Centro Educativo Técnico Laboral Kinal

Taller III

Prof. Josué Noj

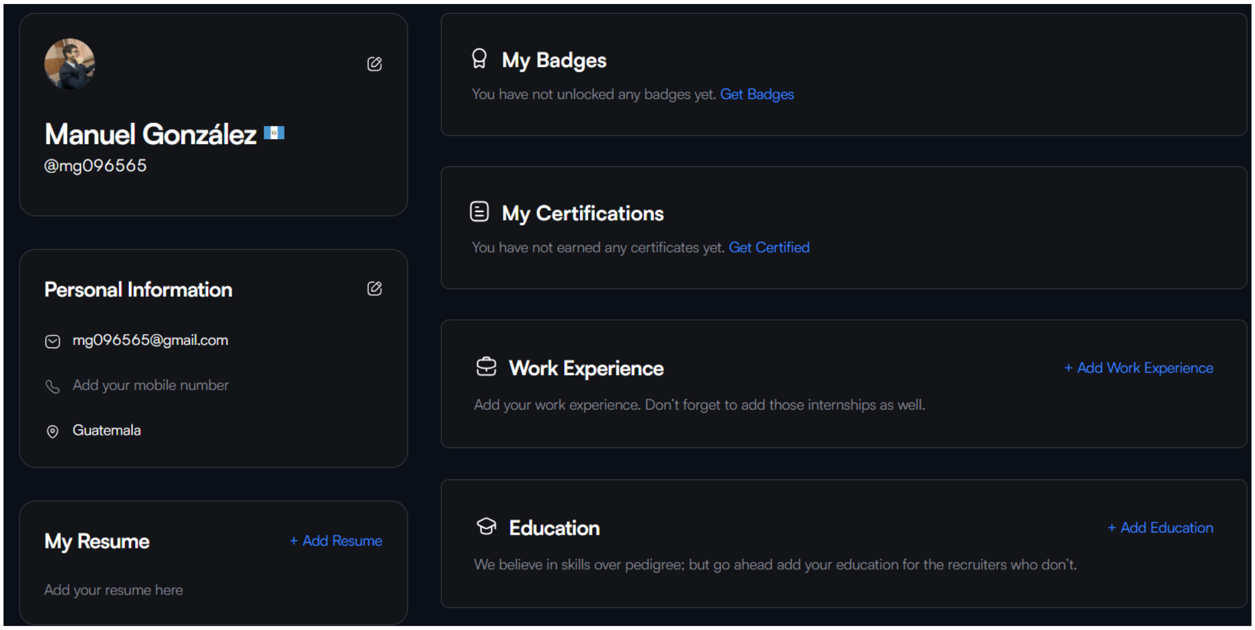
EJERCICIOS REALIZADOS DENTRO DE HACKERRANK

Manuel Eduardo González Avalos

2021391

IN6AM

PERFIL



Manuel González

@mg096565

Personal Information

mg096565@gmail.com

Add your mobile number

Guatemala

My Resume

+ Add Resume

Add your resume here

My Badges

You have not unlocked any badges yet. Get Badges

My Certifications

You have not earned any certificates yet. Get Certified

Work Experience

+ Add Work Experience

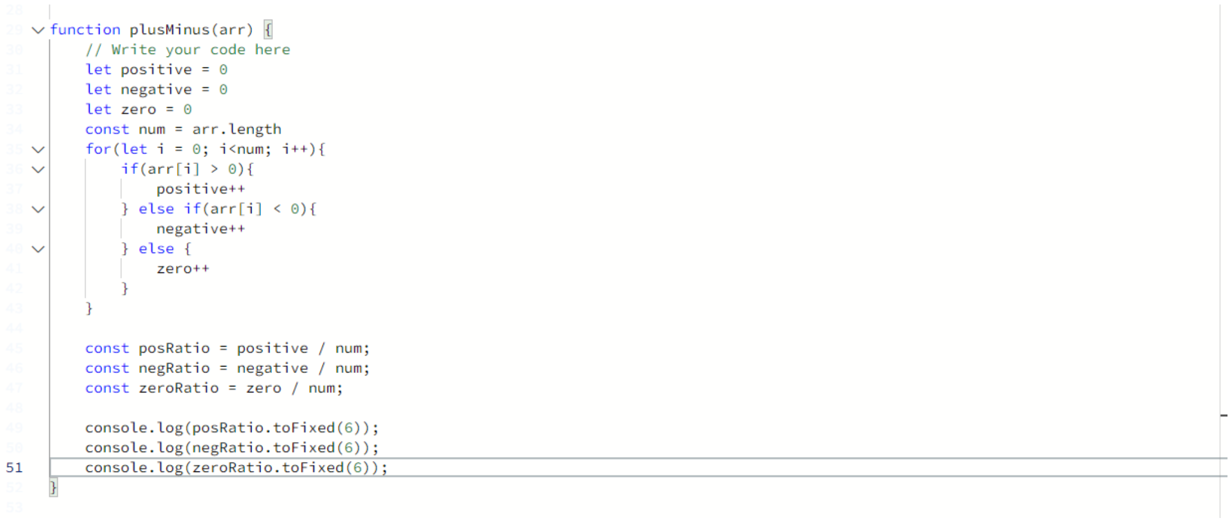
Add your work experience. Don't forget to add those internships as well.

