

Problem A. 方阵

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

有两个 $n \times n$ ($1 \leq n \leq 300$) 的方阵 A, B ，其中每一行每一列都恰好包含了 $1 \sim n$ 。现在你可以交换 A 的任意两行或两列，问经过若干次操作之后，能否将 A 变成 B ？如果可以输出最少的操作次数，否则输出 -1 。

Input

第一行输入一个正整数 n 。

接下来 n 行，每行 n 个元素，表示方阵 A 。

接下来 n 行，每行 n 个元素，表示方阵 B 。

Output

输出一个整数，表示答案。

Example

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

Problem B. 覆盖

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

有一个 n 个点的树，你想要覆盖树上的所有边恰好一次。你有两种操作：

1. 选择两个点 u, v ，将 u 到 v 路径上的每一条边都覆盖一次。
2. 选择一个点 u ，将与 u 直接相邻的每一条边都覆盖一次。

问最少的操作次数可以完成目标。注意每条边不能重复覆盖，也就是如果执行操作的时候要保证对应的边都是空的。

Input

第一行一个整数 n ($2 \leq n \leq 10^5$)。

接下来 $n - 1$ 行，每行两个整数 u, v ($1 \leq u, v \leq n$)。

Output

一个整数，表示答案。

Example

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

Problem C. 负无穷大

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

你有一个长度为 n ($1 \leq n \leq 10^5$) 的数组 $a_1, a_2, \dots, a_n$ ($|a_i| \leq 1 \cdot 10^4$) 和一个整数 k ($|k| \leq 10^9$)。要执行 n 次操作，每次操作会将 a_{p_i} 修改为 -10^{10} ，保证每次修改的位置都是不同的。在每次操作之前，你想知道所有的区间里面，权值和与 k 最接近的是多少。也就是你要找到一个区间 $[i, j]$ ($1 \leq i \leq j \leq n$)，使得 $\left| \sum_{t=i}^j a_t - k \right|$ 最小，输出这个最小的绝对值。

Input

第一行两个整数 n, k 。

接下来一行 n 个整数 $a_1, a_2, \dots, a_n$ 。

接下来一行 n 个整数 $p_1, p_2, \dots, p_n$ 。

Output

输出 n 行，每行一个数表示答案。

Example

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

Problem D. 生成树

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

有一个 n ($1 \leq n \leq 50$) 个点的带权无向图，其中 (u,v) 的权值为 $w_{u,v}$ ($0 \leq w_{i,j} \leq 10^9, w_{i,j} = w_{j,i}, w_{i,i} = 0$)。我们定义生成树的权值为边权之和的 k ($0 \leq k \leq 50$) 次方。问所有生成树的权值之和，对 $10^9 + 7$ 取模。

Input

第一行，两个整数 n, k 。

接下来 n 行，每行 n 个整数，表示 w 。

Output

输出一个数，表示答案。

Example

standard input	standard output
3 1 0 1 2 1 0 3 2 3 0	12

Problem E. 评估

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

小明同学是一家科技公司数据分析部门的员工。一天，他获取到了一组长度为 n ($n \leq 1 \times 10^5$) 的整数数列 a_i ($|a_i| \leq 1000$)，这个数列代表着每个时间段产品的性能数据。为了更深入地了解产品性能的波动情况，他需要计算 $\sum_{i=1}^{n-1} \sum_{j=i+1}^n |a_i - a_j|^2$ 来评估整体的差异程度（数列从 1 开始编号）。

但小明同学并不想去计算，于是他想请你帮忙。

Input

输入的第一行包含一个正整数 n ，表示数列的长度。

输入的第二行包含 n 个整数 a_i ，表示每个时间段产品的性能数据。

Output

输出共一行，包含一个整数，表示数列整体的差异程度。

Examples

standard input	standard output
3 2 8 4	56
5 -5 8 9 -4 -3	950

Note

样例1: $|2 - 8|^2 + |2 - 4|^2 + |8 - 4|^2 = 36 + 4 + 16 = 56$ 。

Problem F. 拆分数字

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

小明同学探索到一个古老的数学遗迹，在遗迹的深处发现了若干到道神秘的谜题。谜题中给出了整数 n 和 k ($1 \leq k \leq n \leq 1 \cdot 10^9$)，并有如下提示："在这个神秘的地方，存在着一类特殊的数字，它们的形式为 3^m (m 是非负整数)。现在需要判断能否通过恰好 k 个这样的特殊数字相加，得到整数 n 。

