Codeforces Round #331 (Div. 2)

A. Wilbur and Swimming Pool

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

After making bad dives into swimming pools, Wilbur wants to build a swimming pool in the shape of a rectangle in his backyard. He has set up coordinate axes, and he wants the sides of the rectangle to be parallel to them. Of course, the area of the rectangle must be positive. Wilbur had all four vertices of the planned pool written on a paper, until his friend came along and erased some of the vertices.

Now Wilbur is wondering, if the remaining n vertices of the initial rectangle give enough information to restore the area of the planned swimming pool.

Input

The first line of the input contains a single integer n ($1 \le n \le 4$) — the number of vertices that were **not** erased by Wilbur's friend.

Each of the following n lines contains two integers x_i and y_i ($-1000 \le x_i, y_i \le 1000$) —the coordinates of the i-th vertex that remains. Vertices are given in an arbitrary order.

It's guaranteed that these points are distinct vertices of some rectangle, that has positive area and which sides are parallel to the coordinate axes.

Output

Print the area of the initial rectangle if it could be uniquely determined by the points remaining. Otherwise, print -1.

Examples

input
2 0 0 1 1
output
1

input
1 1 1
output
-1

Note

In the first sample, two opposite corners of the initial rectangle are given, and that gives enough information to say that the rectangle is actually a unit square.

In the second sample there is only one vertex left and this is definitely not enough to uniquely define the area.

B. Wilbur and Array

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

Wilbur the pig is tinkering with arrays again. He has the array $a_1, a_2, \ldots, a_n$ initially consisting of n zeros. At one step, he can choose any index i and either add 1 to all elements $a_i, a_{i+1}, \ldots, a_n$ or subtract 1 from all elements $a_i, a_{i+1}, \ldots, a_n$. His goal is to end up with the array $b_1, b_2, \ldots, b_n$.

Of course, Wilbur wants to achieve this goal in the minimum number of steps and asks you to compute this value.

Input

The first line of the input contains a single integer n ($1 \le n \le 200\,000$) — the length of the array a_i. Initially $a_i = 0$ for every position i, so this array is not given in the input.

The second line of the input contains n integers $b_1, b_2, \ldots, b_n$ ($-10^9 \le b_i \le 10^9$).

Output

Print the minimum number of steps that Wilbur needs to make in order to achieve $a_i = b_i$ for all i.

Examples

input
5 1 2 3 4 5
output
5

input
4 1 2 2 1
output
3

Note

In the first sample, Wilbur may successively choose indices 1, 2, 3, 4, and 5, and add 1 to corresponding suffixes.

In the second sample, Wilbur first chooses indices 1 and 2 and adds 1 to corresponding suffixes, then he chooses index 4 and subtract 1.

C. Wilbur and Points

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

Wilbur is playing with a set of n points on the coordinate plane. All points have non-negative integer coordinates. Moreover, if some point (x, y) belongs to the set, then all points (x', y'), such that $0 \le x' \le x$ and $0 \le y' \le y$ also belong to this set.

Now Wilbur wants to number the points in the set he has, that is assign them distinct integer numbers from 1 to n. In order to make the numbering *aesthetically pleasing*, Wilbur imposes the condition that if some point (x, y) gets number i, then all (x',y') from the set, such that $x' \ge x$ and $y' \ge y$ must be assigned a number not less than i. For example, for a set of four points $(0, 0)$, $(0, 1)$, $(1, 0)$ and $(1, 1)$, there are two aesthetically pleasing numberings. One is $1, 2, 3, 4$ and another one is $1, 3, 2, 4$.

Wilbur's friend comes along and challenges Wilbur. For any point he defines it's *special value* as $s(x, y) = y - x$. Now he gives Wilbur some $w_1, w_2,..., w_n$, and asks him to find an aesthetically pleasing numbering of the points in the set, such that the point that gets number i has it's special value equal to w_i, that is $s(x_i, y_i) = y_i - x_i = w_i$.

Now Wilbur asks you to help him with this challenge.

Input

The first line of the input consists of a single integer n ($1 \le n \le 100\,000$) — the number of points in the set Wilbur is playing with.

Next follow n lines with points descriptions. Each line contains two integers x and y ($0 \le x, y \le 100\,000$), that give one point in Wilbur's set. It's guaranteed that all points are distinct. Also, it is guaranteed that if some point (x, y) is present in the input, then all points (x', y'), such that $0 \le x' \le x$ and $0 \le y' \le y$, are also present in the input.

The last line of the input contains n integers. The i-th of them is w_i ($-100\,000 \le w_i \le 100\,000$) — the required special value of the point that gets number i in any aesthetically pleasing numbering.

Output

If there exists an aesthetically pleasant numbering of points in the set, such that $s(x_i, y_i) = y_i - x_i = w_i$, then print "YES" on the first line of the output. Otherwise, print "NO".

If a solution exists, proceed output with n lines. On the i-th of these lines print the point of the set that gets number i. If there are multiple solutions, print any of them.

Examples

input
5 2 0 0 0 1 0 1 1 0 1 0 -1 -2 1 0

output
YES 0 0 1 0 2 0 0 1 1 1

input
3 1 0 0 0 2 0 0 1 2

output
NO

Note

In the first sample, point $(2, 0)$ gets number 3, point $(0, 0)$ gets number one, point $(1, 0)$ gets number 2, point $(1, 1)$ gets number 5 and point $(0, 1)$ gets number 4. One can easily check that this numbering is aesthetically pleasing and $y_i - x_i = w_i$.