Education

+ Add Education

We believe in skills over pedigree; but go ahead add your education for the recruiters who don't.

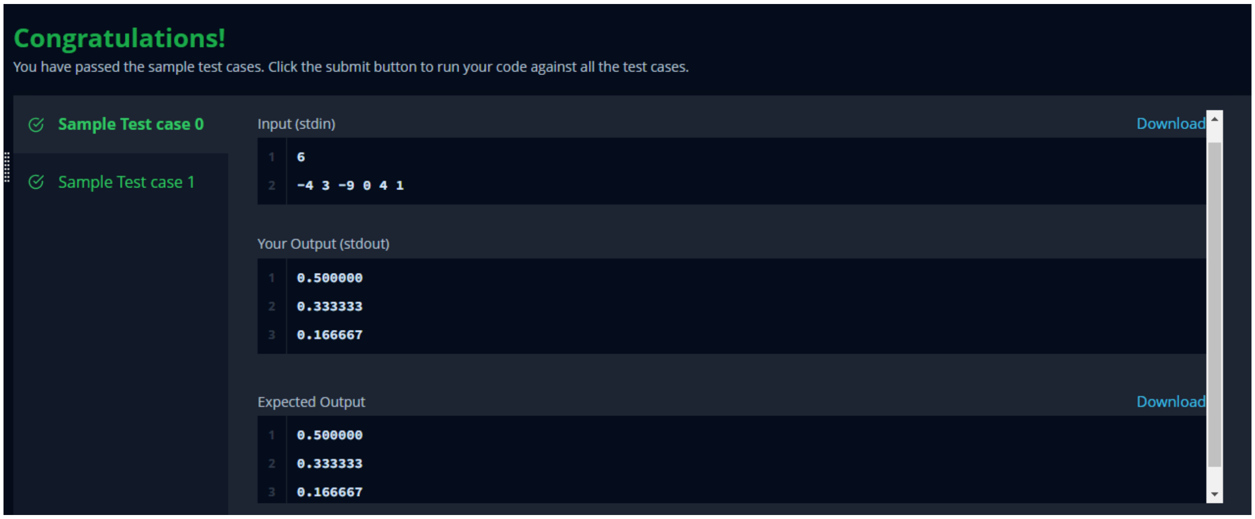
EJERCICIO 1



```
function plusMinus(arr) {
    // Write your code here
    let positive = 0
    let negative = 0
    let zero = 0
    const num = arr.length
    for(let i = 0; i<num; i++){
        if(arr[i] > 0){
            positive++
        } else if(arr[i] < 0){
            negative++
        } else {
            zero++
        }
    }

    const posRatio = positive / num;
    const negRatio = negative / num;
    const zeroRatio = zero / num;

    console.log(posRatio.toFixed(6));
    console.log(negRatio.toFixed(6));
    console.log(zeroRatio.toFixed(6));
}
```



Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Input (stdin)

```
6
-4 3 -9 0 4 1
```

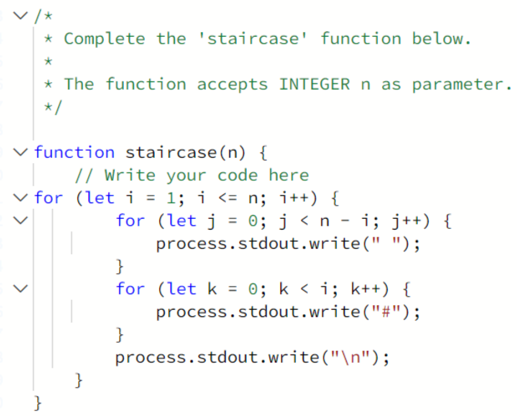
Your Output (stdout)

```
0.500000
0.333333
0.166667
```

