齐鲁软件园杯
2024 中国大学生程序设计竞赛
全国邀请赛（山东）
暨 CCPC 山东省大学生程序设计竞赛

正式赛

2024 年 5 月 26 日

试题列表

A	打印机
B	三角形
C	多彩的线段 2
D	王国英雄
E	传感器
F	分割序列
G	宇宙旅行
H	阻止城堡
I	左移
J	多彩的生成树
K	矩阵
L	路径的交
M	回文多边形

本试题册共 13 题，17 页。
如果您的试题册缺少页面，请立即通知志愿者。

由 SUA 程序设计竞赛命题组命题。
https://sua.ac/

承办方

SHANDONG UNIVERSITY 1901

CIIIC
国家信息通信国际创新园
CHINA INTERNATIONAL ICT INNOVATION CLUSTER

命题方

竞赛过程中访问非竞赛网页是违反竞赛规则的行为。
如果您有兴趣（我们很荣幸），
请在竞赛后扫描二维码。

Problem A. 打印机

SUA 程序设计竞赛命题组的裁判们正在为即将举行的 2024 中国大学生程序设计竞赛全国邀请赛（山东）暨 CCPC 山东省大学生程序设计竞赛打印试题。

文印店里共有 n 台打印机。第 i 台打印机每 t_i 秒可以打印一份试题。然而，第 i 台打印机每次打印出 l_i 份试题后，必须停机 w_i 秒防止过热。也就是说，第 i 台打印机将重复进行以下工作计划：持续工作 $t_i \times l_i$ 秒，然后停机 w_i 秒。

裁判们将同时使用所有打印机。求打印 k 份试题至少需要多少秒。

Input

有多组测试数据。第一行输入一个整数 T（$1 \le T \le 100$）表示测试数据组数，对于每组测试数据：

第一行输入两个整数 n 和 k（$1 \le n \le 100$，$1 \le k \le 10^9$）表示打印机的数量和需要的试题数量。

对于接下来的 n 行，第 i 行输入三个整数 t_i，l_i 和 w_i（$1 \le t_i, l_i, w_i \le 10^9$）。它们的意义如上所述。

Output

每组数据输出一行一个整数，表示打印试题至少需要多少秒。

Example

standard input	standard output
2 3 15 3 4 5 5 7 2 1 2 20 1 100 1 1 100	25 10000

Note

对于第一组样例数据，在 25 秒内，第一台打印机可以打印 6 份试题，第二台打印机可以打印 5 份试题，第三台打印机可以打印 4 份试题。所以一共打印了 $6 + 5 + 4 = 15$ 份试题。

Problem B. 三角形

给定 n 个由小写英文字母构成的字符串 $S_1, S_2, \cdots, S_n$，称三个字符串 S_a，S_b 和 S_c 构成了一个三角形，若它们满足以下所有限制：

- $S_a + S_b > S_c$ 或 $S_b + S_a > S_c$。
- $S_a + S_c > S_b$ 或 $S_c + S_a > S_b$。
- $S_b + S_c > S_a$ 或 $S_c + S_b > S_a$。

这里的 $+$ 表示字符串连接操作。字符串通过字典序比较大小。例如，`ba`，`cb` 和 `cbaa` 构成了一个三角形，因为：

- `cb` + `ba` = `cbba` > `cbaa`.
- `cbaa` + `ba` = `cbaaba` > `cb`.
- `cb` + `cbaa` = `cbcbaa` > `ba`.

计算整数三元组 (a, b, c) 的数量，满足 $1 \le a < b < c \le n$ 且 S_a，S_b，S_c 构成了一个三角形。

Input

有多组测试数据。第一行输入一个整数 T 表示测试数据组数。对于每组测试数据：

第一行输入一个整数 n（$1 \le n \le 3 \times 10^5$）表示字符串的数量。

对于接下来的 n 行，第 i 行输入一个由小写字母构成的字符串 S_i（$1 \le |S_i| \le 3 \times 10^5$）。

保证单组数据所有字符串的总长度不超过 3×10^5，所有数据所有字符串的总长度不超过 10^6。

Output

每组数据输出一行一个整数，表示合法的三元组数量。

Example

standard input	standard output
3 6 cbaa cb cb cbaa ba ba 3 sdcpc sd cpc 1 ccpc	16 0 0

