

Problem A. Sixteen

Input file: standard input
Output file: standard output

给定一个整数 x 。你的任务是构造一个方形整数矩阵，使其行列式恰好等于 x 。

更正式地，对于每个给定的整数 x ，你需要输出一个整数 n 和一个 $n \times n$ 矩阵 A ，满足：

- $1 \leq n \leq 16$;
- A 中的每个元素都是介于 -16 和 16 之间的整数（包括 16 和 -16 ）；
- $\det(A) = x$ 。

如果有多个合法答案，你可以输出其中任意一个。

Input

第一行包含一个整数 T ($1 \leq T \leq 100$) — 测试用例的数量。

接下来的 T 行中，每行包含一个整数 x ($-2^{75} < x < 2^{75}$)。

Output

对于每个测试用例，按如下格式输出一个合法构造。

首先输出一个整数 n ($1 \leq n \leq 16$) — 矩阵的大小。

然后输出 n 行。第 i 行应包含 n 个整数 $a_{i,1}, a_{i,2}, \dots, a_{i,n}$ ($-16 \leq a_{i,j} \leq 16$) — 矩阵的元素。

所打印矩阵的行列式必须等于对应的整数 x 。

如果有多个合法答案，你可以输出其中任意一个。

Example

standard input	standard output
5	1
1	1
2	2
22	0 1
-32	-2 3
138	2
	7 5
	-3 1
	3
	-1 0 2
	3 4 -2
	1 1 7
	3
	-2 12 -1
	-1 3 4
	1 7 4

Problem B. Quadratic Residue

Input file: `standard input`
Output file: `standard output`

给定一个正整数 p ，你的任务是找到三个正整数 x_1 、 x_2 和 q ，使得：

- $1 \leq x_1 < q, 1 \leq x_2 < p$;
- $x_1^2 \equiv p \pmod{q}$;
- $x_2^2 \equiv q \pmod{p}$ 。

这里， $a \equiv b \pmod{c}$ 表示 a 除以 c 的余数与 b 除以 c 的余数相同。

如果存在多个解，输出任意一组合法的整数 (x_1, x_2, q) 即可。如果不存在这样的整数，则报告无法做到。

Input

第一行包含一个整数 T ($1 \leq T \leq 10^4$)，表示测试用例的数量。

接下来的 T 行中，每行包含一个整数 p ($2 \leq p \leq 10^9$)。

Output

对于每个测试用例，输出一行：

如果存在解，输出三个整数 x_1 、 x_2 和 q ($1 \leq q \leq 10^{12}$, $1 \leq x_1 < q, 1 \leq x_2 < p$)，用空格分隔。

如果不存在解，输出单词Impossible。

Example

standard input	standard output
3	12 1 71
2	19 2 89
5	12 3 69
6	

Problem C. 重测队列

Input file: standard input
Output file: standard output

一次大型比赛结束后，评测集群还要做一次复查：把 n 个程序在同一套 m 个测试点上重新评测。每个程序都会使用同一个测试点队列。

评测机处理一个程序时，会从队列开头依次运行测试点。一旦某个测试点给出拒绝结果，评测机在运行完这个测试点后立刻停止处理这个程序。通过全部测试点的程序会运行完整个队列。

复查开始前，历史日志已经给出了每个程序在每个测试点上的表现。第 i 个程序运行第 j 个测试点需要 $d_{i,j}$ 的时间，结果要么是接受，要么是拒绝。无论结果如何，只要这个测试点被实际运行， $d_{i,j}$ 都会计入集群总耗时。

你可以在复查前任意重排这 m 个测试点。求评测完全部 n 个程序所需的最小总耗时。

Input

第一行包含两个整数 n 和 m ，分别表示程序个数和测试点个数。

接下来 n 行，第 i 行描述第 i 个程序。每行先包含 m 个整数 $d_{i,1}, d_{i,2}, \dots, d_{i,m}$ ，随后包含一个长度为 m 的字符串 s_i 。若 s_i 的第 j 个字符为A，表示第 i 个程序在第 j 个测试点上的结果是接受；若为R，表示结果是拒绝。

保证 $1 \leq n \leq 2 \times 10^4$ ， $1 \leq m \leq 20$ ， $1 \leq d_{i,j} \leq 10^9$ ，且 s_i 仅包含字符A和R。

