

Problem A. 你好, HWCWHer

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

今天, 我们欢聚一堂, 以一场算法竞赛的盛宴, 欢迎 2024 级新 HWCWHer 的入学。五所学校, 各有其底蕴, 各有其风采。你可以自由选择参加新生赛的地点, 请仔细阅读下面的介绍, 作出你内心的真实选择。

华中科技大学 (Huazhong University of Science and Technology), 坐落于中国中部的武汉市, 是一所享有盛誉的综合性研究型大学以其卓越的学术成就、深厚的文化底蕴和鲜明的工科特色而闻名遐迩。学校前身为1952年创办的华中工学院、同济医科大学、武汉城市建设学院等, 经过多年的融合与发展, 于2000年由原华中理工大学、同济医科大学、武汉城市建设学院合并组建而成, 开启了华中科技大学新的辉煌篇章。

武汉大学 (WuHan University), 坐落于中国中部历史文化名城——武汉市, 是中国最负盛名的综合性研究型大学之一, 素有"中国最美大学"之称。学校历史悠久, 文化底蕴深厚, 其办学历史可追溯至1893年创办的自强学堂, 历经百年风雨, 始终与国家民族命运紧密相连, 不断追求卓越与卓越。

武汉理工大学 (WuHan University of Technology), 坐落于九省通衢的湖北省武汉市, 是一所历史悠久、底蕴深厚、特色鲜明的大学。学校的前身可以追溯到1898年创办的湖北工艺学堂, 历经一个多世纪的发展变迁, 现已成为一所集工学、理学、文学、管理学、经济学、法学、艺术学、教育学等多学科于一体的综合性大学。

华中师范大学 (Central China Normal University), 坐落于九省通衢的湖北省武汉市。学校自1903年创办以来, 始终秉承"忠诚博雅、朴实刚毅"的华师精神, 致力于培养高素质、全面发展的优秀人才, 为中国乃至世界的教育事业和社会进步作出了重要贡献。

华中农业大学 (HuaZhong Agricultural University), 坐落于湖北省武汉市南湖之滨。学校自1898年创办以来, 始终秉承"勤读力耕, 立己达人"的校训精神, 致力于农业科技创新与人才培养, 为我国乃至全球的农业可持续发展和乡村振兴贡献了卓越力量。

Input

无输入。

Output

输出一个由大写字母组成的字符串, 表示你选择的学校英文简写。

Problem B. 全概率公式

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

GPT等语言模型训练需要爬取网站内容作为数据，然而不少平台的内容已经被AI生成的垃圾所污染

令 $P(A|F_i)$ 表示一份来自 F_i 平台的数据是被AI污染的概率， $P(F_i)$ 表示随机爬取的一份数据来自平台 F_i 的概率

给出所有的数据来源平台和上述概率，求随机爬取的一份数据是被AI污染的概率 $P(A)$

Input

第一行包含一个正整数 n ($1 \leq n \leq 100$) 表示作为数据来源的平台总数

接下来一行包含 n 个由空格分开的实数，其中第 i 个实数 f_i ($0 \leq f_i \leq 1$) 表示 $P(F_i)$ ，(在误差范围内) 保证 $\sum_{i=1}^n f_i = 1$

接下来一行包含 n 个由空格分开的实数，其中第 i 个实数 a_i ($0 \leq a_i \leq 1$) 表示 $P(A|F_i)$

Output

一行一个实数表示答案 $P(A)$ ，如果正确答案为 a 而选手输出的答案为 p ，则当且仅当 $|\frac{p-a}{a}| < 10^{-4}$ 时选手的答案被视为正确

Example

standard input	standard output
2 0.4 0.6 0.6 0.3	0.42

Problem C. LCT

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

给定一棵包含 n 个节点的以 1 为根的树，节点从 1 到 n 编号，边按照输入顺序从 1 到 $n - 1$ 编号。树的每条边一开始都处于"连上"状态。你需要支持以下操作：

