Codeforces Round #313 (Div. 1)

A. Gerald's Hexagon

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

Gerald got a very curious hexagon for his birthday. The boy found out that all the angles of the hexagon are equal to $120°$. Then he measured the length of its sides, and found that each of them is equal to an integer number of centimeters. There the properties of the hexagon ended and Gerald decided to draw on it.

He painted a few lines, parallel to the sides of the hexagon. The lines split the hexagon into regular triangles with sides of 1 centimeter. Now Gerald wonders how many triangles he has got. But there were so many of them that Gerald lost the track of his counting. Help the boy count the triangles.

Input

The first and the single line of the input contains 6 space-separated integers a_1, a_2, a_3, a_4, a_5 and a_6 ($1 \le a_i \le 1000$) — the lengths of the sides of the hexagons in centimeters in the clockwise order. It is guaranteed that the hexagon with the indicated properties and the exactly such sides exists.

Output

Print a single integer — the number of triangles with the sides of one 1 centimeter, into which the hexagon is split.

Examples

input
1 1 1 1 1 1
output
6

input
1 2 1 2 1 2
output
13

Note

This is what Gerald's hexagon looks like in the first sample:

And that's what it looks like in the second sample:

B. Equivalent Strings

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

Today on a lecture about strings Gerald learned a new definition of string equivalency. Two strings a and b of equal length are called *equivalent* in one of the two cases:

1. They are equal.
2. If we split string a into two halves of the same size a_1 and a_2, and string b into two halves of the same size b_1 and b_2, then one of the following is correct:
   a. a_1 is equivalent to b_1, and a_2 is equivalent to b_2
   b. a_1 is equivalent to b_2, and a_2 is equivalent to b_1

As a home task, the teacher gave two strings to his students and asked to determine if they are equivalent.

Gerald has already completed this home task. Now it's your turn!

Input

The first two lines of the input contain two strings given by the teacher. Each of them has the length from 1 to $200\,000$ and consists of lowercase English letters. The strings have the same length.

Output

Print "YES" (without the quotes), if these two strings are equivalent, and "NO" (without the quotes) otherwise.

Examples

input
aaba abaa
output
YES

input
aabb abab
output
NO

Note

In the first sample you should split the first string into strings "aa" and "ba", the second one — into strings "ab" and "aa". "aa" is equivalent to "aa"; "ab" is equivalent to "ba" as "ab" = "a" + "b", "ba" = "b" + "a".

In the second sample the first string can be splitted into strings "aa" and "bb", that are equivalent only to themselves. That's why string "aabb" is equivalent only to itself and to string "bbaa".

C. Gerald and Giant Chess

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

Giant chess is quite common in Geraldion. We will not delve into the rules of the game, we'll just say that the game takes place on an $h \times w$ field, and it is painted in two colors, but not like in chess. Almost all cells of the field are white and only some of them are black. Currently Gerald is finishing a game of giant chess against his friend Pollard. Gerald has almost won, and the only thing he needs to win is to bring the pawn from the upper left corner of the board, where it is now standing, to the lower right corner. Gerald is so confident of victory that he became interested, in how many ways can he win?

The pawn, which Gerald has got left can go in two ways: one cell down or one cell to the right. In addition, it can not go to the black cells, otherwise the Gerald still loses. There are no other pawns or pieces left on the field, so that, according to the rules of giant chess Gerald moves his pawn until the game is over, and Pollard is just watching this process.

Input

The first line of the input contains three integers: h, w, n — the sides of the board and the number of black cells ($1 \le h, w \le 10^5, 1 \le n \le 2000$).

Next n lines contain the description of black cells. The i-th of these lines contains numbers r_i, c_i ($1 \le r_i \le h, 1 \le c_i \le w$) — the number of the row and column of the i-th cell.

It is guaranteed that the upper left and lower right cell are white and all cells in the description are distinct.

Output

Print a single line — the remainder of the number of ways to move Gerald's pawn from the upper left to the lower right corner modulo $10^9 + 7$.

Examples

input
3 4 2 2 2 2 3
output
2

input
100 100 3 15 16 16 15 99 88
output
545732279

D. Randomizer

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

Gerald got tired of playing board games with the usual six-sided die, and he bought a toy called Randomizer. It functions as follows.

A Randomizer has its own coordinate plane on which a strictly convex polygon is painted, the polygon is called a **basic polygon**. If you shake a Randomizer, it draws some nondegenerate (i.e. having a non-zero area) convex polygon with vertices at some vertices of the **basic polygon**. The result of the roll (more precisely, the result of the shaking) is considered to be the number of points with integer coordinates, which were strictly inside (the points on the border are not considered) the selected polygon. Now Gerald is wondering: what is the expected result of shaking the Randomizer?

During the shaking the Randomizer considers all the possible non-degenerate convex polygons with vertices at the vertices of the **basic polygon**. Let's assume that there are k versions of the polygons. Then the Randomizer chooses each of them with probability $\frac{1}{k}$.

Input

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

Next n lines contain the coordinates of the vertices of the basic polygon. The i-th of these lines contain two integers x_i and y_i ($-10^9 \le x_i, y_i \le 10^9$) — the coordinates of the i-th vertex of the polygon. The vertices are given in the counter-clockwise order.

Output

Print the sought expected value with absolute or relative error at most 10^{-9}.

Examples

input
4 0 0 2 0 2 2 0 2
output
0.2

input
5 0 0 2 0 2 2 1 3 0 2
output
0.8125

Note

A polygon is called strictly convex if it is convex and no its vertices lie on the same line.

Let's assume that a random variable takes values $x_1, \ldots, x_n$ with probabilities $p_1, \ldots, p_n$, correspondingly. Then the expected value of this variable equals to $\sum_{i=1}^{n} x_i p_i$.

E. Gerald and Path

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The main walking trail in Geraldion is absolutely straight, and it passes strictly from the north to the south, it is so long that no one has ever reached its ends in either of the two directions. The Geraldionians love to walk on this path at any time, so the mayor of the city asked the Herald to illuminate this path with a few spotlights. The spotlights have already been delivered to certain places and Gerald will not be able to move them. Each spotlight illuminates a specific segment of the path of the given length, one end of the segment is the location of the spotlight, and it can be directed so that it covers the segment to the south or to the north of spotlight.

The trail contains a monument to the mayor of the island, and although you can walk in either directions from the monument, no spotlight is south of the monument.

You are given the positions of the spotlights and their power. Help Gerald direct all the spotlights so that the total length of the illuminated part of the path is as much as possible.

Input

The first line contains integer n ($1 \le n \le 100$) — the number of spotlights. Each of the n lines contains two space-separated integers, a_i and l_i ($0 \le a_i \le 10^8$, $1 \le l_i \le 10^8$). Number a_i shows how much further the i-th spotlight to the north, and number l_i shows the length of the segment it illuminates.

It is guaranteed that all the a_i's are distinct.

Output

Print a single integer — the maximum total length of the illuminated part of the path.

Examples

input
3 1 1 2 2 3 3
output
5

input
4 1 2 3 3 4 3 6 2
output
9