Problem C. 多彩的线段 2

考虑数轴上的 n 条线段，其中第 i 条线段的左端点为 l_i，右端点为 r_i。您需要将每条线段涂上 k 种颜色中的一种，使得任意两条具有相同颜色的线段都没有重合。

求给线段涂色的方案数。

称第 i 条线段和第 j 条线段有重合，若存在一个实数 x 同时满足 $l_i \le x \le r_i$ 且 $l_j \le x \le r_j$。

称两种涂色方案是不同的，若存在一条线段在两种方案中被涂上了不同的颜色。

Input

有多组测试数据。第一行输入一个整数 T 表示测试数据组数。对于每组测试数据：

第一行输入两个整数 n 和 k（$1 \le n \le 5 \times 10^5$，$1 \le k \le 10^9$）表示线段的数量和颜色的数量。

对于接下来的 n 行，第 i 行输入两个整数 l_i 和 r_i（$1 \le l_i \le r_i \le 10^9$）表示第 i 条线段的左右端点。

保证所有数据 n 之和不超过 5×10^5。

Output

每组数据输出一行一个整数表示答案。由于答案可能很大，请将答案对 998244353 取模后输出。

Example

standard input	standard output
2 4 3 4 7 3 4 5 8 1 3 2 1000 100 200 300 400	24 1000000

Note

令 c_i 表示第 i 条线段的颜色。

对于第一组样例数据，一种合法的涂色方案是令 $c_1 = 1$，$c_2 = 3$，$c_3 = 3$ 以及 $c_4 = 1$。因为第 1 条和第 4 条线段没有重合，第 2 条和第 3 条线段也没有重合。

然而， $c_1 = 1$，$c_2 = 2$，$c_3 = 1$ 以及 $c_4 = 3$ 不是一种合法的方案。因为第 1 条和第 3 条线段互相重合，不能有一样的颜色。

Problem D. 王国英雄

《王国英雄》是一款点击解谜类的冒险游戏。游戏中的主人公为了拯救他/她的父亲，踏上了一段危险的旅途，并成为了王国的英雄。

游戏中的货币被称为“金币”，可以用于购买各种补给品，甚至还能用于完成特定任务。俗话说得好，“钱永远不嫌多”，我们的天才玩家堡堡刚刚就找到了一种变得富有的方法。游戏中有一座磨坊，磨坊主以每袋 p 金币的价格售卖面粉。游戏中还有一座酒馆，酒保以每袋 q 金币（$q > p$）的价格收购面粉。显然堡堡可以赚取其中的差价，但在两处地点之间移动，以及点击购买和卖出的按钮都需要时间。

更精确地，如果堡堡一次性从磨坊购买了 x 袋面粉，需要花 $(ax + b)$ 秒以及 px 金币；如果堡堡一次性向酒馆卖出了 x 袋面粉，需要花 $(cx + d)$ 秒，但能赚取 qx 金币。堡堡现在有 m 金币，但因为堡堡马上就要上床睡觉了，他最多只能再玩 t 秒游戏。求堡堡打完游戏时最多能持有多少金币。

Input

有多组测试数据。第一行输入一个整数 T（$1 \le T \le 500$）表示测试数据组数，对于每组测试数据：

第一行输入三个整数 p，a 和 b（$1 \le p, a \le 10^9$，$0 \le b \le 10^9$）。

第二行输入三个整数 q，c 和 d（$p < q \le 10^9$，$1 \le c \le 10^9$，$0 \le d \le 10^9$）。

第三行输入两个整数 m 和 t（$1 \le m, t \le 10^9$）。

Output

每组数据输出一行一个整数，表示堡堡经过至多 t 秒后最多能持有多少金币。

Example

standard input	standard output
3 5 2 3 8 1 5 14 36 5 2 0 8 1 3 17 6 100 1 0 10000 1 0 99 100000	32 20 99

Note

对于第一组样例数据，一种最优方案是：

- 堡堡首先从磨坊购买 2 袋面粉，花费 $2 \times 2 + 3 = 7$ 秒以及 $5 \times 2 = 10$ 金币。然后他把所有面粉卖给酒馆，花费 $1 \times 2 + 5 = 7$ 秒但赚取了 $8 \times 2 = 16$ 金币。堡堡现在有 $14 - 10 + 16 = 20$ 金币，还剩 $36 - 7 - 7 = 22$ 秒。
- 堡堡接下来从磨坊购买 4 袋面粉，花费 $2 \times 4 + 3 = 11$ 秒以及 $5 \times 4 = 20$ 金币。然后他把所有面粉卖给酒馆，花费 $1 \times 4 + 5 = 9$ 秒但赚取了 $8 \times 4 = 32$ 金币。堡堡现在有 $20 - 20 + 32 = 32$ 金币，还剩 $22 - 11 - 9 = 2$ 秒。
- 现在堡堡没有时间买卖面粉了。所以答案是 32。

对于第二组样例数据，堡堡只有时间买卖一袋面粉。所以答案是 $17 - 5 + 8 = 20$。

对于第三组样例数据，堡堡没有足够的金币购买面粉。所以答案是 99。

Problem E. 传感器

有 n 颗红球排成一行，从左到右编号从 0 到 $(n-1)$（含两端）。我们将进行 n 次操作，其中第 i 次操作将第 a_i 颗球涂成蓝色。所有操作结束后，所有球都会变成蓝色。

有 m 个编号从 1 到 m（含两端）的传感器监控球的颜色。若第 l_i 颗球到第 r_i 颗球（含两端）里恰有一颗红球，则第 i 个传感器将进入激活状态；否则传感器将保持非激活状态。

问每次操作结束后，哪些传感器处于激活状态。

更具体地，设第 i 次操作结束后共有 k_i 个传感器处于激活状态，它们的编号是 $s_{i,1}, s_{i,2}, \cdots, s_{i,k_i}$。对于每个 $0 \le i \le n$，输出 $v_i = \sum\limits_{j=1}^{k_i} s_{i,j}^2$。特别地，定义 $v_0 = \sum\limits_{j=1}^{k_0} s_{0,j}^2$，其中 k_0 是第一次操作之前处于激活状态的传感器数量，它们的编号为 $s_{0,1}, s_{0,2}, \cdots, s_{0,k_0}$。

Input

有多组测试数据。第一行输入一个整数 T 表示测试数据组数。对于每组测试数据：

第一行输入两个整数 n 和 m（$1 \le n, m \le 5 \times 10^5$）表示球的数量和传感器的数量。

对于接下来 m 行，第 i 行输入两个整数 l_i 和 r_i（$0 \le l_i \le r_i < n$）表示第 i 个传感器的检测范围。

接下来的一行输入 n 个整数 $a'_1, a'_2, \cdots, a'_n$（$0 \le a'_i < n$），其中 a'_i 表示 **加密后的** 第 i 次操作。a_i 的真实值等于 $(a'_i + v_{i-1}) \bmod n$，其中 v_{i-1} 是第 $(i-1)$ 次操作后的答案，在上述描述中已有定义。这些加密后的操作强制您必须计算好当前操作的答案，才能处理下一个操作。保证解密后 a_i 互不相同。

保证所有数据 n 之和与 m 之和均不超过 5×10^5。

Output

每组数据输出一行 $(n+1)$ 个由单个空格分隔的整数 $v_0, v_1, \cdots, v_n$。v_i 的含义在上述描述中已有定义。

Example

standard input	standard output
3 5 4 2 4 2 3 3 3 0 2 3 2 4 2 0 2 1 1 1 1 0 2 1 0 1 0 0	9 13 29 17 16 0 1 1 0 0 1 0

Note

对于第一组样例数据:

- 在第一次操作之前，只有传感器 3 处于激活状态，所以 $v_0 = 3^2 = 9$。
- 对于第 1 次操作，真实的 $a_1 = (3+9) \bmod 5 = 2$。本次操作后，传感器 2 和 3 处于激活状态，所以 $v_1 = 2^2 + 3^2 = 13$。
- 对于第 2 次操作，真实的 $a_2 = (2+13) \bmod 5 = 0$。本次操作后，传感器 2，3 和 4 处于激活状态，所以 $v_2 = 2^2 + 3^2 + 4^2 = 29$。
- 对于第 3 次操作，真实的 $a_3 = (4+29) \bmod 5 = 3$。本次操作后，传感器 1 和 4 处于激活状态，所以 $v_3 = 1^2 + 4^2 = 17$。
- 对于第 4 次操作，真实的 $a_4 = (2+17) \bmod 5 = 4$。本次操作后，只有传感器 4 处于激活状态，所以 $v_4 = 4^2 = 16$。
- 对于第 5 次操作，真实的 $a_5 = (0+16) \bmod 5 = 1$。本次操作后，没有传感器处于激活状态，所以 $v_5 = 0$。

Problem F. 分割序列

给定长度为 n 的整数序列 $a_1, a_2, \cdots, a_n$，请将序列分成 k 段连续且非空的子数组，使得序列中的每个元素恰属于一个子数组。令 s_i 表示从左到右第 i 个子数组里的元素之和，对于每个 $1 \le k \le n$，求下式的最大值。

$$\sum_{i=1}^{k} i \times s_i$$

更正式地，对于每个 $1 \le k \le n$，令 $r_0 = 0$ 以及 $r_k = n$，您需要找到 $(k-1)$ 个整数 $r_1, r_2, \cdots, r_{k-1}$ 满足 $r_0 < r_1 < r_2 < \cdots < r_{k-1} < r_k$，并最大化下式的值。

$$\sum_{i=1}^{k} i \times (\sum_{j=r_{i-1}+1}^{r_i} a_j)$$

Input

有多组测试数据。第一行输入一个整数 T 表示测试数据组数，对于每组测试数据：

第一行输入一个整数 n（$1 \le n \le 5 \times 10^5$）表示序列的长度。

第二行输入 n 个整数 $a_1, a_2, \cdots, a_n$（$-10^6 \le a_i \le 10^6$）表示序列。

保证所有数据 n 之和不超过 5×10^5。

Output

每组数据输出一行 n 个由单个空格分隔的整数 $v_1, v_2, \cdots, v_n$，其中 v_i 表示 $k = i$ 时的答案。

Example

standard input	standard output
2 6 1 3 -4 5 -1 -2 1 100	2 4 5 3 1 -2 100

Note

对于第一组样例数据，考虑 $k = 3$，可以将序列分割为 $\{\{1\}, \{3, -4\}, \{5, -1, -2\}\}$。答案是 $1 \times 1 + 2 \times (3 - 4) + 3 \times (5 - 1 - 2) = 5$。

Problem G. 宇宙旅行

堡堡是一位宇宙旅行者，穿梭于无穷多个平行宇宙之间。每个宇宙都有一个整数编号，编号从 0 开始。

每个宇宙里都有 n 个魔法苹果。虽然这些宇宙之间有很多相似之处，它们仍然有细微的不同。在第 j 个宇宙里，第 i 个魔法苹果的魔法能量值为 $a_i \oplus j$，这里 $\oplus$ 是按位异或运算。

堡堡是一个优柔寡断的人，所以他准备了 q 个旅行计划。每个旅行计划可以记为三个整数 l，r 和 k，表示堡堡将访问编号从 l 到 r 的每个宇宙（含两端），并从每个宇宙的 n 个苹果里，收集魔法能量值第 k 小的苹果。

对每个旅行计划，求堡堡收集的苹果的魔法能量值之和。请注意，旅行计划不会真的把苹果从每个宇宙中拿走。也就是说，每次询问是独立的。

Input

每个测试文件仅有一组测试数据。

第一行输入两个整数 n 和 q（$1 \le n, q \le 10^5$）表示每个宇宙里苹果的数量以及旅行计划的数量。

第二行输入 n 个整数 $a_1, a_2, \cdots, a_n$（$0 \le a_i < 2^{60}$）。

对于接下来 q 行，第 i 行输入三个整数 l_i, r_i 和 k_i（$0 \le l_i \le r_i < 2^{60}$，$1 \le k_i \le n$）表示第 i 个旅行计划。