对于给定的两个整数 x 和 y ，首先切换编号为 x 的边的状态，如果它原本是"断开"状态，则将其切换为"连上"状态；如果它原本是"连上"状态，则将其切换为"断开"状态。然后查询在只考虑所有当前"连上"状态下的边的情况下，与节点 y 位于同一个连通块中的根节点的编号是多少？根节点定义为连通块中深度最小的节点。

你需要处理共 q 次这样的操作，并在每次操作后输出查询结果。

Input

- 第一行包含两个整数 n, q ，分别表示树的节点数和询问个数。
- 接下来 $n - 1$ 行，每行包含两个整数 u 和 v ，表示树中的一条边连接了节点 u 和 v 。每条边都有唯一的编号，从 1 到 $n - 1$ ，按输入的顺序编号。
- 接下来 q 行，每行包含两个整数 x 和 y ，表示对于边 x 进行切换状态操作，然后查询节点 y 所在连通块的根节点。

Output

对于每次操作，在切换边 x 的状态后，输出节点 y 所在连通块的根节点。

Examples

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

Note

$2 \leq n \leq 10^6$

$1 \leq q \leq 5 \times 10^5, 1 \leq u, v \leq n, 1 \leq x \leq n - 1, 1 \leq y \leq n$

所有输入保证形成一棵树。

Problem D. 股票交易

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 512 megabytes

一只股票每天的价格为 a_i , 你可以通过低价买入高价卖出赚钱。明显, 卖出时间必须严格晚于买入时间。假设你在第 x 天买入, 第 y 天卖出, 则日均盈利为 $\frac{a_y - a_x}{y - x}$ 。特别地, 如果你选择不进行买卖, 则日均盈利为 0。现在最多可以进行一次买卖, 问: 如何买卖才能使日均盈利最高。

如果你提交的答案是 a , 正确答案是 b , 你需要保证 $|a - b| \leq 10^{-2}$ 。

Input

第一行一个整数 n ($1 \leq n \leq 10^5$), 代表你可以进行交易的天数。

第二行 n 个整数, 表示每天股票的价格 a_i ($1 \leq a_i \leq 10^9$)。

Output

最大的日均盈利

Examples

standard input	standard output
5 1 2 3 4 5	1.000
5 5 4 3 2 1	0.000

Problem E. HoMaCoMoHa!

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

HoMaCoMoHa 居住在 HMCMH 大陆上。

根据能量的强弱, HMCMH 大陆上一共有 k 种能量等级的 HoMaCoMoHa: 1 级 HoMaCoMoHa, 2 级 HoMaCoMoHa, \dots , k 级 HoMaCoMoHa。

HMCMH 大陆上的决斗发生在 2 名同为 i ($1 \leq i \leq k - 1$) 级的 HoMaCoMoHa 之间。决斗完毕后, 其中一名 HoMaCoMoHa 立即退化成为 $(i - 1)$ 级 HoMaCoMoHa, 另一名 HoMaCoMoHa 立即进化成为 $(i + 1)$ 级 HoMaCoMoHa。特别地, 如果决斗发生在 2 名同为 1 级的 HoMaCoMoHa 之间, 其中一名 HoMaCoMoHa 会进化成为 2 级 HoMaCoMoHa, 另一名 HoMaCoMoHa 会被淘汰出 HMCMH 大陆。被淘汰出 HMCMH 大陆的 HoMaCoMoHa 不会再与其他 HoMaCoMoHa 发生决斗。

- 只有相同能量等级的 2 个 HoMaCoMoHa 之间才能发生决斗。
- 能量等级为 k 级的 HoMaCoMoHa 之间不会发生决斗。

你需要求出在足够长的时间之后, HMCMH 大陆上会产生 k 级 HoMaCoMoHa 的数量。

如果存在 HoMaCoMoHa 可以发生决斗, 那么决斗一定会发生。在足够长的时间之后, 场面上不存在可以发生决斗的 HoMaCoMoHa。可以证明, 在足够长的时间之后, 能量等级为 k 级的 HoMaCoMoHa 的数量不会因为决斗顺序的改变而改变。