Expected Output

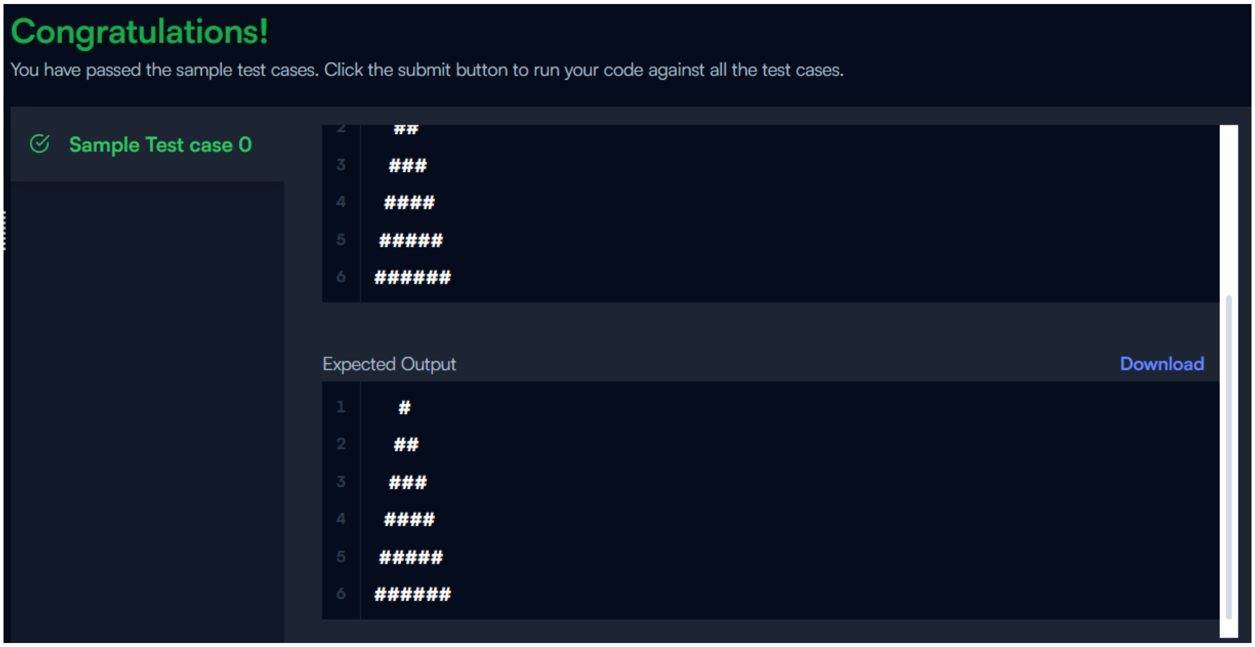
```
0.500000
0.333333
0.166667
```

EJERCICIO 2



```
/*
 * Complete the 'staircase' function below.
 *
 * The function accepts INTEGER n as parameter.
 */

function staircase(n) {
    // Write your code here
for (let i = 1; i <= n; i++) {
        for (let j = 0; j < n - i; j++) {
            process.stdout.write(" ");
        }
        for (let k = 0; k < i; k++) {
            process.stdout.write("#");
        }
        process.stdout.write("\n");
    }
}
```



Congratulations!

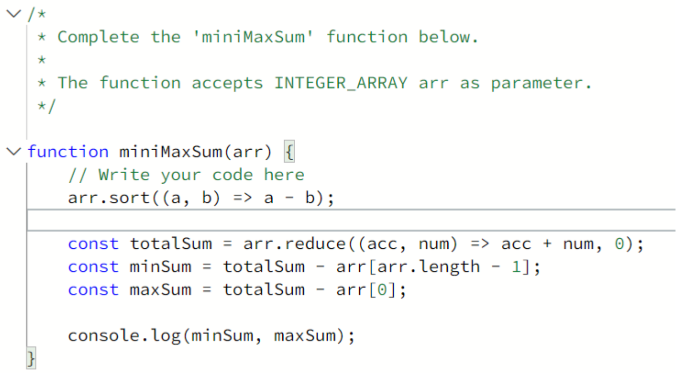
You have passed the sample test cases. Click the submit button to run your code against all the test cases.

```
    ##
   ###
  ####
 #####
######
```