换言之，是否存在一个非负整数序列 $\{a_k\}$ ，使得 $n = 3^{a_1} + 3^{a_2} + \dots + 3^{a_k}$ 。

Input

输入的第一行包含一个正整数 T ($1 \leq T \leq 1 \cdot 10^5$)，表示谜题的个数。

接下来 T 行，每行两个整数 n, k ，表示一道谜题中的信息。

Output

输出共 T 行。对于每一道谜题，如果可以则输出 'Yes'，否则输出 'No'。你可以输出任意类型的Yes或者No。

Example

standard input	standard output
4	Yes
5 3	No
17 2	Yes
163 79	Yes
1000000000000000000 10000000000000000000	

Note

对于第一个测试案例， $5 = 3^1 + 3^0 + 3^0$ ，因此满足了相关条件。

对于第二个测试案例，没有非负整数序列 a_1, a_2 使得 $17 = 3^{a_1} + 3^{a_2}$ ，因此不满足有关条件。

Problem G. 露营

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

小明和他的两位好朋友要在野外露营，经过认真的规划后，他们将整片地区划分为了若干个由 1×1 的小方格共同组成的平面直角坐标系，直角坐标系的坐标表示为 $x\ y (1 \leq x, y \leq 1000)$ 。

经过一番商议后，他们决定在 A, B, C 三点出安营扎寨（ A, B, C 各占据了一个 1×1 的小方格，且相互之间位置不同）。由于野外有许多枯枝败叶，他们希望能够清理一些小方格，以便他们相互拜访。

具体来说，开始时，所有的小方格都被未被清理（包括 A, B, C 三个点）。他们可以通过相邻的清理之后的小方格，从其他方格到达 $A\Delta B\Delta C$ 中的任何一个。如果两个小方格有共同边，则它们相邻。

因此，请你帮忙计算一下，他们最少需要清理的小方格数量有多少？

Input

输入三行，每行各两个整数，分别表示三人安营扎寨的位置。

Output

输出共一行，表示最少需要清理的小方格数量。

Examples

standard input	standard output
0 0 1 1 2 2	5
0 0 2 0 1 1	4

Note

对于样例1，一种最优方案为清理 $(0, 0), (0, 1), (1, 1), (1, 2), (2, 2)$ 。

Problem H. 寻宝

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

小明意外的获得了一张神奇的藏宝图，但想要获得藏宝图中隐藏着的秘密宝藏的地址，就需要解开藏宝图中所隐藏的秘密。

藏宝图上标记着一个 $n \times n$ ($1 \leq n \leq 1000$) 大小的表格，表格中填满了各式各样的小写字母，在探索解密的过程中，小明可以释放法术，随意修改表格中任意最多个格子中的字母，将其变为其他任意一个小写字母。当然，小明的法术也是有限制的，他最多可以修改 k ($1 \leq k \leq n^2$) 个格子上的字母。

想要得到藏宝图中的秘密，小明需要首先从表格的左上角开始移动，最终移动到表格的右下角为止，每一步可以往右或往下移动一格。这样，从表格的左上角移动到右下角，就一共会经过 $2n - 1$ 个格子，而这 $2n - 1$ 个格子中的小写字母将组成一段神奇的文字。而最终秘密宝藏的隐藏地点，就是最多修改个字母后，字典序最小的这一段神奇文字所描述的地方。