Input

输入包含多组数据。

第一行输入一行一个正整数 T ($1 \leq T \leq 10^5$), 表示数据的组数。

对于每组数据, 输入一行两个正整数 n ($1 \leq n \leq 10^9$), k ($2 \leq k \leq 10^9$), 表示 HMCMH 大陆初始状态下 1 级 HoMaCoMoHa 的数量和能量等级的数量。

Output

输出 T 行。对于每一组数据, 输出一行一个整数, 表示在足够长的时间之后, HMCMH 大陆上会产生 k 级 HoMaCoMoHa 的数量。

Example

standard input	standard output
4	0
1 2	1
2 2	1
4 3	45150
135452 3	

Note

在第一个样例中, 如果只有 1 名 HoMaCoMoHa, 那么不会发生决斗, 所以不会有 HoMaCoMoHa 2 段产生。

在第三个样例中, 4 名 HoMaCoMoHa 共产生 2 次决斗, 产生 2 名 2 级 HoMaCoMoHa, 另外 2 名 HoMaCoMoHa 被淘汰出 HMCMH 大陆。进而这 2 名 HoMaCoMoHa 之间产生 1 次决斗, 产生 1 名 1 级 HoMaCoMoHa 和 1 名 3 级 HoMaCoMoHa。

Problem F. Starlight

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

某地的地面上共有 n 块石块堆成一列，从左到右第 i 块的大小为 a_i ，其中序列 a 为一 $1 \sim n$ 的排列。小 H 打算将这 n 块石块堆成一座高塔，并登上高塔，从而摘得星星。

在从第 1 天开始的每天，小 H 将取出地面上所有石块中最靠左的一块，堆到高塔的顶部（最上方）。然而，若此时高塔顶部的石块的大小比高塔中任意一块其它石块要大，则高塔将倒塌，其中所有石块将按照等概率随机的顺序回到地面上石块序列的最左侧。否则，若高塔没有倒塌，且所有的 n 块石块都在高塔中，则小 H 成功摘得星星。

现有 q 次事件，在第 i 次事件中，地面上第 x 块石块和第 y 块石块交换了位置。即，交换了序列 a 中的 a_x 和 a_y 。请你设计程序，在一开始以及每次事件发生之后，计算出小 H 期望在第几天可以摘得星星。

可以证明答案是一个整数，请你输出答案对 $10^9 + 7$ 取模的结果。

Input

第一行两个整数 n 和 q ($1 \leq n, q \leq 2 \times 10^5$)，分别表示石块的数量，和需要处理的事件的个数。

接下来一行 n 个整数 a_1, a_2, \dots, a_n ，保证 $1 \leq a_i \leq n$ 且 a_i 互不相同。换言之， a 是一个 $1 \sim n$ 的排列。

接下来 q 行，每行两个整数 x 和 y ($1 \leq x, y \leq n, x \neq y$)，表示在第 i 次事件中，地面上第 x 块石块和第 y 块石块交换了位置。

Output

输出共 $q + 1$ 行。第一行表示在所有事件发生之前的答案。接下来对每次事件输出一行，表示该事件发生之后的答案。需要注意的是，事件之间不是相互独立的，每次事件交换之后序列 a 不会复原。

Examples

standard input	standard output
2 2	6
1 2	2
1 2	6
1 2	
3 3	22
1 2 3	18
1 2	7
2 3	22
1 3	

Note

对于第一组样例，若石块序列为 $2, 1$ ，则小 H 必然可以在第 2 天摘得星星。否则，若石块的序列为 $1, 2$ ，则高塔必然将在第二天倒塌，石块序列随机变为 $1, 2$ 或 $2, 1$ 之一。

若设小 H 在石块序列为 $1, 2$ 时，期望在第 z 天可以摘得星星，则可以列出方程： $z = 2 + \frac{1}{2}(2 + z)$ ，可以解得 $z = 6$ 。

在一开始，石块序列为 $1, 2$ ，因此你输出 6。第一次事件发生之后，序列被交换为 $2, 1$ ，因此输出 2。第二次事件发生之后，序列又从 $2, 1$ 再次被交换成了 $1, 2$ ，因此输出 6。

