Wunder Fund Round 2016 (Div. 1 + Div. 2 combined)

A. Slime Combining

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Your friend recently gave you some slimes for your birthday. You have n slimes all initially with value 1.

You are going to play a game with these slimes. Initially, you put a single slime by itself in a row. Then, you will add the other $n - 1$ slimes one by one. When you add a slime, you place it at the right of all already placed slimes. Then, while the last two slimes in the row have the same value v, you combine them together to create a slime with value $v + 1$.

You would like to see what the final state of the row is after you've added all n slimes. Please print the values of the slimes in the row from left to right.

Input

The first line of the input will contain a single integer, n ($1 \le n \le 100\,000$).

Output

Output a single line with k integers, where k is the number of slimes in the row after you've finished the procedure described in the problem statement. The i-th of these numbers should be the value of the i-th slime from the left.

Examples

input
1
output
1

input
2
output
2

input
3
output
2 1

input
8
output
4

Note

In the first sample, we only have a single slime with value 1. The final state of the board is just a single slime with value 1.

In the second sample, we perform the following steps:

Initially we place a single slime in a row by itself. Thus, row is initially 1.

Then, we will add another slime. The row is now 1 1. Since two rightmost slimes have the same values, we should replace these slimes with one with value 2. Thus, the final state of the board is 2.

In the third sample, after adding the first two slimes, our row is 2. After adding one more slime, the row becomes 2 1.

In the last sample, the steps look as follows:

1. 1
2. 2

3. 2 1
4. 3
5. 3 1
6. 3 2
7. 3 2 1
8. 4

B. Guess the Permutation

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Bob has a permutation of integers from 1 to n. Denote this permutation as p. The i-th element of p will be denoted as p_i. For all pairs of distinct integers i, j between 1 and n, he wrote the number $a_{i,j} = min(p_i, p_j)$. He writes $a_{i,i} = 0$ for all integer i from 1 to n.

Bob gave you all the values of $a_{i,j}$ that he wrote down. Your job is to reconstruct any permutation that could have generated these values. The input will be formed so that it is guaranteed that there is at least one solution that is consistent with the information given.

Input

The first line of the input will contain a single integer n ($2 \le n \le 50$).

The next n lines will contain the values of $a_{i,j}$. The j-th number on the i-th line will represent $a_{i,j}$. The i-th number on the i-th line will be 0. It's guaranteed that $a_{i,j} = a_{j,i}$ and there is at least one solution consistent with the information given.

Output

Print n space separated integers, which represents a permutation that could have generated these values. If there are multiple possible solutions, print any of them.

Examples

input
2 0 1 1 0
output
2 1

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

Note

In the first case, the answer can be $\{1, 2\}$ or $\{2, 1\}$.

In the second case, another possible answer is $\{2, 4, 5, 1, 3\}$.

C. Constellation

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Cat Noku has obtained a map of the night sky. On this map, he found a constellation with n stars numbered from 1 to n. For each i, the i-th star is located at coordinates (x_i, y_i). No two stars are located at the same position.

In the evening Noku is going to take a look at the night sky. He would like to find three distinct stars and form a triangle. The triangle must have positive area. In addition, all other stars must lie strictly outside of this triangle. He is having trouble finding the answer and would like your help. Your job is to find the indices of three stars that would form a triangle that satisfies all the conditions.

It is guaranteed that there is no line such that all stars lie on that line. It can be proven that if the previous condition is satisfied, there exists a solution to this problem.

Input

The first line of the input contains a single integer n ($3 \le n \le 100\,000$).

Each of the next n lines contains two integers x_i and y_i ($-10^9 \le x_i, y_i \le 10^9$).

It is guaranteed that no two stars lie at the same point, and there does not exist a line such that all stars lie on that line.

Output

Print three distinct integers on a single line — the indices of the three points that form a triangle that satisfies the conditions stated in the problem.

If there are multiple possible answers, you may print any of them.

Examples

input
3 0 1 1 0 1 1
output
1 2 3

input
5 0 0 0 2 2 0 2 2 1 1
output
1 3 5

Note

In the first sample, we can print the three indices in any order.

In the second sample, we have the following picture.