现在，请你帮助小明找到秘密宝藏被隐藏的地点。

Input

第一行输入两个数字 n k ，表示藏宝图上表格的大小和最多可以修改的格子数量。

接下来，输入一个由小写字母组成的 n 行 n 列的表格，表示解密需要的表格。

Output

共一行，输出一个有小写字母组成的字符串，表示字典序最小的宝藏隐藏地点。

Examples

standard input	standard output
4 2 abcd bcde bcad bcde	aaabcde
7 6 ypnxnnp pnxonpm nxanpou xnnpmud nhtdudu npmuduh pmutsnz	aaaaaadudsanz

Note

对于样例1：修改后的结果为

abcd
acde
acad
bcde

先向下后向右即可达成题设条件。

Problem I. 互质划分

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

正整数 a 和 b 的最大公因数是满足 a, b 同时是 d 的倍数的最大的正整数 d 。

而正整数 a 和 b 互质，指的是 a 和 b 的最大公因数为 1。

给定一个正整数 n ，求最少把 $1 \sim n$ ($1 \leq n \leq 1 \cdot 10^{18}$) 共 n 个正整数分成多少堆，才能使得每一堆里面每两个数都互质。

Input

一行一个正整数 n 。

Output

一行一个整数表示最少的堆数。

Examples

standard input	standard output
2	1
5	2

Note

对于样例1： 一种合法方案是把 1, 2 放在同一堆，则 1, 2 的最大公因数是 1，它们互质，所以满足要求。

对于样例2： 一种合法方案是把 1, 2, 5 放在同一堆，3, 4 放在同一堆，可以验证是满足要求的。

Problem J. 出租车

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

A 城是一个独特的城市，因为它是一条无尽的数轴。

打车软件 U 如今非常流行，其被城市中所有的 m 名出租车司机使用，他们每天运送剩下的城市居民—— $n(1 \leq n, m \leq 10^5)$ 名乘客。

A 城的每个居民（包括出租车司机）都住在一个独一无二的位置，也就是说没有两个居民的坐标是相同的。

U 的系统非常聪明：当乘客叫车时，他的呼叫不会传给所有的出租车司机，而只会传给离他最近的那个司机。如果有多个司机距离相同，那么坐标较小的司机会收到呼叫。

但是，有一天早上，出租车司机们好奇：当一个乘客是当天第一个叫车的，会有多少乘客选择给指定的出租车司机打电话？换句话说，你需要为每个出租车司机 i 找到 a_i ——当所有司机和乘客都在家时，会有多少乘客选择给第 i 名出租车司机打电话？

出租车司机不能接送自己或呼叫其他出租车司机。

Input

第一行两个正整数 n, m 。

第二行 $n + m$ 个正整数 $x_1, x_2, \dots, x_{n+m}(1 \leq x_1 < x_2 < \dots < x_{n+m} \leq 10^9)$ ，其中 x_i 表示第 i 位居民家的位置。

第三行 $n + m$ 个整数 $t_1, t_2, \dots, t_{n+m}(0 \leq t_i \leq 1)$ 表示每个居民的身份，如果 $t_i = 1$ ，那么第 i 个居民是司机，否则他是乘客。

保证 $t_i = 1$ 的 i 的数量恰为 m 。

Output

一行 m 个整数 $a_1, a_2, \dots, a_m$ ，其中 a_i 是第 i 名出租车司机的答案。家坐标第 i 小的出租车司机编号为 i 。

Examples

standard input	standard output
3 1 1 2 3 10 0 0 1 0	3
3 2 2 3 4 5 6 1 0 0 0 1	2 1
1 4 2 4 6 10 15 1 1 1 1 0	0 0 0 1

Note

对于样例1：只有一个出租车司机，这意味着所有 n 名乘客的订单都会传给他。

对于样例2：第一个出租车司机住在坐标为 2 的点，第二个出租车司机住在坐标为 6 的点。显然，住在坐标 3 的乘客最接近第一个出租车司机，而住在坐标 5 的乘客最接近的是第二个出租车司机。住在坐标 4 的乘客与第一个和第二个出租车司机的距离相同，但因为第一个出租车司机的坐标较小，所以这个乘