Problem G. Mobiusp的希望树

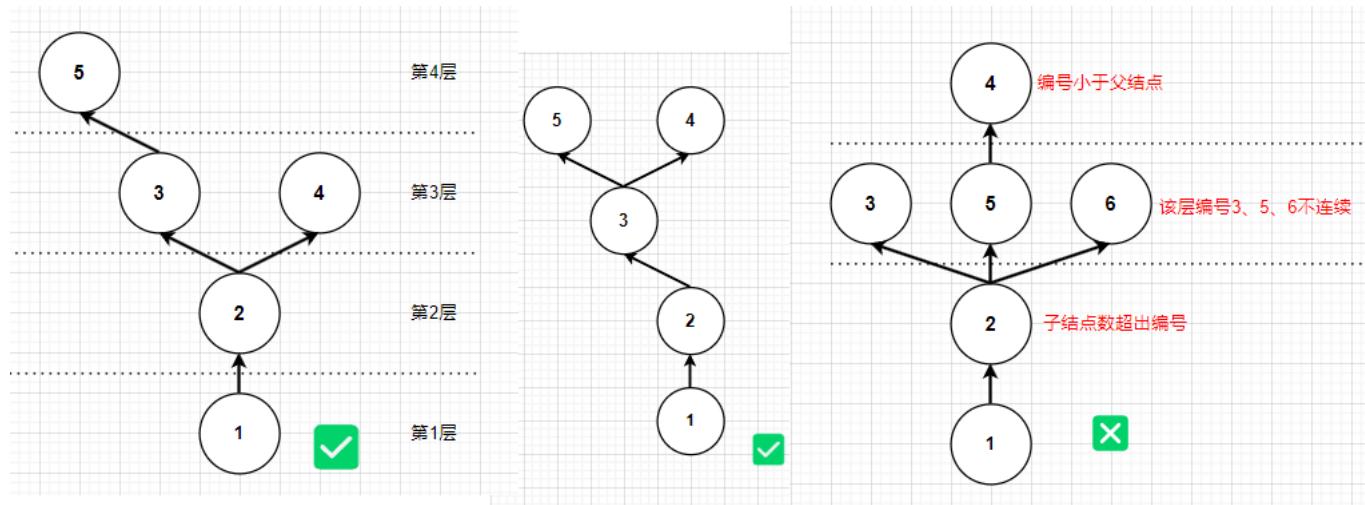
Input file: standard input
 Output file: standard output
 Time limit: 1 second
 Memory limit: 256 megabytes

Mobiusp 有 n 个树结点，编号分别为 1 到 n 。

Mobiusp 需要构建一颗有 n 个结点的根结点编号为 1 的有根树，构建出来的树需要满足以下条件：

1. 每个结点的子结点个数不能超过自己的编号
2. 每个结点的编号必须大于其父结点的编号
3. 每层结点的编号必须是连续的

Mobiusp 希望最大化最后一层（距离根结点最远的一层）的结点个数。



Input

第一行一个整数 n ($2 \leq n \leq 10^{12}$)，代表结点的个数。

Output

输出一个整数，表示最后一层结点的最大个数。

Examples

standard input	standard output
5	2
6	3

Problem H. 父子局

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

给定一个正整数 n , 输出两个不同的正整数 x, y , 满足 $\frac{1}{n} = \frac{1}{x} + \frac{1}{y}$ 。

Input

第一行一个正整数 $T(1 \leq T \leq 1 \times 10^5)$, 表示有 T 组测试数据。

接下来 T 组测试数据, 每组测试数据输入一个正整数 $n(1 \leq n \leq 1 \times 10^5)$, 表示需要分解的数字。

Output

对于每组测试数据, 输出两个不同的正整数 $x, y(1 \leq x, y \leq 1 \times 10^{18})$, 满足 $\frac{1}{n} = \frac{1}{x} + \frac{1}{y}$ 。

