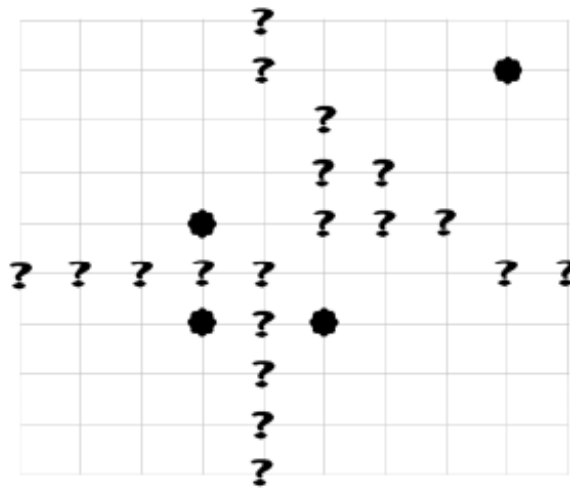


# Mravograd

The hard working ants have built a town called Ant Town. They modeled their town after Manhattan, with  $H$  horizontal and  $V$  vertical streets which cross in  $V \times O$  intersections. As ants don't like water, with the first raindrops comes chaos in Ant Town. Town authorities have placed umbrellas under which any number of ants can hide, but only on  $N$  intersections.

When the rain starts, each ant on an intersection starts running, **using streets**, to the nearest intersection with an umbrella. But, if an ant can choose from more than one such intersection, it panics and, not knowing where to go, **stays on its starting intersection** and gets wet. Town authorities use the name "wet intersections" for such starting intersections.

For example, if Ant Town has 10 horizontal and 10 vertical streets, and if there are 4 intersections with umbrellas, then the question marks in the figure represent "wet intersections":



□

*Picture represents first example. We count streets from left to right from 1 to  $V$  and from down upwards from 1 to  $H$ .*

Write a program which, given the locations of intersections with umbrellas, determines the **number of "wet intersections"** in Ant Town.

## Input

The first line contains two integers  $H$  and  $V$  ( $1 \leq H, V \leq 30\,000$ ), the numbers of horizontal and vertical streets in Ant Town.

Horizontal streets are numbered 1 to  $H$ , vertical streets 1 to  $V$ .

The second line contains an integer  $N$  ( $1 \leq N \leq 10$ ), the number of intersections with umbrellas.

Each of the following  $N$  lines contains two integer  $h$  and  $v$ , meaning that there is an umbrella on the crossing of horizontal street  $h$  and vertical street  $v$ . The locations of all umbrellas will be distinct.

## Output

Output the number of "wet intersections" in Ant Town.

## Example

**Input:**

10 10

4

4 4

4 6

6 4

9 9

**Output:**

19

**Input:**

9 9

3

2 2

5 5

8 8

**Output:**

36

**Input:**

100 100

2

50 50

50 51

**Output:**

0