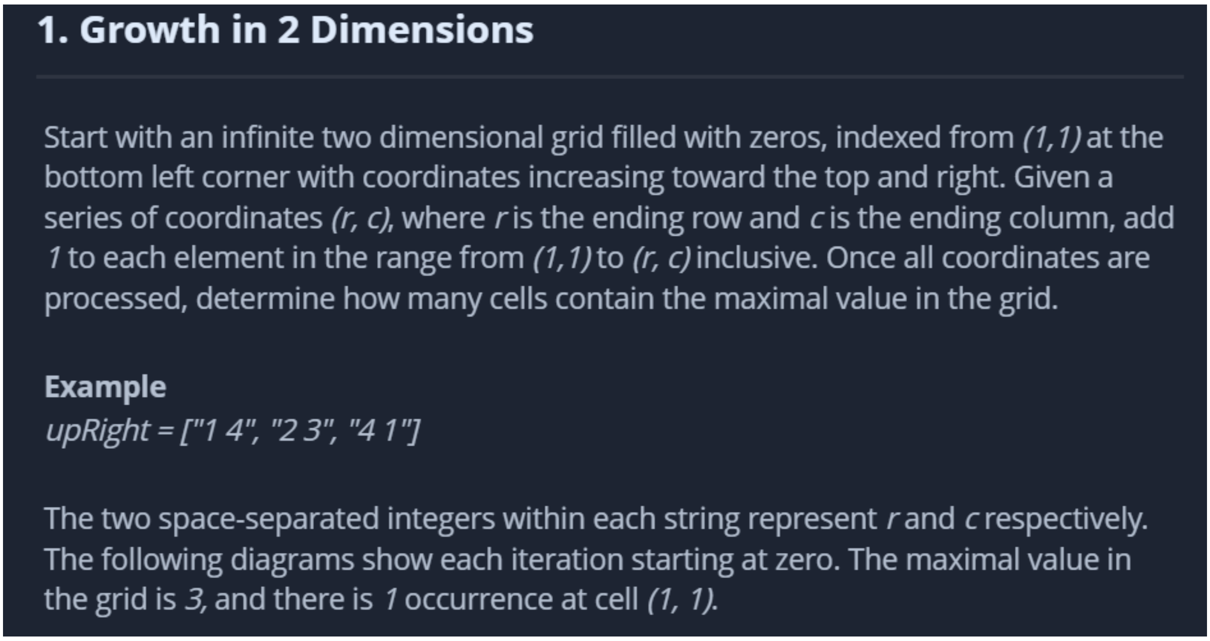


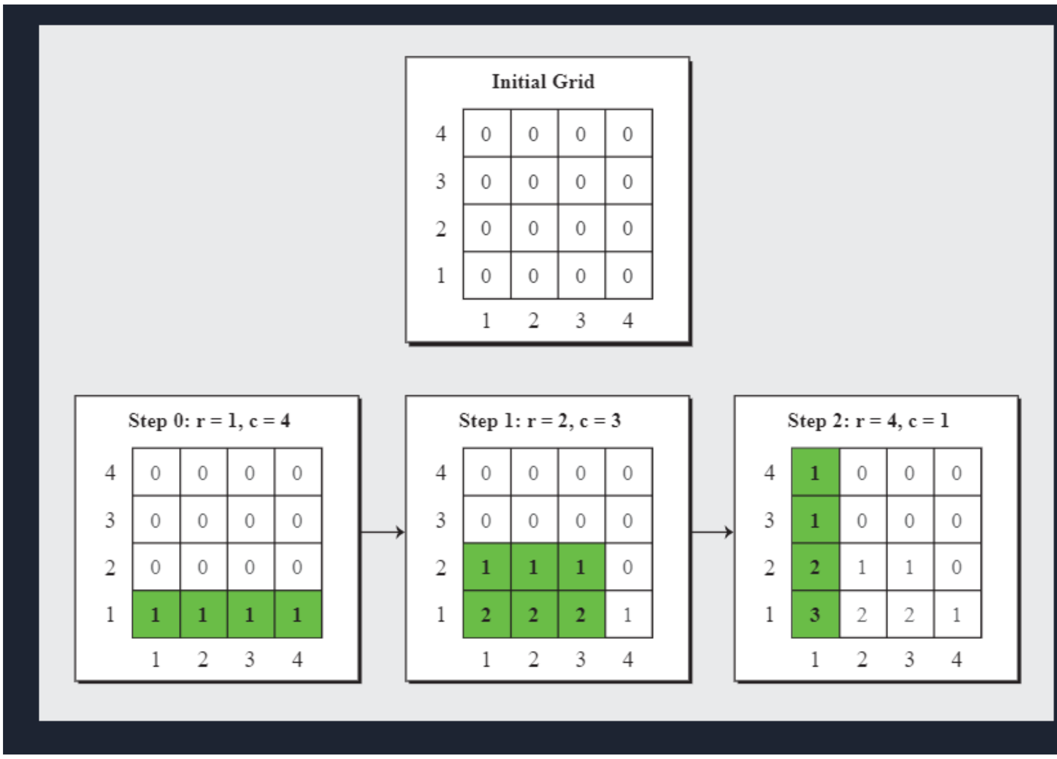
1. Growth in 2 Dimensions

Start with an infinite two dimensional grid filled with zeros, indexed from *(1,1)* at the bottom left corner with coordinates increasing toward the top and right. Given a series of coordinates *(r, c)*, where *r* is the ending row and *c* is the ending column, add *1* to each element in the range from *(1,1)* to *(r, c)* inclusive. Once all coordinates are processed, determine how many cells contain the maximal value in the grid.