如果没有满足条件的 x, y , 请输出 -1 -1。

Example

standard input	standard output
3	-1 -1
1	3 6
2	4 12
3	

Problem I. 轮符雨

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

一场连续 n 天的大雨即将席卷而来，其中第 i 天雨势的大小为 a_i 。Soyo 不喜欢雨天，她期待着这场大雨能停止，在第 1 天的雨结束时期待值为 0，之后在每天的雨结束时Soyo会把当天的雨势与前一天对比，如果当天的雨势相较前一天变小了，她的期待值就会上升，反之则会下降，上升或下降的数值等同于两天雨势差值的绝对值，即在第 i ($1 < i \leq n$) 天的大雨结束时Soyo的期待值变化的数值等于 $|a_i - a_{i-1}|$ 。

Tomori 在大雨降临之前知道了这场大雨的信息，并被给予了交换两天雨势大小的能力，她想使用至多一次交换使得在第 n 天的大雨结束时Soyo的期待值最大来迎接久违的晴天，请你输出这个最大值。

Input

一行一个正整数 t ($1 \leq t \leq 10^4$)，表示有 t 组数据。

每组数据的第一行有一个正整数 n ($1 \leq n \leq 10^5$)，表示下雨的天数。

每组数据的第二行有 n 个正整数，其中第 i 个正整数 a_i ($1 \leq a_i \leq 10^9$) 表示第 i 天的雨势大小。

保证所有数据中 n 的总和不超过 10^5 。

Output

对于每组数据输出一行

每行一个整数，表示Soyo期待值的最大值。

Example

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

Problem J. 炫耀快乐

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

n 个学生排成一行，总共有 m 个幸运数字，第 i 个小朋友获得一个幸运数字 S_i ($1 \leq S_i \leq m$)。幸运数字可以用来兑换礼物，你作为老师，你可以决定哪个幸运数字代表一等奖，哪个代表二等奖，依此类推。安排完成后，你会得到一个大小为 m 的排列 T ，表示幸运数字 i 代表 T_i 等奖。在所有礼物分发给小朋友后，他们会和邻近的小朋友，互相炫耀获得快乐。对于第 i 个小朋友和第 $i - 1$ 个小朋友 ($i > 1$)，如果第 i 个小朋友的礼物等级严格大于第 $i - 1$ 个小朋友的礼物等级，第 i 个小朋友将获得 $k_2(T_{S_i} + T_{S_{i-1}})$ 的快乐；否则，第 $i - 1$ 个小朋友将获得 $k_1(T_{S_i} - T_{S_{i-1}})$ 的快乐。你作为老师，不希望小朋友通过这种幼稚的方式获得快乐，因此你希望通过决定排列 T 来最小化小朋友们的总快乐值。

形式化地：

给定 n, m, k_1, k_2 和长度为 n 的数组 S ，请生成一个长度为 m 的排列 T ，记 i 在排列中排在第 T_i 位，使得以下值最小：

$$\sum_{i=2}^n \begin{cases} k_1(T_{S_i} - T_{S_{i-1}}) & T_{S_i} \geq T_{S_{i-1}}, \\ k_2(T_{S_i} + T_{S_{i-1}}) & T_{S_i} < T_{S_{i-1}}. \end{cases}$$

请输出这个最小值。

Input

第一行三个正整数 n ($2 \leq n \leq 10^5$)， m ($1 \leq m \leq 20$)， k_1 ($1 \leq k_1 \leq 100$)， k_2 ($1 \leq k_2 \leq 100$)，表示学生人数、奖品等级数、比例系数（含义见上文）。

第二行 n 个正整数表示学生抽到的幸运数字 S_i ($1 \leq S_i \leq m$)。

Output

最小的总快乐值。

Examples

standard input	standard output
3 3 1 1 1 2 3	2
4 3 1 1 1 2 3 1	6
12 10 22 33 1 1 4 5 1 4 1 9 1 9 8 10	638

Note

样例一最佳排列：1 2 3

$$k_1 \times (T_2 - T_1) + k_1 \times (T_3 - T_2) = 1 \times 1 + 1 \times 1 = 2$$

样例二最佳排列：1 2 3

$$k_1 \times (T_2 - T_1) + k_1 \times (T_3 - T_2) + k_2 \times (T_1 + T_3) = 1 \times 1 + 1 \times 1 + 1 \times (1 + 3) = 6$$