Expected Output

```
     #
    ##
   ###
  ####
 #####
######
```

EJERCICIO 3

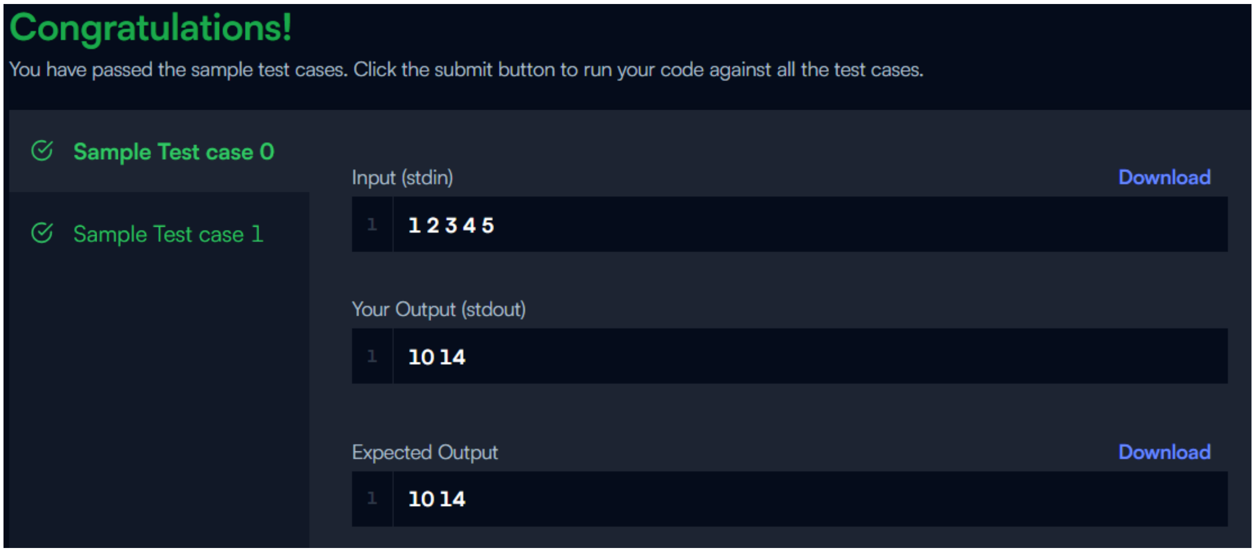


```javascript
/*
 * Complete the 'miniMaxSum' function below.
 *
 * The function accepts INTEGER_ARRAY arr as parameter.
 */

function miniMaxSum(arr) {
    // Write your code here
    arr.sort((a, b) => a - b);

    const totalSum = arr.reduce((acc, num) => acc + num, 0);
    const minSum = totalSum - arr[arr.length - 1];
    const maxSum = totalSum - arr[0];

    console.log(minSum, maxSum);
}
```



Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Input (stdin)

