

# Problem 832: Flipping an Image

## Problem Information

**Difficulty:** Easy

**Acceptance Rate:** 83.35%

**Paid Only:** No

**Tags:** Array, Two Pointers, Bit Manipulation, Matrix, Simulation

## Problem Description

Given an  $n \times n$  binary matrix `image` , flip the image **horizontally** , then invert it, and return \_the resulting image\_.

To flip an image horizontally means that each row of the image is reversed.

\* For example, flipping `[1,1,0]` horizontally results in `[0,1,1]` .

To invert an image means that each `0` is replaced by `1` , and each `1` is replaced by `0` .

\* For example, inverting `[0,1,1]` results in `[1,0,0]` .

**Example 1:**

**Input:** image = [[1,1,0],[1,0,1],[0,0,0]] **Output:** [[1,0,0],[0,1,0],[1,1,1]] **Explanation:**  
First reverse each row: [[0,1,1],[1,0,1],[0,0,0]]. Then, invert the image: [[1,0,0],[0,1,0],[1,1,1]]

**Example 2:**

**Input:** image = [[1,1,0,0],[1,0,0,1],[0,1,1,1],[1,0,1,0]] **Output:**  
[[1,1,0,0],[0,1,1,0],[0,0,0,1],[1,0,1,0]] **Explanation:** First reverse each row:  
[[0,0,1,1],[1,0,0,1],[1,1,1,0],[0,1,0,1]]. Then invert the image:  
[[1,1,0,0],[0,1,1,0],[0,0,0,1],[1,0,1,0]]

**Constraints:**

\*  $n == \text{image.length}$  \*  $n == \text{image[i].length}$  \*  $1 \leq n \leq 20$  \* `image[i][j]` is either `0` or `1` .

## Code Snippets

### C++:

```
class Solution {
public:
    vector<vector<int>> flipAndInvertImage(vector<vector<int>>& image) {
        }
    };
}
```

### Java:

```
class Solution {
    public int[][][] flipAndInvertImage(int[][][] image) {
        }
    }
}
```

### Python3:

```
class Solution:
    def flipAndInvertImage(self, image: List[List[int]]) -> List[List[int]]:
```