The Celebrity Problem

A celebrity is a person who is known to all but **does not know** anyone at a party. A party is being organized by some people. A square matrix **mat** is used to represent people at the party such that if an element of row i and column j is set to 1 it means ith person knows jth person. You need to return the index of the celebrity in the party, if the celebrity does not exist, return **-1**.

Note: Follow 0-based indexing.

Examples:

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

[0 0 0],

[0 1 0]]

Output: 1

Explanation: 0th and 2nd person both know 1. Therefore, 1 is the celebrity.

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

[1 0]]

Output: -1

Explanation: The two people at the party both know each other. None of them is a celebrity.

Expected Time Complexity: O(n^2)
Expected Auxiliary Space: O(1)

Constraints:
1 <= mat.size()<= 3000
0 <= mat[i][j]<= 1

```
class Solution {
  public:
    // Function to find if there is a celebrity in the party
or not.
    int celebrity(vector<vector<int> >& mat) {
        int n = mat.size();
        int candidate = 0;

        for (int i = 1; i < n; ++i) {
            if (mat[candidate][i] == 1) {
                candidate = i;
            }
        }

        for (int i = 0; i < n; ++i) {
            if (i != candidate && (mat[candidate][i] == 1 ||
mat[i][candidate] == 0)) {
                return -1;
            }
        }

        return candidate;
    }
};
```