# 题目

给你一个大小为n x n的整数矩阵 grid 。

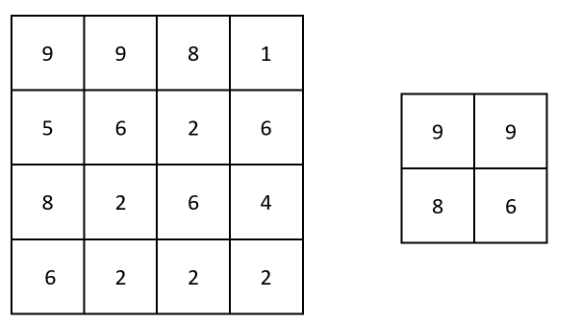
生成一个大小为(n - 2) x (n - 2)的整数矩阵maxLocal，并满足：

maxLocal[i][j]等于grid中以i + 1行和j + 1列为中心的3 x 3矩阵中的最大值 。

换句话说，我们希望找出grid中每个3 x 3矩阵中的最大值。

返回生成的矩阵。

示例 1：



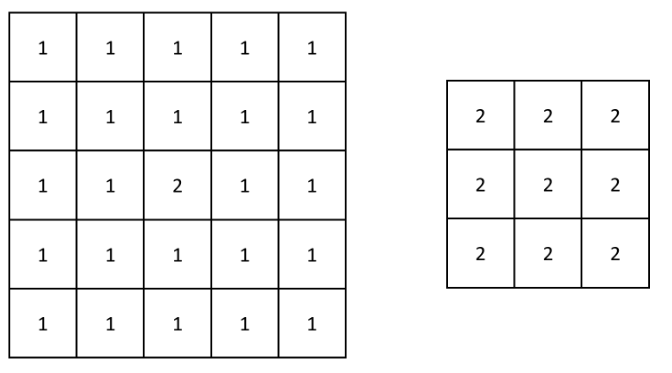
输入：grid = [[9,9,8,1],[5,6,2,6],[8,2,6,4],[6,2,2,2]]

输出：[[9,9],[8,6]]

解释：原矩阵和生成的矩阵如上图所示。

注意，生成的矩阵中，每个值都对应grid中一个相接的3 x 3矩阵的最大值。

示例 2：



输入：grid = [[1,1,1,1,1],[1,1,1,1,1],[1,1,2,1,1],[1,1,1,1,1],[1,1,1,1,1]]

输出：[[2,2,2],[2,2,2],[2,2,2]]

解释：注意，2 包含在 grid 中每个 3 x 3 的矩阵中。

提示：

n == grid.length == grid[i].length

3 <= n <= 100

1 <= grid[i][j] <= 100

# 分析

## 方法一：模拟

思路：

代码：

class Solution {

public:

vector<vector<int>> largestLocal(vector<vector<int>>& grid) {

int n = grid.size();

vector<vector<int>> maxLocal(n - 2, vector<int>(n - 2, 0));

for (int i = 1; i < n - 1; ++i) {

for (int j = 1; j < n - 1; ++j) {

int maxVal = grid[i - 1][j - 1];

for (int x = i - 1; x <= i + 1; ++x) {

for (int y = j - 1; y <= j + 1; ++y) {

maxVal = max(maxVal, grid[x][y]);

}

}

maxLocal[i - 1][j - 1] = maxVal;

}

}

return maxLocal;

}

};