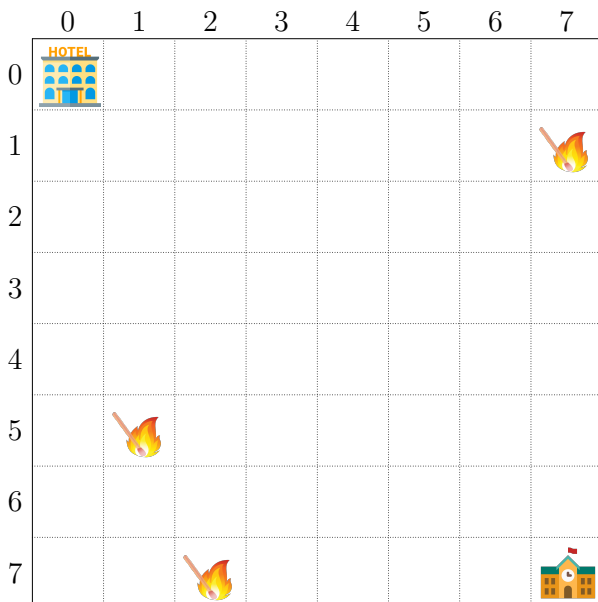




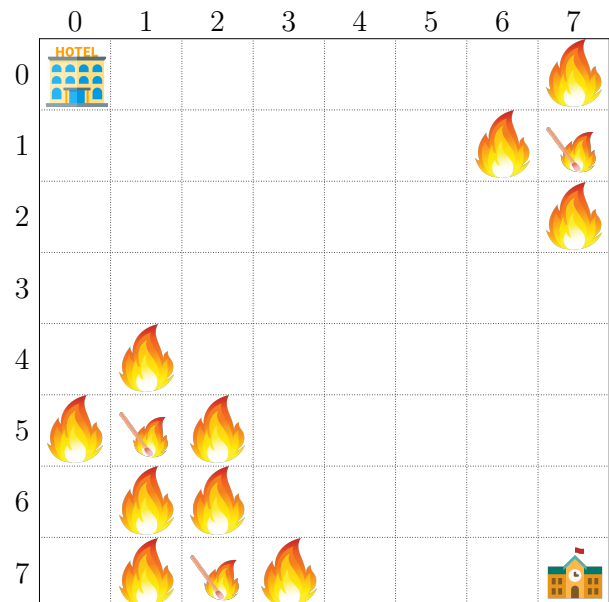
Attenti all'incendio (incendio)

The Italian Olympiad in Informatics committee thinks that this year's tasks might be a tad too easy. For this reason, the contest venue in Campobasso and the hotel were placed on a $N \times N$ grid, where each cell is highly flammable. The hotel is placed at the coordinates $(0,0)$, while the contest venue is placed at the coordinates $(N-1, N-1)$. Thus, besides solving the problems, our contestants this year will also have to avoid the fires they will encounter on the way from the hotel to the contest venue.

The committee managed to get the permission by the city council to start *at the same time* M fires, numbered from 0 to $M-1$. The i -th fire starts at the coordinates $(X[i], Y[i])$. For every minute passed, every cell on fire will ignite a new fire inside its four adjacent cells (north, south, east, west), if those weren't on fire already. The fires will spread on the grid until it is completely on fire, but they are controlled so that they won't escape the grid (for public safety reasons).



Situation at the beginning



Situation after a minute

For the contestants, clearly, it is better to leave the hotel as soon as possible to have more chance to get to the content unharmed. Chupito, instead, like most cats, is pathologically lazy and doesn't want to get up early unless strictly necessary.

At any moment Chupito can move to one of the four cells adjacent to its current location, but he cannot exit the grid, and cannot go across any cell which is on fire. In order to sleep as much as possible, Chupito wants to leave the hotel at the latest possible time.

Help Chupito! Compute **after how many minutes at most** after the fires are started there is still some route to reach the contest venue.

Implementation

You have to submit one file, with extension `.c` or `.cpp`.

🔗 You will find among the attachments the templates `incendio.c` and `incendio.cpp` with an implementation example.

You have to implement the following function:

```
C/C++ | int alzati(int N, int M, int X[], int Y[]);
```

- The integer N is the dimension of the city grid.
- The integer M is the number of fires started initially.
- The arrays X e Y (indexed from 0 to $M - 1$) contain the initial coordinates of the M fires. The i -th fire starts at the coordinates $(X[i], Y[i])$.
- The function must return an integer representing the maximum number of minutes after which it is still possible to reach the contest venue safely.

The grader will call the function `alzati` and will print its return value to the output.

Test grader

In the task folder there is a simplified version of the grader used during evaluation. You can use this sample grader to test your solutions locally. This grader reads the input from `stdin`, calls the functions you implemented, and writes the output on `stdout`.

The input file is formed by $M + 1$ lines, each with the following:

- Line 1: the integers N and M separated by a space.
- Each of the following M lines: the values $X[i]$ and $Y[i]$, separated by a space, for $i = 0 \dots M - 1$.