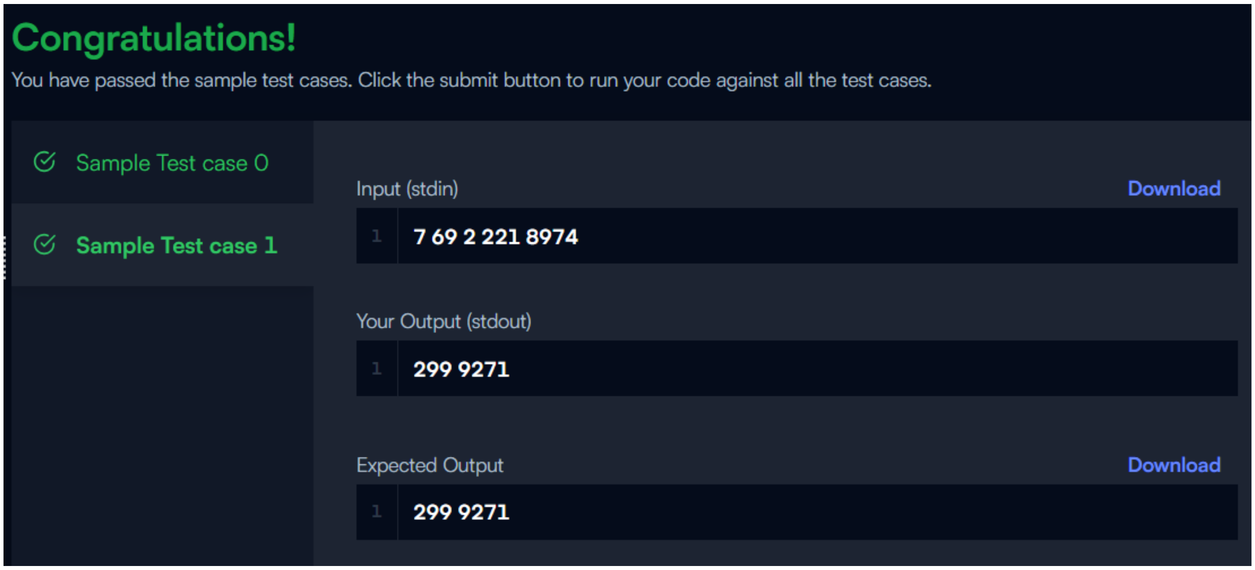
```
1 2 3 4 5
```

Your Output (stdout)

```
10 14
```

Expected Output

```
10 14
```



Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Input (stdin)

```
7 69 2 221 8974
```

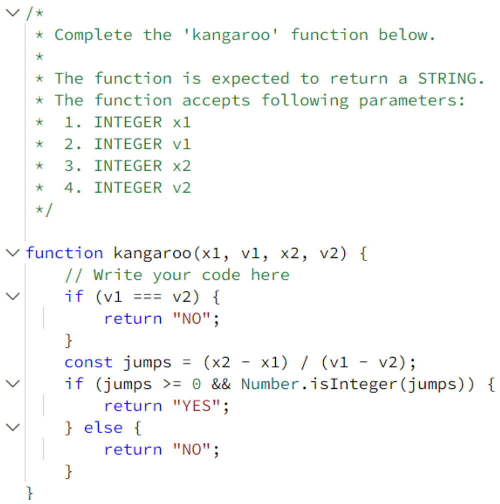
Your Output (stdout)

```
299 9271
```

Expected Output

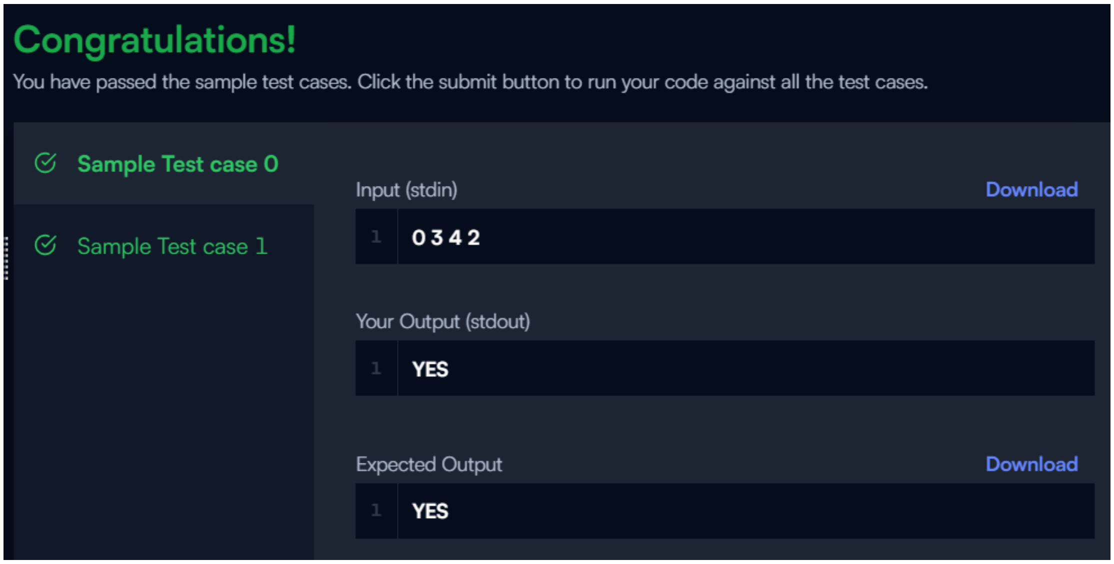
```
299 9271
```

EJERCICIO 4



```
/*
 * Complete the 'kangaroo' function below.
 *
 * The function is expected to return a STRING.
 * The function accepts following parameters:
 *  1. INTEGER x1
 *  2. INTEGER v1
 *  3. INTEGER x2
 *  4. INTEGER v2
 */

function kangaroo(x1, v1, x2, v2) {
    // Write your code here
    if (v1 === v2) {
        return "NO";
    }
    const jumps = (x2 - x1) / (v1 - v2);
    if (jumps >= 0 && Number.isInteger(jumps)) {
        return "YES";
    } else {
        return "NO";
    }
}
```



Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Input (stdin)

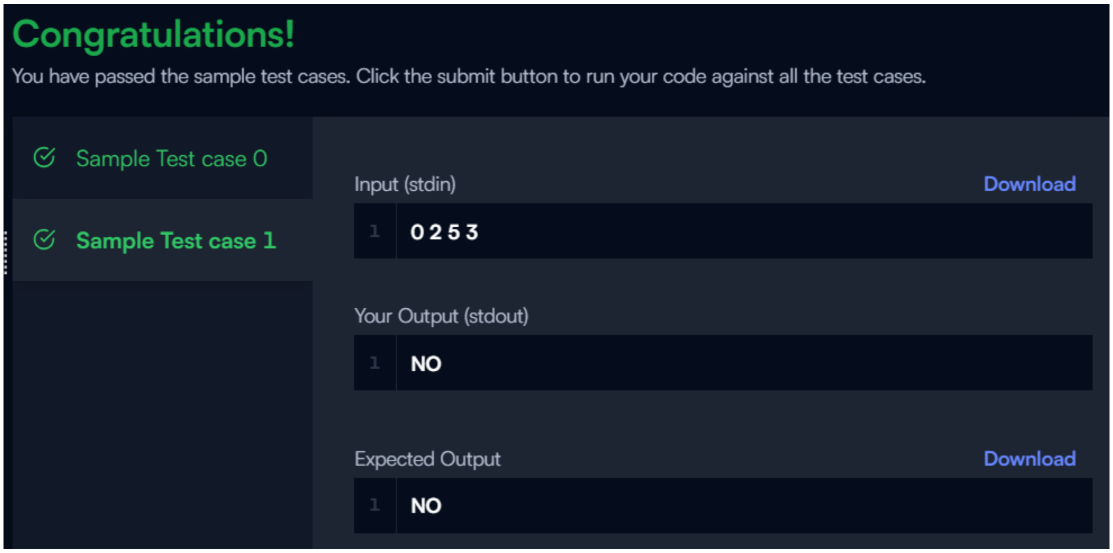
```
0 3 4 2
```

Your Output (stdout)

```
YES
```

Expected Output

```
YES
```



Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Input (stdin)

```
0 2 5 3
```