In the second sample, the special values of the points in the set are 0, -1, and -2 while the sequence that the friend gives to Wilbur is $0, 1, 2$. Therefore, the answer does not exist.

D. Wilbur and Trees

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

Wilbur the pig really wants to be a beaver, so he decided today to pretend he is a beaver and bite at trees to cut them down.

There are n trees located at various positions on a line. Tree i is located at position x_i. All the given positions of the trees are distinct.

The trees are equal, i.e. each tree has height h. Due to the wind, when a tree is cut down, it either falls left with probability p, or falls right with probability $1 - p$. If a tree hits another tree while falling, that tree will fall in the same direction as the tree that hit it. A tree can hit another tree only if the distance between them is strictly less than h.

For example, imagine there are 4 trees located at positions 1, 3, 5 and 8, while $h = 3$ and the tree at position 1 falls right. It hits the tree at position 3 and it starts to fall too. In it's turn it hits the tree at position 5 and it also starts to fall. The distance between 8 and 5 is exactly 3, so the tree at position 8 will not fall.

As long as there are still trees standing, Wilbur will select either the leftmost standing tree with probability 0.5 or the rightmost standing tree with probability 0.5. Selected tree is then cut down. If there is only one tree remaining, Wilbur always selects it. As the ground is covered with grass, Wilbur wants to know the expected total length of the ground covered with fallen trees after he cuts them all down because he is concerned about his grass-eating cow friends. Please help Wilbur.

Input

The first line of the input contains two integers, n ($1 \le n \le 2000$) and h ($1 \le h \le 10^8$) and a real number p ($0 \le p \le 1$), given with no more than six decimal places.

The second line of the input contains n integers, $x_1, x_2, \ldots, x_n$ ($-10^8 \le x_i \le 10^8$) in no particular order.

Output

Print a single real number — the expected total length of the ground covered by trees when they have all fallen down. Your answer will be considered correct if its absolute or relative error does not exceed 10^{-6}.

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 $\frac{|a-b|}{\max(1,b)} \le 10^{-6}$.

Examples

input
2 2 0.500000 1 2
output
3.250000000

input
4 3 0.4 4 3 1 2
output
6.631200000

Note

Consider the first example, we have 2 trees with height 2.

There are 3 scenarios:
1. Both trees falls left. This can either happen with the right tree falling left first, which has $\frac{1}{2} \cdot \frac{1}{2} = \frac{1}{4}$ probability (also knocking down the left tree), or the left tree can fall left and then the right tree can fall left, which has $\frac{1}{4} \cdot \frac{1}{2} = \frac{1}{8}$ probability. Total probability is $\frac{1}{4} + \frac{1}{8} = \frac{3}{8}$.

2. Both trees fall right. This is analogous to (1), so the probability of this happening is $\frac{3}{8}$.

3. The left tree fall left and the right tree falls right. This is the only remaining scenario so it must have $1 - \frac{3}{8} - \frac{3}{8} = \frac{1}{4}$ probability.

Cases 1 and 2 lead to a total of 3 units of ground covered, while case 3 leads to a total of 4 units of ground covered. Thus, the expected value is $3 \cdot (\frac{3}{8}) + 3 \cdot (\frac{3}{8}) + 4 \cdot (\frac{1}{4}) = 3.25$.

E. Wilbur and Strings

time limit per test: 3 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Wilbur the pig now wants to play with strings. He has found an n by m table consisting only of the digits from 0 to 9 where the rows are numbered 1 to n and the columns are numbered 1 to m. Wilbur starts at some square and makes certain moves. If he is at square (x, y) and the digit d ($0 \le d \le 9$) is written at position (x, y), **then he must** move to the square $(x + a_d, y + b_d)$, if that square lies within the table, and he stays in the square (x, y) otherwise. Before Wilbur makes a move, he can choose whether or not to write the digit written in this square on the white board. All digits written on the whiteboard form some string. Every time a new digit is written, it goes to the end of the current string.

Wilbur has q strings that he is worried about. For each string s_i, Wilbur wants to know whether there exists a starting position (x, y) so that by making finitely many moves, Wilbur can end up with the string s_i written on the white board.

Input

The first line of the input consists of three integers n, m, and q ($1 \le n, m, q \le 200$) — the dimensions of the table and the number of strings to process, respectively.

Each of the next n lines contains m digits from 0 and 9 giving the table itself.

Then follow 10 lines. The i-th of them contains the values a_{i-1} and b_{i-1} ($-200 \le a_i, b_i \le 200$), i.e. the vector that Wilbur uses to make a move from the square with a digit $i - 1$ in it.

There are q lines that follow. The i-th of them will contain a string s_i consisting only of digits from 0 to 9. It is guaranteed that the total length of these q strings won't exceed $1\,000\,000$.

Output

For each of the q strings, print "YES" if Wilbur can choose x and y in order to finish with this string after some finite number of moves. If it's impossible, than print "NO" for the corresponding string.

Examples

input
1 1 2 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0000000000000 2413423432432
output
YES NO

input
4 2 5 01 23 45 67 0 1 0 -1 0 1 0 -1 0 1 0 -1 0 1 0 -1 0 1 0 -1 0000000000 010101011101 32232232322 44343222342444324 6767
output

```
YES
YES
YES
NO
YES
```

Note

In the first sample, there is a 1 by 1 table consisting of the only digit 0. The only move that can be made is staying on the square. The first string can be written on the white board by writing 0 repeatedly. The second string cannot be written as there is no 2 on the table.