The output file is formed by a single line, with:

- Line 1: the value returned by function `alzati`.

Constraints

- $2 \leq N \leq 1\,000\,000\,000$.
- $1 \leq M \leq 12\,000$.
- $0 \leq X[i], Y[i] \leq N - 1$ for each $i = 0, \dots, M - 1$.
- All the fires start from distinct coordinates.
- Initially, it is possible to reach the contest venue.
- Initially, there are no fires at the coordinates $(0, 0)$ and $(N - 1, N - 1)$.

Scoring

Your program will be tested on several test cases grouped in subtask. To achieve the score of a subtask, you need to correctly solve all of its test cases.

- **Subtask 1** [0 points]: Sample cases.
- **Subtask 2** [7 points]: $M = 1$.
- **Subtask 3** [10 points]: $X[i] = X[j]$ for every $0 \leq i, j < M$.
- **Subtask 4** [20 points]: $N \leq 500$.

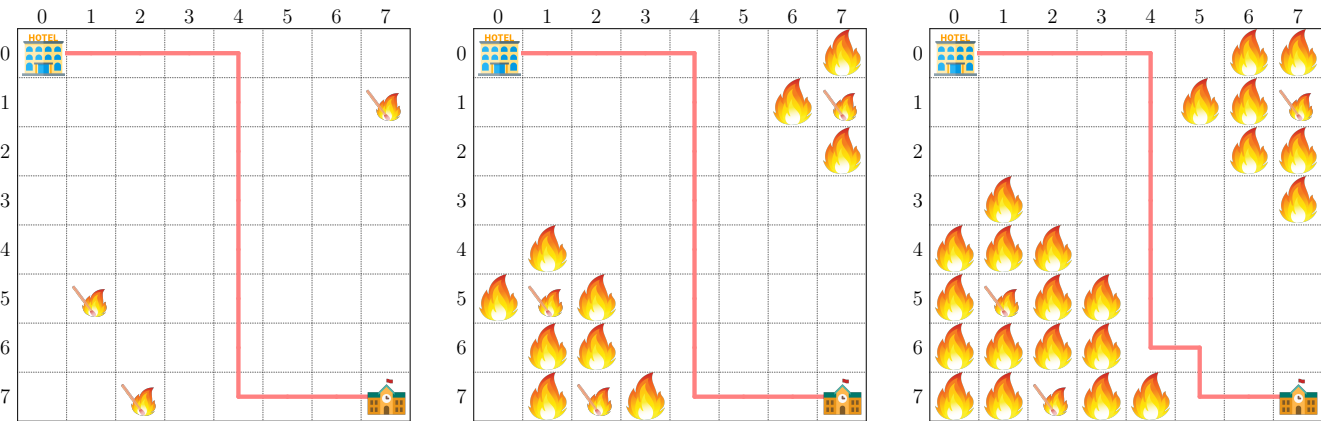
- **Subtask 5 [22 points]:** $N \leq 1000$.
- **Subtask 6 [24 points]:** $M \leq 2000$ and $X[i] + Y[i]$ is even, for every $0 \leq i < M$.
- **Subtask 7 [9 points]:** $M \leq 10\,000$.
- **Subtask 8 [8 points]:** No additional limitations.

Examples

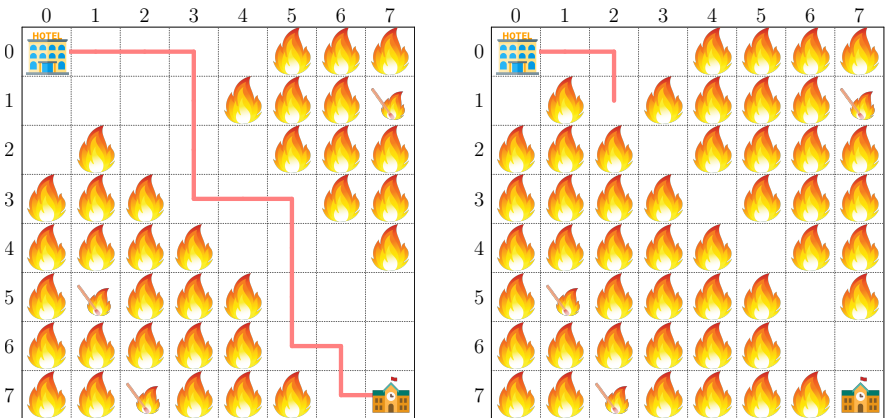
stdin	stdout
<pre> 8 3 1 7 5 1 7 2 </pre>	<pre> 3 </pre>
<pre> 7 3 3 4 0 2 6 3 </pre>	<pre> 1 </pre>

Explanation

In the **first sample case** we have an 8×8 grid initially containing 3 fires. The grid evolves as follows.



Starting with time $t = 4$ it will not be possible to reach the contest venue safely, so the time $t = 3$ is the last minute in which Chupito can get up.



In the **second sample case**, after $t = 1$ it is already not possible to reach the contest venue:

