格开始，小宝会重复执行以下步骤，直到逃出迷宫：

- 如果 "包包" 位于第 1 个网格或第 n 个网格，则认为 "包包" 到达了出口，可以成功逃脱。
- 否则，让小宝进入 t 个网格。如果 $s_t = 'L'$ ，包包将移动到第 $(t - 1)$ 个网格；如果 $s_t = 'R'$ ，小宝将移动到第 $(t + 1)$ 个网格。

在采取上述步骤之前，包包可以改变某些网格中的字符，以帮助自己逃脱。具体来说，对于第 i 个网格，小宝可以把 s_i 从 "L" 改成 "R"，或者从 "R" 改成 "L"。

不过，在格子里换字是件很累的事。您的任务是帮助包包计算出逃离迷宫所需的最少格子数。

输入

有多个测试用例。输入的第一行包含一个整数 T ，表示测试用例的数量。对于每个测试用例

第一行包含两个整数 n 和 m ($3 \leq n \leq 10^5$, $1 < m < n$)，表示迷宫中的网格数和起始网格的索引。

第二行包含由字符 "L" 和 "R" 组成的字符串 s ($|s| = n$)。 s 的第 i 个字符表示第 i 个网格中的字符。

保证所有测试用例的 n 之和不超过 10^6 。

输出

每个测试案例输出一行，每行包含一个整数，表示小宝逃离迷宫所需的最少网格数。

示例

标准输入	标准输出
3	0
3 2	2
LRL	1
10 4	
RRRRRRRLLR	
7 4	



RLLRLLR

备注

在第一个示例测试用例中，"包包"不需要更改任何字符，并且可以从第 3 格逃逸。所以答案是 0。

对于第二个示例测试用例，小宝可以将 s_8 改为 "R"，将 s_9 改为 "R"，并从第 10 个网格中逃脱。因此答案是 2。

在第三个示例测试用例中，小宝可以将 s_4 改为 "L"，并从第 1 格逃出。所以答案是 1。



问题 M. 最安全的建筑

时间限制 1 秒 内存限制
制 64 兆字节

PUBG 是一款多人在线对战视频游戏。在游戏中，多达一百名玩家跳伞到一个小岛上，搜刮武器和装备杀死其他人，同时避免自己被杀死。包包是这款游戏的忠实粉丝，但这次他在选择最安全的建筑时遇到了一些麻烦。

游戏中有 n 座建筑物散布在岛上，我们将这些建筑物视为二维平面上的点。每轮游戏开始时，岛屿上会出现一个以 $(0, 0)$ 为中心、半径为 R 的圆形安全区域。一段时间后，该安全区域将向半径为 r 的随机圆缩小 ($r \leq R$)。整个新安全区域完全包含在原安全区域内（可能与原安全区域相切），新安全区域的中心在原安全区域内均匀选择。

新安全区域覆盖的建筑物称为安全建筑物。考虑到安全区域的半径和建筑物的位置，"包包"希望找到所有成为安全建筑物概率最大的建筑物。

输入

有多个测试用例。第一行输入包含一个整数 T ，表示测试用例的数量。对于每个测试用例

第一行包含三个整数 n ($1 \leq n \leq 100$)、 R 和 r ($1 \leq r \leq R \leq 10^4$)，表示建筑物的数量和两个安全圆的半径。

下面的 n 行分别包含两个整数 x_i 和 y_i ($-10^4 \leq x_i, y_i \leq 10^4$)，表示建筑物的坐标。这里我们假设原安全圆的圆心位于 $(0, 0)$ ，所有建筑物都在原圆内。

保证所有测试用例的 n 之和不超过 5000。

输出

每个测试用例输出两行。

第一行包含一个整数 m ，表示最有可能成为安全楼宇的楼宇数量。

第二行包含 m 个整数，由空格隔开，按升序排列，表示最安全建筑物的指数。

请不要在每行末尾输出额外的空格。

示例

标准输入	标准输出
2	1
3 10 5	1
3 4	2
3 5	2 3



3 6	
3 10 4	
-7 -6	
4 5	
5 4	