Output

输出一个整数，表示最小可能的总耗时。

Example

standard input	standard output
3 3 3 10 2 ARA 5 1 4 RAA 2 2 7 AAR	27

Note

一种最优顺序是先运行第2个测试点，再运行第1个测试点，最后运行第3个测试点。

在这个顺序下，三个程序的耗时分别为 10 、 $1 + 5 = 6$ 、 $2 + 2 + 7 = 11$ ，总耗时为 27 。

Problem D. The Game

Input file: **standard input**
Output file: **standard output**

Alice 和Bob 正在玩一个游戏。

初始时，有一个空序列。他们会得到一个整数 n ，并共同构造一个长度为 n 的排列 p 。

Alice 先手。每一步中，当前玩家需要从1 到 n 中选择一个此前尚未被选择过的整数，并将其添加到序列末尾。恰好进行 n 步后，该序列将成为 $1, 2, \dots, n$ 的一个排列 p 。

对于排列 $p = (p_1, p_2, \dots, p_n)$ ，它的一个循环移位是指：选择一个下标 i ($1 \leq i \leq n$)，并得到序列 $(p_i, p_{i+1}, \dots, p_n, p_1, p_2, \dots, p_{i-1})$ 。

例如，排列 $(2, 3, 1)$ 的所有循环移位为 $(2, 3, 1)$ 、 $(3, 1, 2)$ 和 $(1, 2, 3)$ 。

对于一个排列 p ，定义 $f(p)$ 为 p 的所有循环移位中字典序最小的一个。

Alice 希望让 $f(p)$ 的字典序尽可能小，而Bob 希望让它的字典序尽可能大。

假设双方都采取最优策略，请求出最终得到的排列 $f(p)$ 。

Input

第一行包含一个整数 T ($1 \leq T \leq 10^5$)，表示测试用例的数量。

接下来 T 行，每行包含一个整数 n ($1 \leq n \leq 5 \cdot 10^5$)。

保证所有测试用例中 n 的总和不超过 $5 \cdot 10^5$ 。

Output

对于每个测试用例，输出 n 个整数，表示双方均采取最优策略时最终得到的排列 $f(p)$ 。

Example

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

Problem E. DPRS

Input file: standard input
Output file: standard output

给定一个强连通的带权有向图，图中有 n 个顶点和 m 条有向边。顶点编号从1到 n ，边编号从1到 m 。

第 k 条边记为 (u_k, v_k, w_k) ，表示存在一条从 u_k 到 v_k 的有向边，权值为 w_k 。

设 $dis(i, j)$ 表示从顶点 i 到顶点 j 的最短路长度。

对于一个图，定义它的价值为

$$\max_{\substack{1 \leq i, j \leq n \\ i \neq j}} \frac{dis(i, j)}{dis(j, i)}.$$

有 q 个询问。每个询问给出一条边的编号 k 和一个新的权值 x ，其中 $1 \leq x < w_k$ 。对于这个询问，仅在当前询问中，第 k 条边的权值临时改为 x 。你需要输出在这次临时修改后图的价值。

所有询问彼此独立。也就是说，每个询问结束后，图都会恢复为原始状态。

Input

第一行包含一个整数 T ($1 \leq T \leq 1000$)，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, q ($2 \leq n \leq m \leq 2000, 1 \leq q \leq 2000$)。

所有测试用例中， n 的总和、 m 的总和以及 q 的总和都不超过2000。也就是说， $\sum n \leq 2000$ ， $\sum m \leq 2000$ ，且 $\sum q \leq 2000$ 。

接下来 m 行，每行包含三个整数 u_k, v_k, w_k ，描述第 k 条有向边，其中 $1 \leq u_k, v_k \leq n, u_k \neq v_k$ ，且 $1 \leq w_k \leq 10^9$ 。

保证每个测试用例中的图都是强连通的。

接下来 q 行，每行包含两个整数 k, x ，描述一个询问，其中 $1 \leq k \leq m$ 且 $1 \leq x < w_k$ 。

在这个询问中，第 k 条边的权值会被临时改为 x 。

Output

对于每个询问，输出一行一个实数，表示所有有序点对 (i, j) ($i \neq j$) 中 $\frac{dis(i, j)}{dis(j, i)}$ 的最大值。

如果你的答案的绝对误差或相对误差不超过 10^{-6} ，则会被判定为正确。

Example

standard input	standard output
2	6.000000000000
3 3 2	7.000000000000
1 2 2	6.000000000000
2 3 3	3.750000000000
3 1 4	3.600000000000
2 1	7.500000000000
1 1	
5 7 4	
1 2 7	
1 4 3	
2 3 4	
4 1 2	
1 5 5	
5 2 6	
3 1 8	
5 3	
6 5	
1 4	
3 2	