客的呼叫会传给第一个出租车司机。

对于样例3: 有一个乘客，离他最近的是第四个出租车司机。

Problem K. 木雕玩具

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

在一个小镇上，有一个专门从事木雕工艺的工作室。由于小镇规模不大，只有三位雕刻师在那里工作。

小镇上计划举办一个木制玩具节。工作室的员工们想要为此做好准备。

将会有 $n(1 \leq n \leq 2 \cdot 10^5)$ 个人来到工作室请求制作木制玩具。每个人都是独一无二的，他们可能想要不同的玩具。为了简化问题，让我们用 $a_i(1 \leq a_i \leq 10^9)$ 表示第 i 个人想要的玩具图案。

每位雕刻师都可以事先选择一个图案，用一个 1 到 10^9 之间的整数 x 来表示，不同的雕刻师可以选择不同的图案。在节日准备期间，雕刻师将完全掌握制作所选图案玩具的技巧，这将使他们能够立刻切割出木制玩具。对于选择了图案 x 的雕刻师来说，制作图案为 y 的玩具将需要 $|x - y|$ 的时间，因为玩具图案越接近他能立即制作的，雕刻师就越能快速完成工作。

在节日当天，当一个人来到工作室请求制作木制玩具时，雕刻师可以选择谁来接手这份工作。同时，雕刻师们都是非常熟练的人，可以同时为不同的人工作。

由于人们不喜欢等待，雕刻师们希望选择准备的图案，使得所有人的最大等待时间尽可能小，请你求出这个值。

Input

第一行包含一个整数 n 表示来到工作室的人数。

第二行包含 n 个整数 $a_{1 \sim n}$ 表示玩具的图案。

Output

一行一个整数表示答案。

Examples

standard input	standard output
6 1 7 7 9 9 9	0
6 5 4 2 1 30 60	2
9 14 19 37 59 1 4 4 98 73	13

Note

对于样例1：三位雕刻师事先选择图案 1, 7, 9。

对于样例2：三位雕刻师事先选择图案 3, 30, 60。

对于样例3：三位雕刻师事先选择图案 14, 50, 85。

Problem L. 幽默数

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

给定一个长度为 n ($1 \leq n \leq 10^5$) 的正整数序列 a 。一个正整数 x 被称为「幽默的」，当且仅当不存在一个子区间，使得其所有元素的最小公倍数等于 x 。

你需要找到最小的「幽默的」数。

序列 a 的子区间指的是序列中的一组元素 $a_l, a_{l+1}, \dots, a_r$ ，其中 $1 \leq l \leq r \leq n, 1 \leq a_i \leq 10^9$ 。我们将这样的子区间表示为 $[l, r]$ 。

Input

第一行一个整数 T ($1 \leq T \leq 10$) 表示数据组数。

对于每组数据，第一行一个整数 n ，第二行 n 个整数 $a_1 \sim a_n$ 。

Output

对于每组数据，输出一行一个整数表示答案。

Example

standard input	standard output
6	4
3	7
1 2 3	1
5	1
1 2 3 4 5	16
2	13
2 3	
1	
1000000000	
12	
1 8 4 2 3 5 7 2 9 10 11 13	
12	
7 2 5 4 2 1 1 2 3 11 8 9	

Note

对于样例1：在第一组样例数据中，4 是一个幽默数，并且是最小的，因为数组中出现了整数 1, 2, 3，这意味着存在长度为 1 的子区间，其最小公倍数分别为 1, 2, 3，并且不存在最小公倍数等于 4 的子区间。

在第二组样例数据中，7 是一个幽默数，整数 1 ~ 5 明确出现在数组中，而整数 6 是子区间 $[2, 3]$ 和 $[1, 3]$ 的最小公倍数。

在第三组样例数据中，1 是一个幽默数，因为子区间 $[1, 1], [1, 2], [2, 2]$ 的最小公倍数分别为 2, 6, 3。