Example

upRight = ["1 4", "2 3", "4 1"]

The two space-separated integers within each string represent *r* and *c* respectively. The following diagrams show each iteration starting at zero. The maximal value in the grid is *3*, and there is *1* occurrence at cell *(1, 1)*.



Initial Grid

4	0	0	0	0
3	0	0	0	0
2	0	0	0	0
1	0	0	0	0
	1	2	3	4

Step 0: r = 1, c = 4

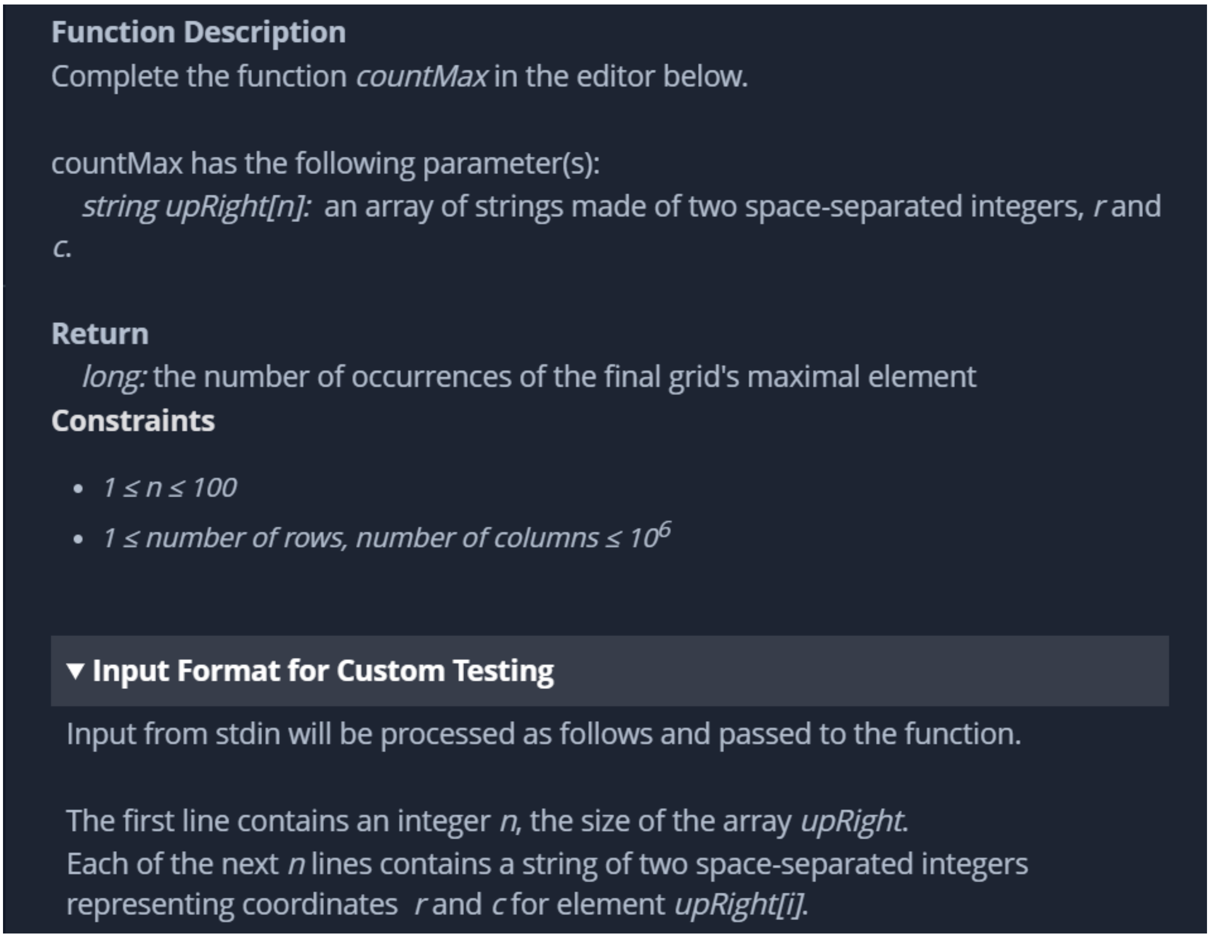
4	0	0	0	0
3	0	0	0	0
2	0	0	0	0
1	**1**	**1**	**1**	**1**
	1	2	3	4

