Author

[danilaml](https://codefights.com/profile/danilaml)

https://codefights.com/img/coins_new.png2000

You are given a number N and a two-dimensional array signals, which contains N equal-sized arrays of0s and 1s.

Return an array of the same length as the elements ofN, such that its ith element is the one that appears most frequently at the ith position of signals' elements.

**Example:**

errorProcessing(3, [[1,1,0], [1,0,0], [0,1,1])) = [1,1,0]

At the 1st position there're two 1s and one 0, so in the 1st element of the resulting array is 1.  
At the 2nd position there're two 1s and one 0, so in the 2nd element of the resulting array is 1.  
At the 3rd position there're two 0s and one 1, so in the 3rd element of the resulting array is 0.

* **[input] integer N**
  + A positive odd number.
* **[input] array.array.integer signals**
  + Array of N non-empty elements.
* **[output] array.integer**
  + The resulting array.

<https://codefights.com/challenge/f5P4XTrswLvzx6vsr>

std::vector<int> errorProcessing(int N, std::vector<std::vector<int>> signals) {

std::vector<int> ans;

    for (int j = 0; j < signals[0].size(); j++)

    {

        int unos = 0;

        for (int i = 0; i < N; i++)

        {

            unos += signals[i][j];

        }

        int ceros = N - unos;

        if (unos > ceros)

            ans.push\_back(1);

        else

            ans.push\_back(0);

    }

    return ans;

}