Problem A. Buttons

Input file:	`standard input`
Output file:	`standard output`
C++ time limit:	1 second
Memory limit:	256 megabytes

You have found a very unusual device. Apart from a display that shows some positive integer, the device also has two buttons on the front: a **red** button and a **blue** button. When you press the red button, the displayed integer gets multiplied by some positive integer A. When pressing the blue button, the device subtracts one from the displayed integer.

The device initially displays the integer N. Your task is to get the device to display the integer M using the **fewest** number of button presses.

Input

Input consists of a single line containing three space-separated integers A, N, M where $2 \le A \le 1000$ and $1 \le N, M \le 10^{15}$.

Output

Output consists of a single integer giving the minimum number of button presses required to obtain M from N.

Scoring

Subtask 1 (4 points): $M \le N \le 10^6$

Subtask 2 (11 points): $A = 2$, $N \le M \le 5$

Subtask 3 (50 points): $A \le 100$ and $M \le 50\,000$

Subtask 4 (15 points): $A = 2$

Subtask 5 (20 points): No further constraints

Examples

standard input	standard output
2 4 6	2
3 3 11	6
5 10 1	9

Note

In the first example given above, the optimal solution is to first press the blue button (subtract 1), then press the red button (multiply by 2) to get from 4 to 6 in just 2 moves.

In the second example, the optimal solution is: $3 \to 2 \to 6 \to 5 \to 4 \to 12 \to 11$.

In the third example, multiplying the number by 5 is unnecessary, thus the optimal solution is to simply press the blue button nine times.

Problem B. Stones

Input file:	`standard input`
Output file:	`standard output`
C++ time limit:	2 seconds
Memory limit:	256 megabytes

Ayanda would like to build a stone tower to help mark trails to prevent hikers from getting lost. In order to build this tower, she will need to collect a number of large stones in the surrounding area. Naturally, she would like to be as efficient as possible, and minimize the number of steps she has to take.

The area is modelled as an $N \times M$ grid of numbers (N rows, M columns). A number s ($0 \leq s \leq 10^6$) denotes that there are s stones at that cell. We will use the notation (i, j) to indicate the cell in row i, column j. The tower will be built at some cell (a, b) ($0 \leq a < N, 0 \leq b < M$). In one step, Ayanda can move from her current cell to any of the 8 neighbouring cells (the cells which share an edge or vertex with his current cell). Ayanda collects one stone at a time, taking a number of steps from (a, b), to a cell containing a stone, and finally bringing the stone back to (a, b).

Ayanda is not sure where she would like to build the tower, and how many stones she should use. Every now and then after receiving new information, she also updates a cell of the grid with a new number of stones.

Please help her by determining the number of steps required for a number of different queries, using the latest grid values each time.

Input

The first line consists of the 3 space separated integers N, M, Q, where Q denotes the number of queries ($1 \leq N \leq 1\,000, 1 \leq M \leq 1\,000, 1 \leq Q \leq 1\,000$).

The next N lines each contain M space separated integers, denoting the numbers s in the grid. The first of these lines contains the values of cells $(0, 0), (0, 1), (0, 2), ...(0, M - 1)$. The next line contains the values of cells $(1, 0), (1, 1), ...$ and so on, up to the last line containing the values of cells $(N - 1, 0), (N - 1, 1), ..., (N - 1, M - 1)$.

The next Q lines each contain a single query, of which there are two types.

- Type 0 queries consist of the 4 integers, 0, a, b, K, space separated. Here K ($0 \leq K \leq 10^{12}$) is the number of stones required for the tower.
- Type 1 queries consist of the 4 integers, 1, i, j, s, space separated. This indicates the value of cell (i, j) should be updated to s.

Output

For each query of type 0, output a single integer, the number of steps required to collect K stones for the tower if it was built at (a, b), taking note only of any type 1 queries which happened before. If it is not possible to collect K stones, output -1 instead.

Scoring

Subtask 1 (20 points): $N, M \leq 50$.

Subtask 2 (20 points): For each query of type 0, (a, b) is always (0, 0).

Subtask 3 (30 points): All queries will be of type 0.

Subtask 4 (30 points): No further restrictions.

Example

standard input	standard output
4 5 6 0 0 1 0 2 1 0 0 0 0 0 0 0 1 0 4 0 0 0 1 0 0 0 2 0 2 0 6 0 3 4 3 0 3 3 12 1 0 0 1 0 0 0 2	6 14 8 -1 2

Note

For the first query, the tower must be built at the top left corner of the grid, at position $(0, 0)$. One stone can be collected from the cell below, requiring 2 steps, and one stone from the cell two to the right, requiring another 4 steps, giving a total of 6.

For the second query, the tower must be built in the cell above the '4', at position $(2, 0)$. The stones above and below this cell can be collected with 10 steps. A further 4 steps are required to collect the stone at position $(0, 3)$, giving a total of 14.

For the last query, the tower must be built at the top left corner of the grid, at position $(0, 0)$. $(0, 0)$ has been updated to 1 by the previous query. Hence the first stone is already there, requiring 0 steps. The second stone can be collected from the cell below, requiring 2 steps.

Problem C. Stadium

Input file:	`standard input`
Output file:	`standard output`
C++ time limit:	5 seconds
Memory limit:	256 megabytes

City planners are planning to build a new sports stadium in a park. They want to make it as big as possible! The catch is that the park contains a number of very old and beloved trees that cannot be cut down to make space for the stadium. Thus, the sides of the stadium can touch the trees, but not contain them.

The stadium must also be square, with the sides parallel to the sides of the park. The park itself is a square with side length W.

Given the coordinates of the trees, determine the largest possible stadium that can be built.

Input

The first line of input contains N ($0 \le N \le 100\,000$), the number of trees, and W ($1 \le W \le 10^9$), the side length of the park. This is followed by N lines, each containing the coordinates of one tree, as two integers x and y ($0 \le x, y \le W$), with the coordinate axes aligned to the sides of the park. No two trees have the same coordinates.

Output

Output a line containing a single integer, the maximum possible side length of the stadium.

Scoring

Subtask 1 (10 points): $N \le 100$

Subtask 2 (15 points): $N \le 5000$

Subtask 3 (15 points): $W \le 2000$

Subtask 4 (60 points): No further restrictions

Example

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