


[Practice](#) [Compete](#) [Jobs](#) [Rank](#) [Leaderboard](#)

 bor0s

Dashboard > Algorithms > Implementation > Cavity Map

Badge Progress
 ★ ★ ★

Points: 1156.20 Rank: 21006

Cavity Map

 by Gera1d

Problem	Submissions	Leaderboard	Discussions	Editorial	Topics
---------	-------------	-------------	-------------	-----------	--------

You are given a square map of size $n \times n$. Each cell of the map has a value denoting its depth. We will call a cell of the map a *cavity* if and only if this cell is not on the border of the map and each cell adjacent to it has *strictly smaller depth*. Two cells are adjacent if they have a common side (edge).

You need to find all the cavities on the map and depict them with the uppercase character **X**.

Input Format

The first line contains an integer, n , denoting the size of the map. Each of the following n lines contains n positive digits without spaces. Each digit (1-9) denotes the depth of the appropriate area.

Constraints

$$1 \leq n \leq 100$$

Output Format

Output n lines, denoting the resulting map. Each cavity should be replaced with character **X**.

Sample Input

```
4
1112
1912
1892
1234
```

Sample Output

```
1112
1X12
18X2
1234
```

Explanation

The two cells with the depth of 9 fulfill all the conditions of the Cavity definition and have been replaced by X.

[f](#) [t](#) [in](#)

Submissions: [37288](#)

Max Score: 30

Difficulty: Easy



Rate This Challenge:

☆☆☆☆☆

Need Help?

[String Basics](#)

[More](#)

Current Buffer (saved locally, editable)  

C++



```
1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 vector<string> convert_cavity_map(vector<string> grid) {
6     vector<string> new_grid = grid;
7     for (size_t i = 1; i < grid.size() - 1; i++) {
8         for (size_t j = 1; j < grid.size() - 1; j++) {
9             if (grid[i][j-1] < grid[i][j] && grid[i][j+1] < grid[i][j] && grid[i-1][j] < grid[i][j] && grid[i+1][j] <
10 grid[i][j]) {
11                 new_grid[i][j] = 'X';
12             }
13         }
14     }
15     return new_grid;
16 }
17
18 int main(){
19     int n;
20     cin >> n;
21     vector<string> grid(n);
22     for(int grid_i = 0; grid_i < n; grid_i++){
23         cin >> grid[grid_i];
24     }
25     grid = convert_cavity_map(grid);
26     for (int grid_i = 0; grid_i < n; grid_i++) {
27         cout << grid[grid_i] << "\n";
28     }
29     return 0;
30 }
```

Line: 1 Col: 1

[Upload Code as File](#) ☐ Test against custom input

Run Code

Submit Code

Copyright © 2017 HackerRank. All Rights Reserved

Join us on IRC at [#hackerrank](#) on freenode for hugs or bugs.[Contest Calendar](#) | [Interview Prep](#) | [Blog](#) | [Scoring](#) | [Environment](#) | [FAQ](#) | [About Us](#) | [Support](#) | [Careers](#) | [Terms Of Service](#) | [Privacy Policy](#) | [Request a Feature](#)