Problem K. agKc 与太阳甩在身后

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

继在梦中完成了"羊之主"的游戏后，agKc 又在梦中见到了祖拜尔，于是一场新的游戏开始了。

这一次，他再次来到了那块 $N \times M$ 的场地上，身边出现了若干被强化了的 "结晶消除桩"。

场地上除了 (X, Y) 地块，其他地块均被结晶覆盖。agKc 可以在未被结晶覆盖的地块上放置 "结晶消除桩"，并选择横向或竖向放置，然后 "结晶消除桩" 会消除掉一行或一列地块上的所有结晶。

agKc 可以在同一地块上多次放置"结晶消除桩"。

与之前相同，agKc 再次将作战记录交给了你，并请你告诉他最少需要放置多少个"结晶消除桩"才能消除掉场地上的所有结晶。

Input

第一行一个正整数 t ($1 \leq t \leq 10^5$) 表示测试点数量。每个测试点输入格式如下

一行四个正整数 N, M, X, Y ($1 \leq N, M \leq 10^9$, $1 \leq X \leq N$, $1 \leq Y \leq M$)

Output

共 t 行

每行一个正整数，代表 agKc 需要放置"结晶消除桩"的最少量。

Example

standard input	standard output
4	6
5 5 3 3	3
2 4 1 1	5
4 5 2 4	12
11 45 1 4	

Problem L. 新生赛与邪恶计划

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

n 名考生参加新生赛。比赛的考号与座位号为 1 到 n 的整数，每位考生应坐在与其考号对应的座位上。新生赛有独特的考场规则：考生们应当按考号从小到大依次进场，如果进场时发现自己考号的座位被占，考生需要进行吉列的豆蒸：等概率随机选择一个空座位坐下。

但现在 m 名老生混入考场展开他们的邪恶计划：他们每个人会事先为自己分配一个互不相同的座位号（与该老生的考号不一定相同），并计划在所有考生入场前进入考场，抢先坐在自己事先被分配到的座位上。为了使考场秩序更加混乱，每名老生都有独立的 50% 概率退出计划。所有退出计划的老生与剩下的 $(n - m)$ 名新生将会按照原有的新生赛考场规则入场。

现在给出 n ， m ，以及 m 位老生的考号与其被分配到的座位号，问：考号为 n 的新生最终坐在 n 号座位上的概率在模 998244353 意义下是多少？（保证 n 号一定为新生）

Input

第一行包含两个整数 n ， m ($0 \leq m < n \leq 3 \cdot 10^5$)，其含义如题目所述。

接下来 m 行，每行包含两个正整数 c_i ($1 \leq c_i < n$)， s_i ($1 \leq s_i \leq n$)，表示第 i 个老生的考号与其被分配的座位号。保证 c_i 互不相同， s_i 互不相同。

Output

输出一个整数，表示 n 号新生坐在 n 号座位模 998244353 意义下的概率。

Example

standard input	standard output
5 4 1 2 2 3 3 1 4 5	686292993

Note

样例输出为模意义下的 0.3125

Problem M. 战争游戏 II

Input file: standard input
Output file: standard output
Time limit: 8 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 m 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”，接着，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。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）。现在，他们对你提出了 q 个询问，每个询问形如：若当前游戏的 $m = a_i$ ， $s = b_i$ ， $r_1 = c_i$ ， $r_2 = d_i$ ，那么谁将获胜？

Input

本题包含多组测试数据。

输入第一行一个整数 T ，表示测试数据组数 ($1 \leq T \leq 10^6$)

每组测试数据，第一行一个整数 n ($1 \leq n \leq 10^6$)

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 ($1 \leq u_i, v_i \leq n$)

之后一行一个整数 q ，表示询问次数 ($1 \leq q \leq 10^6$)

之后 q 行，每行四个整数 a_i ， b_i ， c_i ， d_i ，表示一次询问 ($1 \leq a_i \leq 10^9, 1 \leq b_i, c_i, d_i \leq n$)

数据保证 $\sum n \leq 10^6$ ， $\sum q \leq 10^6$

Output

对于每组测试数据的每个询问，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。不同询问的答案之间需换行，不同测试数据的答案之间需换行。

