# Find the Treasure (islands)

Giorgio and Luca, while looking through old stuff in the basement of the OIS mansion, found a treasure map! However, the map has faded and is now impossible to understand the exact location of the hidden treasure. The map M can be represented as a two-dimensional grid with R rows and C columns, where each cell can either be land or sea. In particular, for the cell on the i-th row and j-th column, M[i][j] = 1 if the cell represents a portion of land, otherwise M[i][j] = 0 if it is part of the sea.

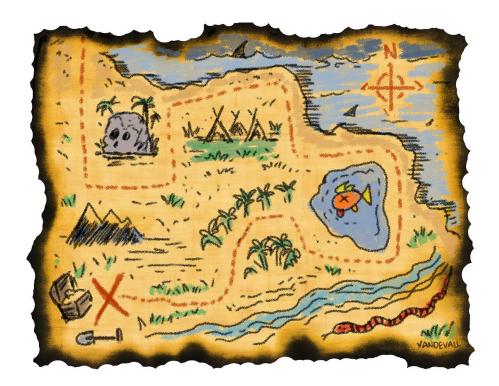


Figure 1: Edoardo's treasure map partially recovered with advanced techniques.

They want to find the treasure at all costs and are ready to start a trip to visit all islands on the map. An island I can be defined as a maximal (i.e., the largest possible) set of land cells, such that:

- For every pair of land cells of I, it is possible to move from one cell to the other by only moving between side-adjacent cells of I;
- No land cell of *I* is on the border of the map (first or last row or column): those are just *peninsulas*, there's no point in wasting time on them.

Help Luca and Giorgio find the number of islands present in the map!

Among the attachments of this task you may find a template file islands.\* with a sample incomplete implementation.

islands Page 1 of 3

### Input

The first line contains two integers R and C, the number of rows and columns. Then R lines follow, each containing C values describing a row of the map. The j-th value of the i-th line represents M[i][j], which is equal to 1 if the cell on the i-th row and j-th column is a portion of land, 0 otherwise.

# Output

You need to write a single line with an integer: the number of islands present in the map.

#### **Constraints**

- $1 \le R, C \le 1000$ .
- M[i][j] is equal to zero or one, for each  $i = 0 \dots R 1, j = 0 \dots C 1$ .

# **Scoring**

Your program will be tested against several test cases grouped in subtasks. In order to obtain the score of a subtask, your program needs to correctly solve all of its test cases.

- Subtask 1 (0 points)	Examples.
- Subtask 2 (10 points)	$R, C \leq 3.$
- Subtask 3 (15 points)	R=3.
- Subtask 4 (15 points)	C=3.
- Subtask 5 (20 points)	There exist no two land cells which are side-adjacent.
- Subtask 6 (25 points)	No land cell is on the border of the map.
- Subtask 7 (15 points)	No additional limitations.

islands Page 2 of 3

### **Examples**

input	output
4 4 0 0 0 0 0 1 0 0 0 0 1 1 0 0 1 1	1
6 6 0 0 0 0 0 0 0 0 1 0 0 1 0 0 1 1 0 1 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0	3
7 7 0 0 0 0 0 0 0 0 0 1 1 1 1 1 0 0 1 0 0 0 1 0 0 1 0 1	2

### **Explanation**

In the first sample case, there is one island formed by the land cell in position (1,1). The set of connected land cells in position (2,2), (2,3), (3,2) and (3,3) is not an island since at least one of those land cells is placed on the border.

In the **second sample case**, there are three islands. The first island is composed of the cells in position (1,1), (2,1), (3,1) and (2,2). The second island is composed of land cells (1,4) and (2,4). The third island is composed only of a single land cell in position (4,3).

In the **third sample case**, there are two islands, one "inside" the other.

islands Page 3 of 3