Output

每个旅行计划输出一行一个整数表示答案。由于答案可能很大，请将答案对 998244353 取模后输出。

Example

standard input	standard output
8 3 2 0 2 4 0 5 2 6 1 1 6 2 7 5 0 1048575 4	4 23 720895450

Problem H. 阻止城堡

一块有 10^9 行和 10^9 列的棋盘上放着 n 个城堡与 m 个障碍物。每个城堡或障碍物恰好占据一个格子，且被占据的格子两两不同。两座城堡可以互相攻击，若它们位于同一行或同一列，且它们之间没有障碍物或其它城堡。更正式地，令 (i,j) 表示位于第 i 行第 j 列的格子，位于 (i_1,j_1) 和 (i_2,j_2) 的两座城堡可以互相攻击，若以下条件中有一条成立：

- $i_1=i_2$，且对于所有 $\min(j_1,j_2)<j<\max(j_1,j_2)$，不存在位于 (i_1,j) 的障碍物或城堡。
- $j_1=j_2$，且对于所有 $\min(i_1,i_2)<i<\max(i_1,i_2)$，不存在位于 (i,j_1) 的障碍物或城堡。

找出一种方法，向棋盘上额外添加最少的障碍物，使得任意两座城堡都不能互相攻击。请注意：不能将额外的障碍物放在已经被占据的格子里。

Input

有多组测试数据。第一行输入一个整数 T 表示测试数据组数，对于每组测试数据：

第一行输入一个整数 n（$2\le n\le 200$）表示城堡的数量。

对于接下来 n 行，第 i 行输入两个整数 r_i 和 c_i（$1\le r_i,c_i\le 10^9$），表示第 i 座城堡位于第 r_i 行第 c_i 列。

接下来的一行输入一个整数 m（$0\le m\le 200$）表示障碍物的数量。

对于接下来 m 行，第 i 行输入两个整数 r'_i 和 c'_i（$1\le r'_i,c'_i\le 10^9$），表示第 i 个障碍物位于第 r'_i 行第 c'_i 列。

保证被占据的格子两两不同。同时保证所有数据 n 之和与 m 之和均不超过 400。

Output

对于每组数据：

如果能阻止城堡之间互相攻击，首先输出一行一个整数 k，表示最少需要额外添加多少障碍物。接下来输出 k 行，其中第 i 行包含两个由单个空格分隔的整数 x_i 和 y_i（$1\le x_i,y_i\le 10^9$），表示您准备将第 i 个额外障碍物放在格子 (x_i,y_i) 里。如果有多种合法答案，您可以输出任意一种。

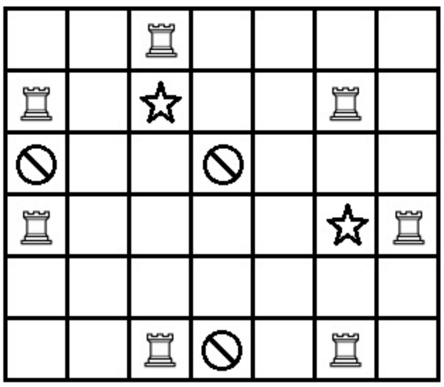
如果无法阻止城堡之间互相攻击，只要输出一行 `-1`。

Example

standard input	standard output
4 7 1 3 6 6 4 7 2 1 6 3 4 1 2 6 3 3 4 6 4 3 1 2 1 1 2 2 0 3 1 1 1 3 3 3 1 1 2 3 1 1 1 3 2 3 0	2 2 3 4 6 0 1 2 3 -1

Note

第一组样例数据如下图所示。我们只需要添加 2 个额外的障碍物（图中用星星标识），其中一个位于 $(2, 3)$，另一个位于 $(4, 6)$。



对于第二组样例数据，两座城堡既不在同一行也不在同一列，因此不需要障碍物。

Problem I. 左移

称一个字符串是美丽的，若它的第一个字符和最后一个字符相同。

给定长度为 n 的字符串 $S = s_0 s_1 \cdots s_{n-1}$，令 $f(S,d)$ 表示将 S 左移 d 次后获得的字符串。也就是说 $f(S,d) = s_{(d+0) \bmod n} s_{(d+1) \bmod n} \cdots s_{(d+n-1) \bmod n}$。求最小的非负整数 d 满足 $f(S,d)$ 是美丽的。

