Problem A. Donut

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

In this problem, we will use Chebyshev distance on a Cartesian plane. The Chebyshev distance between two points (p_x, p_y) and (q_x, q_y) on the plane is $\max\left(|p_x - q_x|, |p_y - q_y|\right)$.

Let us define a **donut** on the plane as the set of points in a certain distance range from a certain point, the donut's center. More formally, the donut with center (x_c, y_c), inner radius l and outer radius r is the set of points (x, y) such that $l \le \max\left(|x - x_c|, |y - y_c|\right) \le r$.

There are n points selected on the Cartesian plane. Points are numbered from 1 to n, and each point has a score associated with it.

We want to place a donut on the plane. The inner radius of the donut will be l, the outer radius will be r, and the center of the donut is not yet determined: you have to decide where to place it. However, both coordinates of the center must be integers. After you place the donut, its score will be the sum of the scores of points which belong to the donut.

You are given the coordinates of the selected points and the score associated with each point. Calculate the maximum possible score of the donut you can place.

Input

The first line contains three integers: n, the number of selected points on the plane, l, the inner radius of the donut, and r, the outer radius of the donut ($1 \le n \le 10^5$, $1 \le l \le r \le 10^9$).

The following n lines describes selected points, one per line. Each of these lines contains three integers x, y, and s: the coordinates of the point and the score associated with it ($-10^9 \le x, y \le 10^9$, $-10^4 \le s \le 10^4$). Note that some points may coincide.

Output

Print the maximum score of the donut.

Examples

standard input	standard output
4 1 1 0 1 1 0 -1 1 1 0 -100 -1 0 -100	2
4 1 2 0 1 1 0 -1 1 1 0 -100 -1 0 -100	1

Problem C. Earthquake

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

The Island of Sodor (or just Sodor) can be reached from England via bridges.

Because Sodor and England are far away, there exist multiple routes (exactly n routes) that connect the two lands via one or more bridges. Specifically, route i connects Sodor and England through k_i ($k_i \geq 1$) bridges. Let us denote the j-th bridge (counting from Sodor to England) of route i as $B[i, j]$. Below, we have two routes ($n = 2$) and five bridges with $k_1 = 2$ and $k_2 = 3$.

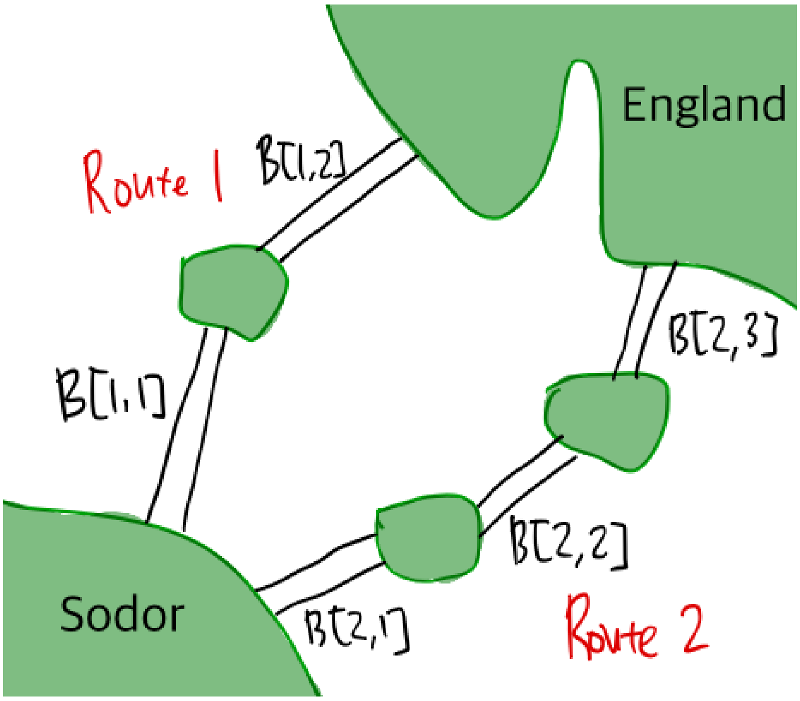

Figure 1: Two routes and five bridges

For any two adjacent bridges of a route, $B[i, j]$ and $B[i, j+1]$ where $j + 1 \leq k_i$, their joint is a small island that simply serves as a connecting point of the two bridges. As demonstrated in the figure, there are exactly n routes that connect Sodor and England, as the bridges do not intersect, and all joints are also distinct. In particular, if any one of the bridges of route i is damaged, then route i cannot be used to travel between Sodor and England.

Due to recent earthquakes in the area, some of the bridges may have experienced severe damage and become unusable. At this point we do not know exactly which bridges withstood the earthquakes and which ones were destroyed. Thanks to the inspections that were performed on the bridges prior to the incidents, for each bridge, we know the exact probability of whether it is still intact or not. Let $p[i, j]$ be the probability ($0 < p[i, j] < 1$) that bridge $B[i, j]$ is still intact after the earthquakes. Assume that the events of bridges being intact are mutually independent.

We want to know whether there is still a path between Sodor and England. However, determining whether a bridge is still intact or not is a costly operation because one needs to dispatch a large team of inspectors via choppers and boats, and therefore we want to minimize the number of inspections. Using an optimal sequence of inspections (that minimizes the expected number of inspections), what is the expected number of inspections we must perform until we are certain whether there is still a safe route between Sodor and England?

Input

The first line contains n, the number of routes ($2 \le n \le 1000$).

The following n lines contain the information on each route. The i-th of them starts with an integer k_i, the number of bridges in the i-th route ($1 \le k_i \le 1000$). Then follow k_i integers labeled as $q[i, j]$, where $1 \le j \le k_i$. Each $q[i, j]$ is an integer between 1 and 999, inclusive, and it means that $p[i, j] = q[i, j]/1000$.

Output

Simply output the expected number of inspections for an optimal inspection sequence of the bridges. Your answer will be considered correct if its relative or absolute error is within 10^{-9}.

Examples

standard input	standard output
2 3 900 900 900 2 100 100	3.0081
3 1 240 1 310 1 50	2.2144

Note

The first input describes the example discussed in the problem description. Intuitively, route 1 is very likely to be fine, as all three bridges are expected to be intact, whereas route 2 is most likely destroyed. The optimal sequence is to inspect the three bridges of route 1 (in any order, until either route 1 is declared to be safe or damaged), and then the two bridges of route 2.

For the second input, it is optimal to inspect route 2, then route 1, and then route 3 (if needed).

Problem D. Dynamic Input Tool

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

CodingSlave 1.0 is a brand new text editor. There are two possible operations to compose a string in CodingSlave 1.0:

1. Add one character at the end of the current string.
2. Choose a nonempty subsequence of the current string and add it at the end of the current string.

Initially, the current string is empty.

Given a word consisting of lowercase English letters, calculate the minimum number of operations needed to compose this string.

Input

The first line contains a string S consisting of lowercase English letters ($1 \le |S| \le 10^6$).

Output

Print one integer: the minimum number of operations needed to compose S using CodingSlave 1.0.

Examples

standard input	standard output
aaa	3
aabaaaabaa	5

Problem E. Central Lake

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

The town where Jaehyun lives is a circle with a radius of R. There are 360 000 points on the circumference, numbered in counterclockwise order: if you go counterclockwise from point 359 999, the next point is point 0. All the distances between pairs of neighboring points are equal.

Initially, there were N houses in some of the 360 000 points. Because the entire territory was originally flat, people could always go to each other's house by direct path. However, Sunghyeon, the mayor of the city, ordered to dig a lake in the center of the town to make it look good. The center of the lake and the center of the town is the same point. He also plans to demolish old houses and build new ones.

Jaehyun is worried that it will take a long time to walk between the houses because of the central lake. Your task is to calculate the maximum value of the shortest distance between two houses. Of course, every time the mayor orders to build or demolish a house, you have to recalculate the answer.

Input

The first line of input contains two space-separated integers R and r: the radius of the country and the radius of the lake, respectively ($10 \le R \le 10^5$, $1 \le r < R$).

The second line contains an integer N ($2 \le N \le 100\,000$). The third line contains N space-separated integers $a_1, a_2, \ldots, a_N$: the locations of N houses ($0 \le a_i < 360\,000$, all a_i are pairwise distinct).

The next line contains an integer Q, the number of queries ($1 \le Q \le 100\,000$).

The following Q lines describe the queries. Each of these lines contains two space-separated integers q and x ($1 \le q \le 2$, $0 \le x < 360\,000$).

Each query has one of following formats depending on its type:

- "`1` x": Build new house at point x.
- "`2` x": Demolish a house at point x.

It is guaranteed that there is no house at point x when q is `1` and there is a house at point x when q is `2`. Also, it is guaranteed that there will be at least two houses at all times.

Output

Print $Q + 1$ lines. On the first line, print the maximum distance between two houses after the lake appears, but before all queries. For the next Q lines, output the required answer after executing each query. Absolute or relative error 10^{-6} or better will be tolerated.

Example

standard input	standard output
10 5 2 0 90000 4 1 180000 1 240000 2 0 2 90000	14.14213562 22.55649583 22.55649583 19.93850195 10

Problem F. Computing MDSST

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

In this problem, we will consider weighted undirected graphs where all edges have positive weights.

Let $D(G, i, j)$ be the length of the shortest path in graph G between vertex i and vertex j.

We are given a complete weighted undirected graph G which consists of n vertices numbered from 1 to n. Among the spanning trees of G, the MDSST (Minimum Distance Sum Spanning Tree) is such T for which the value $S(T) = \sum_{1 \le i < j \le n} D(T, i, j)$ is minimum. Your task is to find MDSST of G and print $S(T)$.

Input

The first line contains an integer n, the number of vertices in the graph ($2 \le n \le 15$). The i-th of the following $n-1$ lines contains $n-i$ integers separated by spaces. The j-th integer of this the line is the length of the edge between vertex i and vertex $i+j$.

All the lengths are between 1 and 10^9, inclusive.

Output

On the first line, print one integer: the value $S(T)$ for the MDSST you found.

Example

standard input	standard output
4 3 2 1 5 6 7	18

Problem L. XOR Transformation

Input file: `standard input`
Output file: `standard output`
Time limit: 2 seconds
Memory limit: 1024 mebibytes

You are given an integer array of length N: $X = [x_0, x_1, \ldots, x_{N-1}]$.

Let us define a transformation of X, which is $F_k(X) = \left[f_{k,0}(X), f_{k,1}(X), \ldots, f_{k,N-1}(X)\right]$, as follows:

- k is an integer between 1 and N, inclusive.
- $f_{k,i}(X) = \bigoplus_{j=0}^{k-1} x_{(i+j) \bmod N}$, where i is an integer between 0 and $N-1$, inclusive, and $\oplus$ is bitwise XOR.

You are also given two integers T and K. Calculate the value $F_K^T(X)$ and print it. Note that $F_K^1(X) = F_K(X)$ and $F_K^t(X) = F_K\left(F_K^{t-1}(X)\right)$ for $t > 1$.

Input

The first line contains three integers: N, K, and T ($1 \le K \le N \le 10^5$, $1 \le T \le 10^{18}$).

The second line contains N non-negative integers $x_0, x_1, \ldots, x_{N-1}$ ($0 \le x_i \le 10^9$), where x_i is the i-th element of the array X, numbered from zero.

Output

Let $F_K^T(X) = [a_0, a_1, \ldots, a_{N-1}]$. Print the N integers $a_0, a_1, \ldots, a_{N-1}$ on the first line.