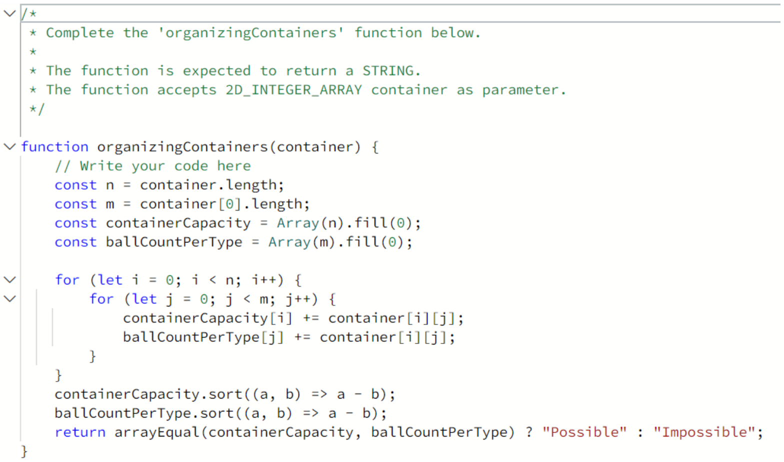
Your Output (stdout)

```
NO
```

Expected Output

```
NO
```

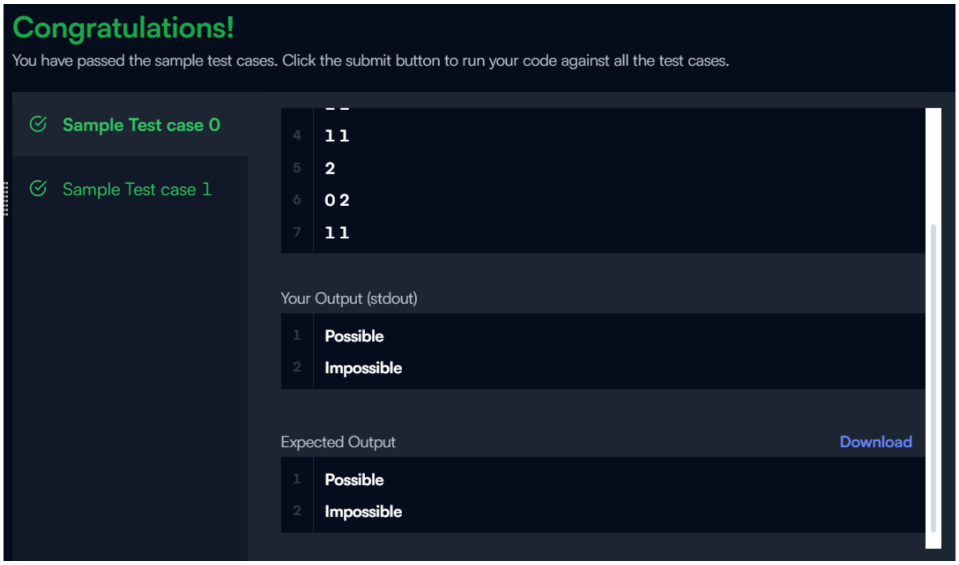
EJERCICIO 5



```
/*
 * Complete the 'organizingContainers' function below.
 *
 * The function is expected to return a STRING.
 * The function accepts 2D_INTEGER_ARRAY container as parameter.
 */

function organizingContainers(container) {
    // Write your code here
    const n = container.length;
    const m = container[0].length;
    const containerCapacity = Array(n).fill(0);
    const ballCountPerType = Array(m).fill(0);

    for (let i = 0; i < n; i++) {
        for (let j = 0; j < m; j++) {
            containerCapacity[i] += container[i][j];
            ballCountPerType[j] += container[i][j];
        }
    }
    containerCapacity.sort((a, b) => a - b);
    ballCountPerType.sort((a, b) => a - b);
    return arrayEqual(containerCapacity, ballCountPerType) ? "Possible" : "Impossible";
}
```



Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ **Sample Test case 0**

✓ Sample Test case 1

```
1 1
2
0 2
1 1
```

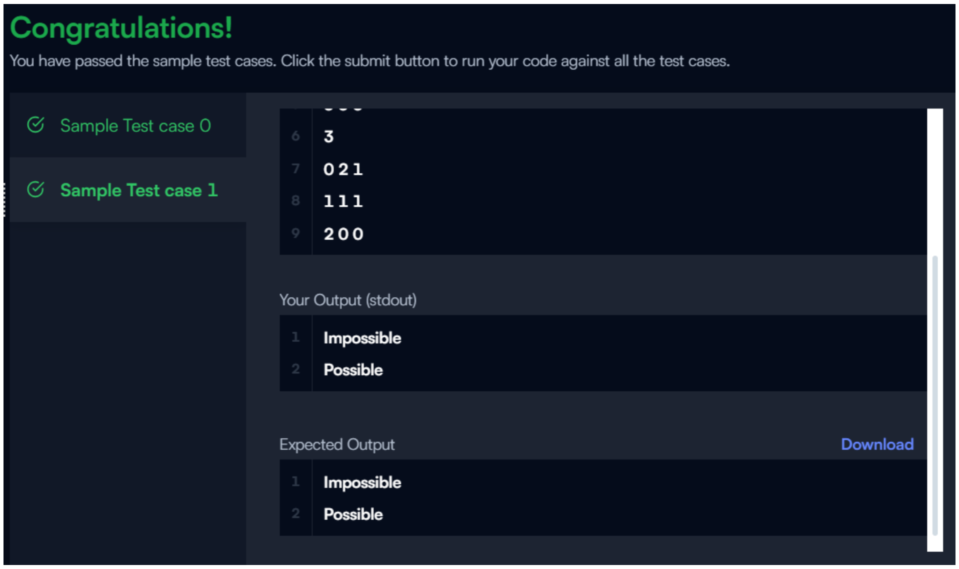
Your Output (stdout)

```
Possible
Impossible
```

Expected Output

Download

```
Possible
Impossible
```



Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

✓ Sample Test case 0

✓ **Sample Test case 1**

```
3
0 2 1
1 1 1
2 0 0
```

Your Output (stdout)

