NORT

Input File: *nortin.txt*
Output File: *nortout.txt*

Time Limit: 1 second

It is the year 1982. Malcolm Fraser is the Prime Minister of Australia, you frequently listen to *Down Under* by Men at Work on your boombox and roller blading is the default mode of transport. As a budding computer scientist, you spend most weekends at the local arcade trying to top your score in the popular video game *NORT*.

NORT is played on a rectangular grid of square cells with H columns and W rows. You ride a light-bike through the grid, starting in the top-left corner cell. Each second, you can move your light-bike up, down, left or right to any of the four adjacent grid cells (as long as you don't go off the grid!). As you move, your bike's exhaust pipe creates an impenetrable wall of light in the cell you were previously in. If you ever move into a cell containing a wall of light (i.e. a cell you have travelled through before), your bike will disintegrate into pixels in a dramatic 8-bit explosion. The only exception is if you return to the top-left corner cell where you started, in which case the wall of light will be connected and your score for the game will be the total length of wall you have created.

Your goal is therefore to plan the longest possible route through the grid starting and ending in the top-left cell without ever passing through a cell more than once.

Input

Your program should read from the file `nortin.txt`. The first and only line of this file will consist of two space-separated integers W and H respectively.

Output

Your program should write to the file `nortout.txt`. Your output file should consist of one line containing one integer: the length of the longest possible connected wall of light you can create.

Sample Input 1

```
4 2
```

Sample Output 1

```
8
```

Sample Input 2

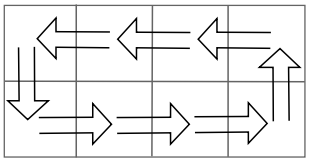
```
3 3
```

Sample Output 2

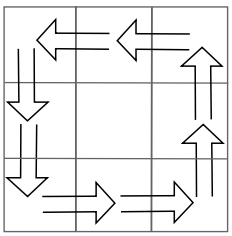
```
8
```

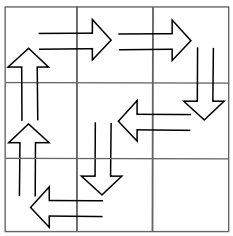
Explanation

In the first example, we are able to find a route which travels through every cell as follows:



In the second example, there are many routes that travel through 8 cells, but none that travel through 9. Two possible routes that would score 8 are shown below.





Constraints

To evaluate your solution, the judges will run your program against several different input files. All of these files will adhere to the following bounds:

- $2 \leq W, H \leq 1,000$

As some of the test cases will be quite large, you may need to think about how well your solution scales for larger input values. However, not all the cases will be large. In particular:

- For 50% of cases, $2 \leq W, H \leq 5$.

Scoring

The score for each input scenario will be 100% if the correct answer is written to the output file, and 0% otherwise.