Input

有多组测试数据。第一行输入一个整数 T 表示测试数据组数，对于每组测试数据：

第一行输入一个仅由小写英文字母组成的字符串 $s_0 s_1 \cdots s_{n-1}$（$1 \le n \le 5 \times 10^5$）。

保证所有数据 n 之和不超过 5×10^5。

Output

每组数据输出一行一个整数，表示满足 $f(S,d)$ 是美丽的最小非负整数 d。若不存在这样的 d，输出 `-1`。

Example

standard input	standard output
4 helloccpc abcdcba x abc	3 0 0 -1

Note

对于第一组样例数据，$f(S,3) =$ `loccpchel`。它的第一个字符和最后一个字符都是 `l`，所以它是一个美丽字符串。虽然 $f(S,6) =$ `cpchelloc` 也是美丽的，我们需要回答最小的非负整数 d。所以答案是 3。

Problem J. 多彩的生成树

堡堡有很多彩色的节点。颜色的编号从 1 到 n（含两端），第 i 种颜色共有 a_i 个节点。因为堡堡刚刚在算法课上学习了最小生成树问题，他打算利用这些节点做一些练习。

每一对节点都会被一条带有权值的边连接。每一条边的权值只和它两个端点的颜色有关。具体来说，令 c_u 表示节点 u 的颜色，若一条边连接了节点 u 和 v，它的权值就是 b_{c_u,c_v}。

请帮助堡堡求出这张图的最小生成树的总权值。

请回忆：最小生成树是一张带权连通图的边的子集，这些边连通了所有节点，不会形成环，且总权值最小。

Input

有多组测试数据。第一行输入一个整数 T 表示测试数据组数。对于每组测试数据：

第一行输入一个整数 n（$1 \le n \le 10^3$）表示颜色的种数。

第二行输入 n 个整数 $a_1, a_2, \cdots, a_n$（$1 \le a_i \le 10^6$），其中 a_i 表示颜色 i 有几个节点。

对于接下来的 n 行，第 i 行输入 n 个整数 $b_{i,1}, b_{i,2}, \cdots, b_{i,n}$（$1 \le b_{i,j} \le 10^6$），其中 $b_{i,j}$ 表示两个端点的颜色分别为 i 和 j 的边的权值。保证对于所有 $1 \le i, j \le n$ 有 $b_{i,j} = b_{j,i}$。

保证所有数据 n 之和不超过 10^3。

Output

每组数据输出一行一个整数，表示最小生成树的总权值。

Example

standard input	standard output
3 3 100 1 1 1 100 2 100 100 1 2 1 100 2 3 3 100 1 1 100 1 1 5	102 5 0

Problem K. 矩阵

构造一个 n 行 n 列的矩阵，满足以下所有条件：

- 矩阵的元素是从 1 到 $2n$ 的整数（含两端）。
- 每个从 1 到 $2n$ 的整数（含两端）在矩阵里至少出现一次。
- 令 $a_{i,j}$ 表示第 i 行第 j 列的元素，恰有一个整数四元组 (x, y, z, w) 满足：
  - $1 \le x < z \le n$。
  - $1 \le y < w \le n$。
  - $a_{x,y}$，$a_{x,w}$，$a_{z,y}$，$a_{z,w}$ 互不相同。

Input

每个测试文件仅有一组测试数据。

第一行输入一个整数 n（$2 \le n \le 50$）表示矩阵的大小。

Output

如果可以构造出这样的矩阵，首先输出一行 `Yes`。接下来输出 n 行，其中第 i 行输出 n 个由单个空格分隔的整数 $a_{i,1}, a_{i,2}, \cdots, a_{i,n}$（$1 \le a_{i,j} \le 2n$），其中 $a_{i,j}$ 表示矩阵第 i 行第 j 列的元素。如果有多种合法答案，您可以输出任意一种。

如果无法构造出这样的矩阵，只需要输出一行 `No`。

Examples

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

Problem L. 路径的交

一棵树有 n 个节点与 $(n-1)$ 条边，其中第 i 条边连接节点 u_i 与 v_i，权值为 w_i。

您需要处理 q 次询问。第 i 次询问可以记为三个整数 a_i，b_i 和 k_i。本次询问首先临时将第 a_i 条边的权值改为 b_i。之后您需要选择 $2k_i$ 个不同的节点 $s_1, s_2, \cdots, s_{k_i}, e_1, e_2, \cdots, e_{k_i}$ 并考虑树上的 k_i 条简单路径，其中第 p 条路径从节点 s_p 出发，到节点 e_p 结束。称一条边是好的，若它被所有 k_i 条路径包含。最大化好边的总权值。

请再次注意，所有询问对权值的修改都是临时的。在每次询问后，您需要把权值恢复原状。

Input

每个测试文件仅有一组测试数据。

第一行输入两个整数 n 和 q（$2 \le n \le 5 \times 10^5$，$1 \le q \le 5 \times 10^5$）表示节点的数量和询问的数量。

对于接下来的 $(n-1)$ 行，第 i 行输入三个整数 u_i，v_i 和 w_i（$1 \le u_i, v_i \le n$，$1 \le w_i \le 10^9$）表示第 i 条边连接节点 u_i 和 v_i，权值为 w_i。

对于接下来的 q 行，第 i 行输入三个整数 a_i，b_i 和 k_i（$1 \le a_i \le n-1$，$1 \le b_i \le 10^9$，$1 \le k_i \le \lfloor \frac{n}{2} \rfloor$）表示第 i 次询问。

Output

每次询问输出一行一个整数表示答案。

Example

standard input	standard output
7 3 1 2 20 2 3 10 2 4 40 4 6 10 1 5 30 5 7 10 2 100 1 5 50 2 2 100 3	160 110 20

Note

对于第一次询问，选择 $s_1 = 3$ 和 $e_1 = 7$。

对于第二次询问，选择 $s_1 = 4$，$s_2 = 6$，$e_1 = 7$ 和 $e_2 = 5$。

对于第三次询问，选择 $s_1 = 3$，$s_2 = 4$，$s_3 = 6$，$e_1 = 5$，$e_2 = 1$ 和 $e_3 = 7$。

Problem M. 回文多边形

给定一个有 n 个顶点的凸多边形。顶点按逆时针顺序编号从 1 到 n（含两端），第 i 个顶点有一个权值 $f(i)$。

称一个顶点的子集是回文的，若它们的权值能够按逆时针顺序组成一个回文序列。更正式地，设子集里有 k 个顶点，它们的编号按逆时针顺序为 $v_0, v_1, \cdots, v_{k-1}$。需要存在一个整数 d 满足 $0 \le d < k$，且对于所有 $0 \le i < k$ 有 $f(v_{(d+i) \bmod k}) = f(v_{(d-1-i) \bmod k})$。

在所有回文的顶点子集中，找出凸包面积最大的子集。

Input

有多组测试数据。第一行输入一个整数 T 表示测试数据组数。对于每组测试数据：

第一行输入一个整数 n（$3 \le n \le 500$）表示凸多边形的顶点数。

第二行输入 n 个整数 $f(1), f(2), \cdots, f(n)$（$1 \le f(i) \le 10^9$），其中 $f(i)$ 表示第 i 个顶点的权值。

对于接下来的 n 行，第 i 行输入两个整数 x_i 和 y_i（$-10^9 \le x_i, y_i \le 10^9$）表示第 i 个顶点的坐标。顶点按逆时针顺序列出。保证凸多边形的面积为正，且没有重合的顶点。但可能存在三点共线的情况。

保证所有数据 n 之和不超过 10^3。

Output

每组数据输出一行一个整数，表示回文顶点子集的最大凸包面积乘以 2。可以证明这个值总是一个整数。

Example

standard input	standard output
3 8 2 4 2 4 3 4 5 3 2 3 0 6 -3 3 -3 0 -2 -3 1 -5 3 -3 4 0 3 1 2 3 0 0 1 0 0 1 3 1 1 1 0 0 1 0 0 1	84 0 1

Note

第一组样例数据如下图所示。选择顶点 2，4，5，6，8，并考虑 $d=1$，权值序列 $\{4,3,4,3,4\}$ 是一个回文序列。