Problem F. 23 子序列

Input file: standard input
Output file: standard output

定义一个序列 $b = (b_1, b_2, \dots, b_m)$ 是好的, 当且仅当对于每一个 $i = 2, 3, \dots, m$, 都有

$$2 \cdot b_{i-1} \leq b_i \leq 3 \cdot b_{i-1}.$$

特别地, 长度为1 的序列总是好的。

现在给定一个长度为 n 的正整数序列 $a = (a_1, a_2, \dots, a_n)$ 。你需要回答 q 次询问, 每次询问给出一个区间 $[l, r]$ ($1 \leq l \leq r \leq n$), 请找出在子序列 $a_l, a_{l+1}, \dots, a_r$ 中 (保持原顺序), 最长的好的子序列的长度。

注意: 子序列不一定是连续的, 但必须保持原序列中的相对顺序。

Input

第一行两个整数 n, q , 分别表示序列长度和询问次数。

第二行 n 个整数 $a_1, a_2, \dots, a_n$ 。

接下来 q 行, 每行两个整数 l, r , 表示一次询问的区间。

Output

输出 q 行, 每行一个整数, 依次为每个询问的答案。

Example

standard input	standard output
5 4	3
1 2 4 6 17	2
1 3	4
2 4	3
1 5	
1 4	

Note

$1 \leq n, q \leq 2 \times 10^5$, $1 \leq a_i \leq 10^{18}$, 对于每次询问, $1 \leq l \leq r \leq n$ 。

Problem G. 消逝的回忆

Input file: `standard input`
 Output file: `standard output`

Please remember that I was once here—and then, slowly, let the memory fade.
 But look: those words that lingered unspoken, those gestures still suspended in midair, have all remained here, as though they were never truly brought to an end. The wind will still pass through this street, stirring the hem of your clothes, and carrying back some distant fragment of the past—
 If everything is destined to fade away, could we walk this road once more?

— Anonymous

给定一个长度为 n 的序列 b 。

定义一个区间 $[l, r]$ 是好的，当且仅当对于所有 $l \leq i \leq r$ ，满足

$$\min(b_l, b_r) \leq b_i \leq \max(b_l, b_r).$$

你可以进行任意次如下操作：

选择一个好的区间 $[l, r]$ ，然后将下标在 $l + 1$ 到 $r - 1$ 之间的元素全部删除。删除后序列长度缩短，原先 l 与 r 变为相邻元素，后续的操作都在新的序列上进行。

你的目标是通过最优的操作顺序，使最终序列的长度尽可能小。这个最小值被称为该序列的消除后长度。

现在你需要处理 q 个事件，每个事件是以下两种之一：

- $1 \ x \ y$ —— 将 b_x 修改为 y ；
- $2 \ L \ R$ —— 令初始序列为子段 $b[L \dots R]$ ，询问其消除后长度。

请你输出所有询问的答案。

Input

第一行两个整数 n, q 。

第二行 n 个整数 $b_1, b_2, \dots, b_n$ 。

接下来 q 行，每行第一个整数 opt 表示操作类型：

- 若 $opt = 1$ ，后跟两个整数 x, y ；
- 若 $opt = 2$ ，后跟两个整数 L, R 。

Output

对于每个 $opt = 2$ 的询问，输出一行一个整数，表示该子段的最优消除后长度。

Examples

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

Note

$1 \leq n, q \leq 2 \times 10^5$, $1 \leq b_i, y \leq 10^9$, $1 \leq x \leq n$, $1 \leq L \leq R \leq n$.

Problem H. String

Input file: **standard input**
Output file: **standard output**

给定 n 个字符串 $s_1, s_2, \dots, s_n$ 。

你可以以任意方式重新排列这 n 个字符串，然后按该顺序将它们连接起来，得到一个字符串 S 。

对于从 0 到 L 的每个整数 k （其中 $L = \sum_{i=1}^n |s_i|$ ），你可以独立地以任意方式重新排列这 n 个字符串，按该顺序连接得到字符串 S ，然后最多修改 S 中的 k 个字符。一次修改中，你可以把 S 中的一个字符替换成另一个小写英文字母。

你的任务是，对于每个 k ，找出能够得到的字典序最小的字符串。

Input

第一行包含一个整数 T （ $1 \leq T \leq 500$ ），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 （ $1 \leq n \leq 500$ ）。