Step 1: r = 2, c = 3

4	0	0	0	0
3	0	0	0	0
2	**1**	**1**	**1**	0
1	**2**	**2**	**2**	1
	1	2	3	4

Step 2: r = 4, c = 1

4	**1**	0	0	0
3	**1**	0	0	0
2	**2**	1	1	0
1	**3**	2	2	1
	1	2	3	4



Function Description

Complete the function *countMax* in the editor below.

countMax has the following parameter(s):

string upRight[n]: an array of strings made of two space-separated integers, *r* and *c*.

Return

long: the number of occurrences of the final grid's maximal element

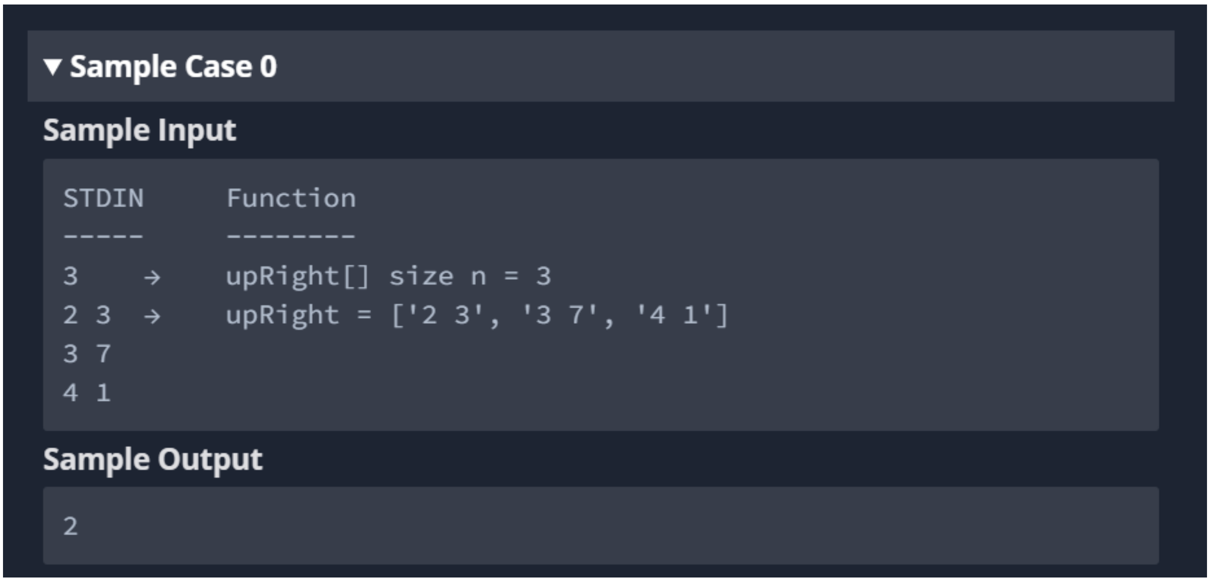
Constraints

- *$1 \le n \le 100$*
- *$1 \le$ number of rows, number of columns $\le 10^6$*

▼ Input Format for Custom Testing

Input from stdin will be processed as follows and passed to the function.

The first line contains an integer *n*, the size of the array *upRight*.
Each of the next *n* lines contains a string of two space-separated integers representing coordinates *r* and *c* for element *upRight[i]*.



