**Problem:**

You are given an m x n binary matrix grid, where 0 represents a sea cell and 1 represents a land cell.

A move consists of walking from one land cell to another adjacent (4-directionally) land cell or walking off the boundary of the grid.

Return the number of land cells in grid for which we cannot walk off the boundary of the grid in any number of moves.

Example 1:

https://assets.leetcode.com/uploads/2021/02/18/enclaves1.jpg

Input: grid = [[0,0,0,0],[1,0,1,0],[0,1,1,0],[0,0,0,0]]

Output: 3

Explanation: There are three 1s that are enclosed by 0s, and one 1 that is not enclosed because its on the boundary.

Example 2:

https://assets.leetcode.com/uploads/2021/02/18/enclaves2.jpg

Input: grid = [[0,1,1,0],[0,0,1,0],[0,0,1,0],[0,0,0,0]]

Output: 0

Explanation: All 1s are either on the boundary or can reach the boundary.

Constraints:

m == grid.length

n == grid[i].length

1 <= m, n <= 500

grid[i][j] is either 0 or 1.

**Input:**

The first line of the input contains a single integer T denoting the number of test cases. The description of T test cases follows.

The first line of each test case contains 4 space separated integers.

The first 3 as Carol’s, Kamala’s and Monica’s number respectively. And the 4th one as N (Fury’s number).

**Output:**

For each test case, print a single line containing one integer – the Nth number of the series.

**Constraints:**

1 ≤ T ≤ 100

1 ≤ N ≤ 20

**Explanation:**

**Example 1:**

Series: 0, 1, 2, (0+1+2=)3, (1+2+3=)6, …

Here the 3rd number is 2

**Example 1:**

Series: 1, 2, 3, (1+2+3=)6, (2+3+6=)11…

Here the 4th number is 6