第二行包含 n 个非空字符串 $s_1, s_2, \dots, s_n$ ，字符串之间用空格分隔。每个字符串只包含小写英文字母。

对于每个测试用例，设 $L = \sum_{i=1}^n |s_i|$ 。

保证所有测试用例中 n 的总和不超过 500，并且所有测试用例中 L 的总和不超过 500。

Output

对于每个测试用例，输出 $L + 1$ 行。

第 k 行（ $0 \leq k \leq L$ ）输出在最多修改 k 个字符后能够得到的字典序最小字符串。

每个测试用例的答案必须按 k 递增的顺序输出，即从 0 到 L 。

Example

standard input	standard output
3	aabbcaczz
5	aaabbcazz
bca a zz ab c	aaaaabczz
4	aaaaaabzz
ba b aa aba	aaaaaaabc
3	aaaaaaaab
az za m	aaaaaaaaa
	aaaaaaaaa
	aaaaaaaaa
	aaaaaaaaa
	aaababab
	aaaaabab
	aaaaaaaab
	aaaaaaaaa
	aaaaaaaaa
	aaaaaaaaa
	aaaaaaaaa
	azmza
	aaazm
	aaaam
	aaaaa
	aaaaa
	aaaaa

Problem I. Rounddog II

Input file: **standard input**
 Output file: **standard output**

你是Rounddog 的狂热粉丝，所以你要出一道关于Rounddog 的题目。

设 $T_1 = \text{Rounddog}$ 。

对于每个整数 $i > 1$ ，定义 $T_i = T_{i-1} + g$ ，其中 $+$ 表示字符串拼接。

换句话说， T_i 是字符串Rounddo 后面紧跟着恰好 i 个字符 g 。

给定一个字符串 S 和一个整数 k 。考虑 S 的所有循环移位。

字符串 S 的一个循环移位是通过选择一个位置，并将该位置之前的前缀移动到字符串末尾得到的。例如， $abcd$ 的循环移位有 $abcd$ 、 $bcda$ 、 $cdab$ 和 $dabc$ 。

你的任务是统计有多少个 S 的循环移位包含 T_k 作为一个连续子串。

Input

第一行包含一个整数 t ($1 \leq t \leq 10^5$)，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 ($1 \leq |S| \leq 10^5$)。

第二行包含一个整数 k ($1 \leq k \leq 100$)。

保证所有测试用例中 $|S|$ 的总和不超过 10^5 ，并且 S 只包含大写字母和小写字母。

Output

对于每个测试用例，输出一个整数，表示包含 T_k 作为连续子串的循环移位数量。

Example

standard input	standard output
3	12
Rounddoggggggggggggggggg	8
6	0
RounddogRounddo	
1	
abcdefgabcdegf	
2	

Problem J. Walk

Input file: standard input
Output file: standard output

给定一个包含 n 个顶点和 m 条边的简单连通无向图，以及一个特殊顶点序列 $x_1, x_2, \dots, x_L$ 。

你希望从 x_1 出发，通过一个类似DFS的过程，依次访问 $x_1, x_2, \dots, x_L$ 。你可以额外访问其他顶点，但必须按照给定顺序访问 $x_1, x_2, \dots, x_L$ 。

该过程由一个栈定义。初始时，栈中只有 x_1 ，并认为 x_1 已被访问。每次操作中，你必须恰好执行以下两种操作之一：

- 设 u 为当前栈顶。选择图中一个与 u 相邻的顶点 v ，并将 v 压入栈中。此时认为顶点 v 被访问。
- 弹出当前栈顶顶点。如果弹出后栈非空，则认为新的栈顶顶点被访问。

最终栈中的内容不作要求。

在所有以最少操作次数依次访问 $x_1, x_2, \dots, x_L$ 的合法过程中，求过程中出现过的最大栈大小的最小可能值。

如果序列 x 中有若干个连续且相同的顶点，那么在第一次访问该顶点时，就认为这些连续出现的顶点均已被访问。

可以证明，最少操作次数为 $\sum_{i=1}^{L-1} \text{dist}(x_i, x_{i+1})$ ，其中 $\text{dist}(a, b)$ 表示顶点 a 与顶点 b 之间最短路径的长度。

Input

第一行包含一个整数 T ($1 \leq T \leq 60$)，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 ($2 \leq n \leq 200$)、 m ($n-1 \leq m \leq \frac{n(n-1)}{2}$)和 L ($1 \leq L \leq 60$)，分别表示顶点数、边数以及特殊顶点序列的长度。

接下来 m 行，每行包含两个整数 u ($1 \leq u \leq n$)和 v ($1 \leq v \leq n$)，表示顶点 u 和顶点 v 之间存在一条无向边。

