# Task: ARK
# Arkanoid

**Farias Brito High School Programming Camp, Day 2. Available memory: 512 MB.** *26.07.3016*

*Arkanoid* is a computer game in which the player bounces a ball with a moving paddle (racket). The goal is to remove all bricks from the playing field, where each brick is removed when the ball strikes it (and bounces back). All who played the game know how frustrating and time-consuming striking the last few bricks can be. It is convenient then to have a program that, for a given initial playing field configuration, determines the time required to win the game. For the purpose of this task, we assume that the player plays perfectly, i.e., always bounces the ball off the mid-point of their paddle.

The playing field has a width of $m$ and a height of $n$, where $m$ is odd and $m$ and $n$ are co-prime (their greatest common divisor is 1). We introduce a Cartesian coordinate system on the playing field such that the bottom left corner has coordinates $(0, 0)$ and the top right corner has coordinates $(m, n)$. For simplicity, we assume that both the ball's size and the paddle's thickness are negligible. The paddle moves along the line $y = 0$, the initial position of the ball is $(\frac{m}{2}, 0)$, and its initial velocity vector is $(-\frac{1}{2}, \frac{1}{2})$.

When the ball hits the paddle, an edge of the field, or any brick, it bounces back in an elastic collision. However, any brick that is hit crumbles and is immediately removed from the field. How long until all the bricks are removed?

## Input

In the first line of the input, there are three integers, $m$, $n$, and $k$ ($m, n, k \geq 1$, $k \leq nm - 1$), separated by single spaces, that specify the playing field's dimensions and the initial number of bricks in it. The $k$ lines that follow describe the bricks: the $i$-th such line contains a pair of integers $x_i$ and $y_i$ ($1 \leq x_i \leq m$, $1 \leq y_i \leq n$), separated by a single space, which specify that there is a square brick in the field whose opposite corners are at points $(x_i - 1, y_i - 1)$ and $(x_i, y_i)$. You may assume that there is no brick in the square corresponding to $x_i = \frac{m+1}{2}$, $y_i = 1$.

## Output

In the sole line of the output, a single integer should be printed, equal to the number of time units which pass until all the bricks are removed.

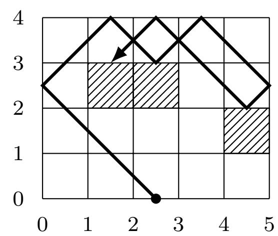
## Example

For the input data:

```
5 4 3
2 3
5 2
3 3
```

the correct result is:

```
22
```




## Grading

| Subset | Constraints | Score |
|---|---|---|
| 1 | $m, n \leq 100$, $k \leq 1000$ | 25 |
| 2 | $n, m \leq 100\,000$, $k \leq 50$ | 25 |
| 3 | $m, n, k \leq 100\,000$, no brick shares a side with either another brick or the field's edge (note that the bricks may share a corner) | 25 |
| 4 | $m, n, k \leq 100\,000$ | 25 |