Example

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

Note

本题输入输出量较大，请采用较快的输入输出方式。

你可以在你的代码中加入以下代码关闭 cin/cout 的同步流，使你的 cin/cout 的输入输出效率更高。

```
ios::sync_with_stdio(false); cin.tie(nullptr); cout.tie(nullptr);
```

请注意：加入以上代码后，你的程序中不能再使用 scanf/printf 等语句，而且你的程序中不能使用 endl 来换行（请使用 \n 来换行），不然可能会导致你的程序接受到 Wrong Answer 或者 Time Limit Exceeded 的评测结果。

Problem N. 虚拟换乘

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 512 megabytes

S 国是一个繁荣发达的大国，而刚刚高考结束的你，顺利考上了 S 国最知名的大学。经过了一个紧张刺激的暑假后，从拿到录取通知书那一刻起你就在期待的日子即将来临。你早早申请好了 S 国的签证，在 9 月 1 日这天顺利入境 S 国。现在，离你向往的大学生活只差最后一个需要解决的问题：交通出行。

S 国是一个具有极大纵深的国家，它的交通网络是一条单向的链，其中只有相邻的站点存在路径连接。在 S 国中一共只有 K 种不同的交通方式，但对于每条路径，可以选用的交通方式集合一定是这 K 种交通方式的一个非空子集。

我们可以通过以下的方式来描述交通方式：

对于编号为 $i - 1$ 的交通方式，它的速度为 $\frac{1}{rv_{i-1}}$ m/s，起步价为 w_{i-1} ，每行进 1 m 的路程需要收费 $cost_{i-1}$ 。

为了方便大家在 S 国遨游，当地推出了一种称作 "虚拟换乘" 的优惠策略，如果在两次使用某一种交通方式之间只使用过一种其它的交通方式，且这期间的耗时小于 T s，则第二次乘坐该交通方式可以通过虚拟换乘免除起步价。此外，在连续使用同一种交通方式时，只需要支付一次该交通方式的起步价。请注意，换乘只能在站点进行。

为了检验你是否已经对 S 国的交通网络了如指掌，裁判会对你进行一次询问，问你从第一个站点到最后一个站点所需的最小开支。

Input

第 1 行为三个整数 n , K , T 描述路径中站点的数目，S 国交通方式的总数以及虚拟换乘的最长时间间隔。

第 2 行为 K 个整数，第 i 个整数为编号为 $i - 1$ 的交通方式的起步价 w_{i-1}

第 3 行为 K 个整数，第 i 个整数为编号为 $i - 1$ 的交通方式的速度的倒数 rv_{i-1}

第 4 行为 K 个整数，第 i 个整数为编号为 $i - 1$ 的交通方式行进 1 m 路程所需的费用 $cost_{i-1}$

第 5 行至第 $n + 3$ 行，第 i 行有 $2 + t_{i-4}$ 个整数，依次为 d_{i-4} t_{i-4} 以及 t_{i-4} 个整数，描述第 $i - 4$ 条单向路径。这条路径的长度为 d_{i-4} m，共存在 t_{i-4} 种可用的交通方式，具体可用的交通方式由后续的 t_{i-4} 个整数给出。我们认为站点的编号为 0 到 $n - 1$ 之间的整数，其中第 i 条路径从站点 $i - 1$ 出发，连向 i 。数据保证所有路径构成一条链。

Output

1 行 1 个整数，为从站点 0 到站点 $n - 1$ 所需的最小开支。

Example

standard input	standard output
3 2 1000 80094184 93188668 5526 9059 4432 8085 7572 1 1 1706 1 0	242063464

Note

$1 \leq n \leq 64$

$1 \leq K \leq 8$ ，交通方式的编号从 0 开始。

$$0 \leq T \leq 65535$$

$$0 \leq w_i \leq 10^8$$

$$1 \leq rv_i \leq 10000$$

$$0 \leq cost_i \leq 10000$$

$$0 \leq d_i \leq 10000$$

$$1 \leq t_i \leq K$$

$$1 \leq u \leq v \leq n$$