最后一行包含 L 个整数 $x_1, x_2, \dots, x_L$ ($1 \leq x_i \leq n$)。

保证每个给定的图均为简单连通图。

同时保证所有测试用例中 $\sum n \leq 200$ ，且 $\sum L \leq 60$ 。

Output

对于每个测试用例，输出一个整数，表示在所有使用最少操作次数的合法过程中，最大栈大小的最小可能值。

Example

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

Problem K. Decomposition Trees

Input file: **standard input**
Output file: **standard output**

给定一棵有 n 个顶点的树 T ，顶点编号从 1 到 n 。

我们递归地定义 T 的一种点分树 如下。

对于 T 的任意非空连通诱导子图 S ，任选一个顶点 $c \in S$ 作为当前分治中心。顶点 c 成为 S 的点分树的根。然后从 S 中删除 c 。剩余的每个连通块都按照同样的规则独立分解，它们的点分树根都成为 c 的子节点。

不同于通常的点分治，这里选取的分治中心 c 不 必须是重心。当前连通块中的任意顶点都可以被选作中心。

如果两棵点分树作为同一顶点集合 $\{1, 2, \dots, n\}$ 上的有根树，存在至少一个顶点的父亲不同，则认为它们不同。特别地，根不同的点分树一定不同。

你的任务是统计从 T 中能够得到的不同点分树的数量。

由于答案可能非常大，请输出对 $10^9 + 7$ 取模后的结果。

Input

第一行包含一个整数 n ($1 \leq n \leq 3000$)。

接下来的 $n - 1$ 行中，每行包含两个整数 u 和 v ($1 \leq u, v \leq n$)，表示顶点 u 和 v 之间有一条边。

保证给定图是一棵树。

Output

输出一个整数： T 的不同点分树数量，对 $10^9 + 7$ 取模。

Examples

standard input	standard output
3 1 2 2 3	5
5 1 2 1 3 2 4 2 5	51
7 1 2 1 4 2 3 2 5 4 6 5 7	605

Problem L. Geometry

Input file: **standard input**
 Output file: **standard output**

给定平面上的一个凸多边形 A 。保证原点 $(0,0)$ 严格位于 A 的内部。

另外给定 m 个有价值的点。第 i 个点为 $p_i = (x_i, y_i)$ ，其价值为正整数 a_i 。

对于一个实数 $\lambda \geq 0$ ，定义 $\lambda A = \{\lambda p : p \in A\}$ ，也就是说， A 中每个点的每个坐标都相对于原点乘以 λ 。

你需要找到最小的实数 $\lambda \geq 0$ ，使得可以画出一个闭圆盘 C ，满足以下两个条件：

- $C \subseteq \lambda A$;
- C 中包含的给定点的总价值至少为 W 。

闭圆盘定义为形如 $C = \{p \in \mathbb{R}^2 : \|p - o\|_2 \leq r\}$ 的集合，其中 o 是圆盘中心， $r \geq 0$ 是半径。

位于圆盘边界上的点也视为被圆盘包含。圆盘也允许与 λA 的边界相切。

保证 $1 \leq W \leq \sum_{i=1}^m a_i$ ，因此答案一定存在且有限。

Input

第一行包含一个整数 T ($1 \leq T \leq 100$)，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 ($3 \leq n \leq 1000$)、 m ($1 \leq m \leq 1000$) 和 W ($1 \leq W \leq \sum_{i=1}^m a_i$)。

接下来的 n 行描述凸多边形 A 。其中第 j 行包含两个整数 X_j 和 Y_j ($|X_j|, |Y_j| \leq 10^6$)，表示 A 的第 j 个顶点坐标。

多边形 A 的顶点按逆时针顺序给出。该多边形是凸的，且原点严格位于其内部。

接下来的 m 行描述关键点。第 i 行包含三个整数 x_i 、 y_i 和 a_i ($|x_i|, |y_i| \leq 10^6$, $1 \leq a_i \leq 10^9$)，表示点 (x_i, y_i) 的价值为 a_i 。

保证所有测试用例中 n 的总和不超过1000，所有测试用例中 m 的总和不超过1000。

所有输入坐标和价值均为整数。 W 的值可以用有符号64 位整数表示。

Output

对于每个测试用例，输出一个实数： λ 的最小可能值。

如果你的答案的绝对误差或相对误差不超过 10^{-6} ，则会被判定为正确。

更准确地说，若你的输出为 x ，正确答案为 y ，当 $|x - y| \leq 10^{-6} \cdot \max(1, |y|)$ 时，你的答案将被接受。

Example

standard input	standard output
3	0.0000000000
4 1 5	1.0000000000
-1 -1	2.0000000000
1 -1	
1 1	
-1 1	
0 0 5	
4 2 2	
-1 -1	
1 -1	
1 1	
-1 1	
1 0 1	
0 1 1	
4 2 2	
-1 -1	
1 -1	
1 1	
-1 1	
-2 0 1	
2 0 1	