Examples

standard input	standard output
5 3 1 3 0 2 1 2	1 3 1 0 1
5 3 2 3 0 2 1 2	3 2 0 0 3
5 3 3 3 0 2 1 2	1 2 3 0 2
5 3 15 3 0 2 1 2	3 0 2 1 2
11 5 1000000000000000000 2 2 4 5 9 1 5 7 7 1 8	13 4 5 8 1 0 5 10 3 4 8

Problem M. Ants

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

When moving, ants form rows so that each ant except the first is behind another ant. It is not widely known what happens when two rows of ants moving in opposite directions run into each other in a passage too narrow for both rows to pass through. One theory says that, in that situation, ants will jump over each other. From the moment the rows meet, each second every ant jumps over (or gets jumped over, as they agree upon) the ant in front of himself so that the two ants swap places, but only if the other ant is moving in the opposite direction. Find the order of the ants after T seconds.

Input

The first line contains two integers N_1 and N_2, the numbers of ants in the first and second rows, respectively. The next two rows contain the orders of ants in the first and second row (first to last). Each ant is uniquely determined by an uppercase letter of the English alphabet (this letter is unique between both rows). The last line of input contains the integer T ($0 \leq T \leq 50$).

Output

Output the order of the ants after T seconds on a single line. Our viewpoint is such that the first row of ants comes from our left side and the other one from our right side.

Example

standard input	standard output
3 3 ABC XYZ 0	CBAXYZ
3 3 ABC XYZ 2	CXBYAZ
3 4 XLQ CRUZ 3	CQRLUXZ

Problem N. Manhattan Circle

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

In Manhattan metrics the distance between two points A (x_1, y_1) and B (x_2, y_2) is defined as:

$$D(A, B) = |x_1 - x_2| + |y_1 - y_2|$$

All other definitions are the same as in Euclidian geometry, including that of a circle:

A circle is the set of all points in a plane at a fixed distance (the radius) from a fixed point (the centre of the circle).

We are interested in calculating of area of a circle with radius R in Manhattan metrics.

Input

The first and only line of input will contain the radius R, an integer smaller than or equal to 10000.

Output

You should output the area of a circle with radius R in Manhattan metrics. Outputs within 10^{-9} of the official solution will be accepted.

Examples

standard input	standard output
1	2.0

Problem O. Polygon

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

Consider a convex polygon with N vertices, with the additional property that no three diagonals intersect in a single point. Find the number of intersections between pairs of diagonals in such a polygon.

Input

The first and only line of input contains a single integer N, $3 \leq N \leq 100$.

Output

Output the number of intersections on a single line.

Example

standard input	standard output
3	0
4	1
6	15

Problem P. Sequence

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

You are given an infinite sequence of integers:

$$A_0 = 0 \quad (1)$$

$$A_i = (A_{i-1} + q) \mod p, \text{for all } i \text{ ¿ } 0 \quad (2)$$

Where p is a prime number, $a \mod p$ is reminder of division a by p.

Your task is to count how many integers in the range [0..p-1] are not included in this sequence. Notice that p can be very large.

Input

The first line contains one prime integer p. The second line contains one integer q. ($0 \leq q \leq p \leq 10^{100}$).

Output

Output a single number - the answer for the problem.

Example

*

standard input	standard output
5 1	0

Problem Q. Songs

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

Every evening villagers in a small village gather around a big fire and sing songs. A prominent member of the community is the singer. Every evening, if the singer is present, he sings a brand new song that no villager has heard before, and no other song is sung that night. In the event that the singer is not present, other villagers sing without him and exchange all songs that they know. Given the list of villagers present for E consecutive evenings, output all villagers that know all songs sung during that period.

Input

The first line of input contains an integer N, $1 \leq N \leq 100$, the number of villagers. The villagers are numbered from 1 to N. Villager number 1 is the singer. The second line contains an integer E, $1 \leq E \leq 50$, the number of evenings. The next E lines contain the list of villagers present on each of the E evenings. Each line begins with a positive integer K, $2 \leq K \leq N$, the number of villagers present that evening, followed by K positive integers separated by spaces representing the villagers. No villager will appear twice in one night and the singer will appear at least once across all nights.

Output

Output all villagers that know all songs, including the singer, one integer per line in ascending order.

Example

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