

Problem H. One Step Closer

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

Consider a table of size $n \times m$ consisting of characters “+” and “-”. A single operation with the table is to choose a cell and invert all characters in its row and column simultaneously (the cell itself is also inverted). Inverting a character means replacing “+” by “-” and vice versa.

Vasya wants to fill all cells with “-”. To achieve that, he uses the following algorithm:

- If all cells are “-”, end the algorithm.
- Otherwise, remember the current positions of “+”. Assume they are $(r_1, c_1), (r_2, c_2), \dots, (r_k, c_k)$. After that, he performs one operation for each cell (r_i, c_i) and proceeds to step 1.

Sasha is interested whether Vasya can reach his goal. If this is the case, he wants to know the total number of operations Vasya will perform before he stops.

Input

The first line of input contains three integers n, m and q ($1 \leq n, m \leq 10^5, 0 \leq q \leq 10^5$).

Then follows the description of the initial configuration which looks as follows. The i -th of the next q lines contains four integers $x_{i,1}, y_{i,1}, x_{i,2}$ and $y_{i,2}$ ($1 \leq x_{i,1} \leq x_{i,2} \leq n, 1 \leq y_{i,1} \leq y_{i,2} \leq m$). They define a rectangle containing all such cells (x, y) that $x_{i,1} \leq x \leq x_{i,2}$ and $y_{i,1} \leq y \leq y_{i,2}$. A cell (x, y) is initially “+” if and only if it is contained in an odd number of rectangles.

See Notes section for further explanation.

Output

If Vasya will never finish performing operations, print “-1”. Otherwise, print the number of operations he will perform.

Examples

standard input	standard output
3 3 2 1 1 2 2 2 2 3 3	6
3 3 2 1 2 1 3 2 1 3 1	-1

Note

In the first example, the board initially looks as follows:

```
++-  
++-  
--+
```

After the only iteration of the algorithm, which consists of six operations, the board is filled with “-”.

In the second example, the board looks as follows and stays the same after each iteration of the algorithm:

```
--+  
+--  
+--
```