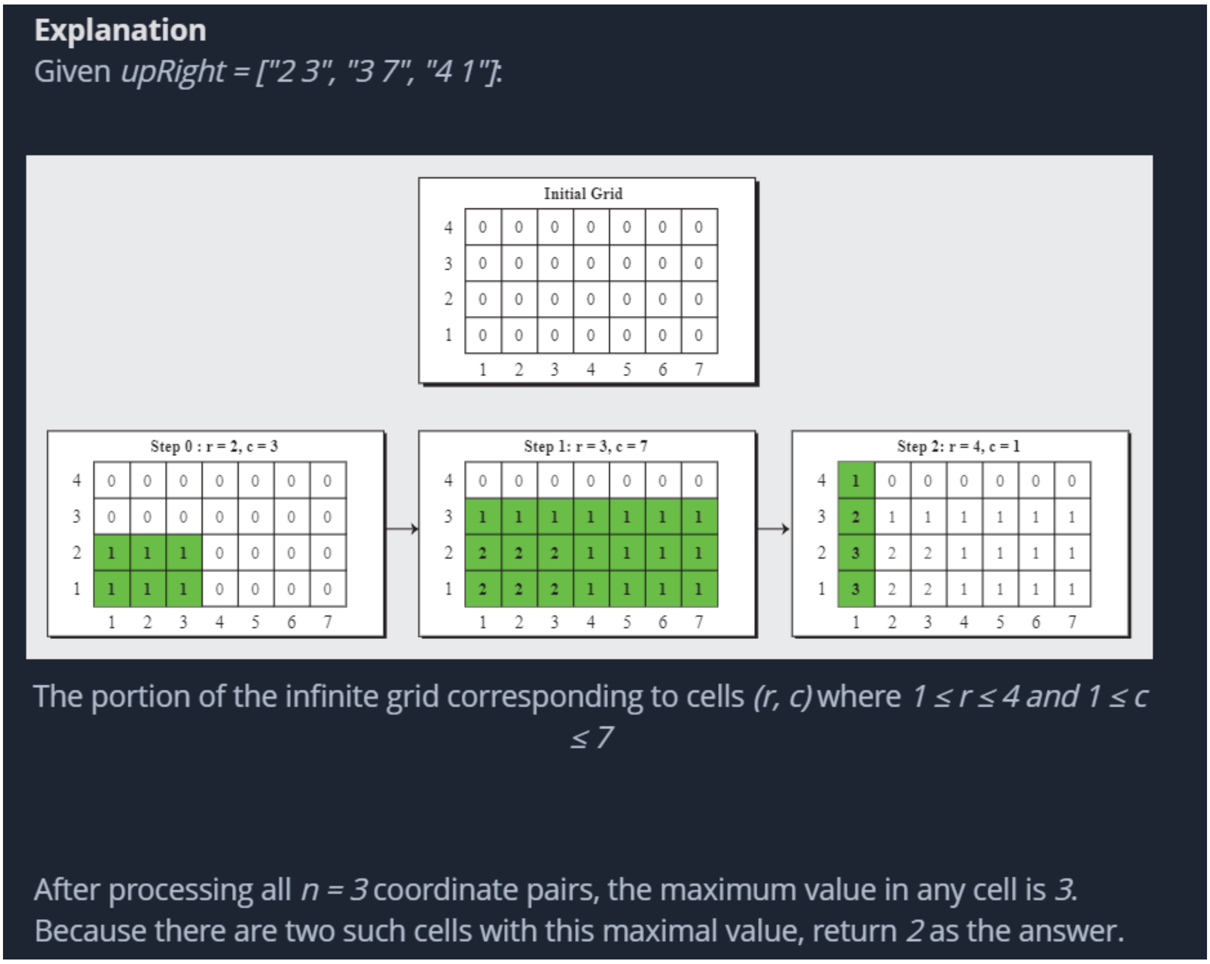
▼ Sample Case 0

Sample Input

```
STDIN     Function
-----     --------
3    →    upRight[] size n = 3
2 3  →    upRight = ['2 3', '3 7', '4 1']
3 7
4 1
```

Sample Output

```
2
```



Explanation

Given *upRight = ["2 3", "3 7", "4 1"]*:

Initial Grid

4	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
	1	2	3	4	5	6	7

Step 0 : r = 2, c = 3

4	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0
2	1	1	1	0	0	0	0
1	1	1	1	0	0	0	0
	1	2	3	4	5	6	7

Step 1: r = 3, c = 7

4	0	0	0	0	0	0	0
3	1	1	1	1	1	1	1
2	2	2	2	1	1	1	1
1	2	2	2	1	1	1	1
	1	2	3	4	5	6	7

Step 2: r = 4, c = 1

4	1	0	0	0	0	0	0
3	2	1	1	1	1	1	1
2	3	2	2	1	1	1	1
1	3	2	2	1	1	1	1
	1	2	3	4	5	6	7

The portion of the infinite grid corresponding to cells *(r, c)* where *1 ≤ r ≤ 4 and 1 ≤ c ≤ 7*

After processing all *n = 3* coordinate pairs, the maximum value in any cell is *3*. Because there are two such cells with this maximal value, return *2* as the answer.