Note that the triangle formed by starts 1, 4 and 3 doesn't satisfy the conditions stated in the problem, as point 5 is not strictly outside of this triangle (it lies on it's border).

D. Hamiltonian Spanning Tree

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

A group of n cities is connected by a network of roads. There is an undirected road between every pair of cities, so there are $\frac{n \cdot (n-1)}{2}$ roads in total. It takes exactly y seconds to traverse **any** single road.

A *spanning tree* is a set of roads containing exactly $n - 1$ roads such that it's possible to travel between any two cities using only these roads.

Some spanning tree of the initial network was chosen. For every road in this tree the time one needs to traverse this road was changed from y to x seconds. Note that it's not guaranteed that x is smaller than y.

You would like to travel through all the cities using the shortest path possible. Given n, x, y and a description of the spanning tree that was chosen, find the cost of the shortest path that starts in any city, ends in any city and visits all cities **exactly once**.

Input

The first line of the input contains three integers n, x and y ($2 \le n \le 200\,000$, $1 \le x, y \le 10^9$).

Each of the next $n - 1$ lines contains a description of a road in the spanning tree. The i-th of these lines contains two integers u_i and v_i ($1 \le u_i, v_i \le n$) — indices of the cities connected by the i-th road. It is guaranteed that these roads form a spanning tree.

Output

Print a single integer — the minimum number of seconds one needs to spend in order to visit all the cities exactly once.

Examples

input
5 2 3 1 2 1 3 3 4 5 3
output
9

input
5 3 2 1 2 1 3 3 4 5 3
output
8

Note

In the first sample, roads of the spanning tree have cost 2, while other roads have cost 3. One example of an optimal path is $5 \to 3 \to 4 \to 1 \to 2$.

In the second sample, we have the same spanning tree, but roads in the spanning tree cost 3, while other roads cost 2. One example of an optimal path is $1 \to 4 \to 5 \to 2 \to 3$.

E. Robot Arm

time limit per test: 8 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Roger is a robot. He has an arm that is a series of n segments connected to each other. The endpoints of the i-th segment are initially located at points $(i - 1, 0)$ and $(i, 0)$. The endpoint at $(i - 1, 0)$ is colored red and the endpoint at $(i, 0)$ is colored blue for all segments. Thus, the blue endpoint of the i-th segment is touching the red endpoint of the $(i + 1)$-th segment for all valid i.

Roger can move his arm in two different ways:

1. He can choose some segment and some value. This is denoted as choosing the segment number i and picking some positive l. This change happens as follows: the red endpoint of segment number i and segments from 1 to $i - 1$ are all fixed in place. Imagine a ray from the red endpoint to the blue endpoint. The blue endpoint and segments $i + 1$ through n are translated l units in the direction of this ray.

   

   In this picture, the red point labeled A and segments before A stay in place, while the blue point labeled B and segments after B gets translated.

2. He can choose a segment and rotate it. This is denoted as choosing the segment number i, and an angle a. The red endpoint of the i-th segment will stay fixed in place. The blue endpoint of that segment and segments $i + 1$ to n will rotate clockwise by an angle of a degrees around the red endpoint.

   

   In this picture, the red point labeled A and segments before A stay in place, while the blue point labeled B and segments after B get rotated around point A.

Roger will move his arm m times. These transformations are a bit complicated, and Roger easily loses track of where the blue endpoint of the last segment is. Help him compute the coordinates of the blue endpoint of the last segment after applying each operation. Note that these operations are cumulative, and Roger's arm may intersect itself arbitrarily during the moves.

Input

The first line of the input will contain two integers n and m ($1 \le n, m \le 300\,000$) — the number of segments and the number of operations to perform.

Each of the next m lines contains three integers x_i, y_i and z_i describing a move. If $x_i = 1$, this line describes a move of type 1, where y_i denotes the segment number and z_i denotes the increase in the length. If $x_i = 2$, this describes a move of type 2, where y_i denotes the segment number, and z_i denotes the angle in degrees. ($1 \le x_i \le 2, 1 \le y_i \le n, 1 \le z_i \le 359$)

Output

Print m lines. The i-th line should contain two real values, denoting the coordinates of the blue endpoint of the last segment after applying operations $1, \ldots, i$. Your answer will be considered correct if its absolute or relative error does not exceed 10^{-4}.

