# Problem 1351: Count Negative Numbers in a Sorted Matrix

## Problem Information

**Difficulty:** Easy
**Acceptance Rate:** 77.85%
**Paid Only:** No
**Tags:** Array, Binary Search, Matrix

## Problem Description

Given a `m x n` matrix `grid` which is sorted in non-increasing order both row-wise and column-wise, return _the number of**negative** numbers in_ `grid`.

**Example 1:**

**Input:** grid = [[4,3,2,-1],[3,2,1,-1],[1,1,-1,-2],[-1,-1,-2,-3]] **Output:** 8 **Explanation:** There are 8 negatives number in the matrix.

**Example 2:**

**Input:** grid = [[3,2],[1,0]] **Output:** 0

**Constraints:**

* `m == grid.length` * `n == grid[i].length` * `1 <= m, n <= 100` * `-100 <= grid[i][j] <= 100`

**Follow up:** Could you find an `O(n + m)` solution?

## Code Snippets

**C++:**

```
class Solution {
public:
int countNegatives(vector<vector<int>>& grid) {

}
};
```

### Java:

```
class Solution {
public int countNegatives(int[][] grid) {

}
}
```

### Python3:

```
class Solution:
def countNegatives(self, grid: List[List[int]]) -> int:
```