Namely, let's assume that your answer for a particular value of a coordinate is a and the answer of the jury is b. The checker program will consider your answer correct if $\frac{|a-b|}{\max(1,b)} \le 10^{-4}$ for all coordinates.

Examples

input
5 4 1 1 3 2 3 90 2 5 48 1 4 1
output
8.0000000000 0.0000000000 5.0000000000 -3.0000000000 4.2568551745 -2.6691306064 4.2568551745 -3.6691306064

Note

The following pictures shows the state of the arm after each operation. The coordinates of point F are printed after applying each operation. For simplicity, we only show the blue endpoints of a segment (with the exception for the red endpoint of the first segment). For instance, the point labeled B is the blue endpoint for segment 1 and also the red endpoint for segment 2.

Initial state:

Extend segment 1 by 3.

Rotate segment 3 by 90 degrees clockwise.

Rotate segment 5 by 48 degrees clockwise.

Extend segment 4 by 1.

F. Double Knapsack

time limit per test: 2.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given two multisets A and B. Each multiset has exactly n integers each between 1 and n inclusive. Multisets may contain multiple copies of the same number.

You would like to find a nonempty subset of A and a nonempty subset of B such that the sum of elements in these subsets are equal. Subsets are also multisets, i.e. they can contain elements with equal values.

If no solution exists, print -1. Otherwise, print the indices of elements in any such subsets of A and B that have the same sum.

Input

The first line of the input contains a single integer n ($1 \le n \le 1\,000\,000$) — the size of both multisets.

The second line contains n integers, denoting the elements of A. Each element will be between 1 and n inclusive.

The third line contains n integers, denoting the elements of B. Each element will be between 1 and n inclusive.

Output

If there is no solution, print a single integer -1. Otherwise, your solution should be printed on four lines.

The first line should contain a single integer k_a, the size of the corresponding subset of A. The second line should contain k_a distinct integers, the indices of the subset of A.

The third line should contain a single integer k_b, the size of the corresponding subset of B. The fourth line should contain k_b distinct integers, the indices of the subset of B.

Elements in both sets are numbered from 1 to n. If there are multiple possible solutions, print any of them.

Examples

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

input
5 4 4 3 3 3 2 2 2 2 5
output
2 2 3 2 3 5

G. Combining Slimes

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Your friend recently gave you some slimes for your birthday. You have a very large amount of slimes with value 1 and 2, and you decide to invent a game using these slimes.

You initialize a row with n empty spaces. You also choose a number p to be used in the game. Then, you will perform the following steps while the last space is empty.

1. With probability $\frac{p}{10^9}$, you will choose a slime with value 1, and with probability $1 - \frac{p}{10^9}$, you will choose a slime with value 2. You place the chosen slime on the last space of the board.
2. You will push the slime to the left as far as possible. If it encounters another slime, and they have the same value v, you will merge the slimes together to create a single slime with value $v + 1$. This continues on until the slime reaches the end of the board, or encounters a slime with a different value than itself.

You have played the game a few times, but have gotten bored of it. You are now wondering, what is the expected sum of all values of the slimes on the board after you finish the game.

Input

The first line of the input will contain two integers n, p ($1 \le n \le 10^9, 1 \le p < 10^9$).

Output

Print the expected sum of all slimes on the board after the game finishes. Your answer will be considered correct if its absolute or relative error does not exceed 10^{-4}.

Namely, let's assume that your answer is a and the answer of the jury is b. The checker program will consider your answer correct, if $\frac{|a-b|}{\max(1,b)} \le 10^{-4}$.

Examples

input
2 500000000
output
3.562500000000000

input
10 1
output
64.999983360007620

input
100 123456789
output
269.825611298854770

Note

In the first sample, we have a board with two squares, and there is a 0.5 probability of a 1 appearing and a 0.5 probability of a 2 appearing.

Our final board states can be 1 2 with probability 0.25, 2 1 with probability 0.375, 3 2 with probability 0.1875, 3 1 with probability 0.1875. The expected value is thus
$(1 + 2)\cdot 0.25 + (2 + 1)\cdot 0.375 + (3 + 2)\cdot 0.1875 + (3 + 1)\cdot 0.1875 = 3.5625$.