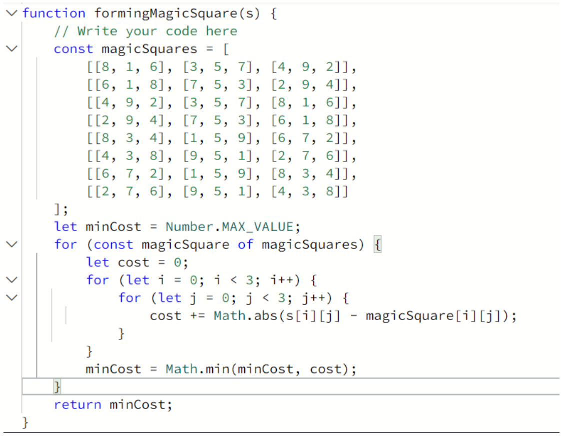
```
Impossible
Possible
```

Expected Output

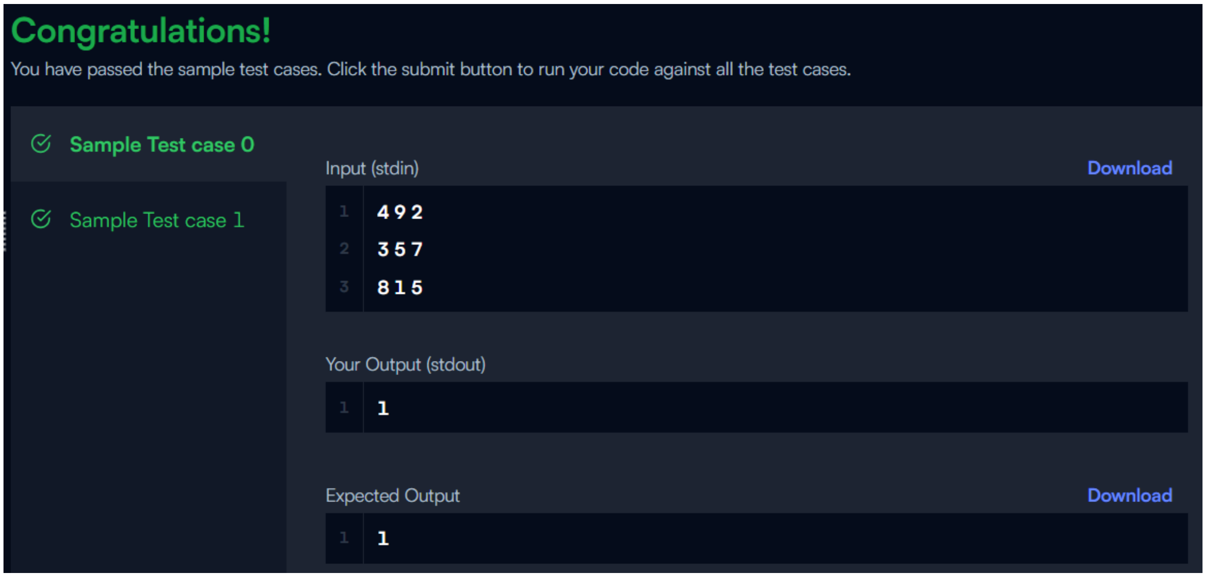
Download

```
Impossible
Possible
```

EJERCICIO 6



```javascript
function formingMagicSquare(s) {
    // Write your code here
    const magicSquares = [
        [[8, 1, 6], [3, 5, 7], [4, 9, 2]],
        [[6, 1, 8], [7, 5, 3], [2, 9, 4]],
        [[4, 9, 2], [3, 5, 7], [8, 1, 6]],
        [[2, 9, 4], [7, 5, 3], [6, 1, 8]],
        [[8, 3, 4], [1, 5, 9], [6, 7, 2]],
        [[4, 3, 8], [9, 5, 1], [2, 7, 6]],
        [[6, 7, 2], [1, 5, 9], [8, 3, 4]],
        [[2, 7, 6], [9, 5, 1], [4, 3, 8]]
    ];
    let minCost = Number.MAX_VALUE;
    for (const magicSquare of magicSquares) {
        let cost = 0;
        for (let i = 0; i < 3; i++) {
            for (let j = 0; j < 3; j++) {
                cost += Math.abs(s[i][j] - magicSquare[i][j]);
            }
        }
        minCost = Math.min(minCost, cost);
    }
    return minCost;
}
```



Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Input (stdin)

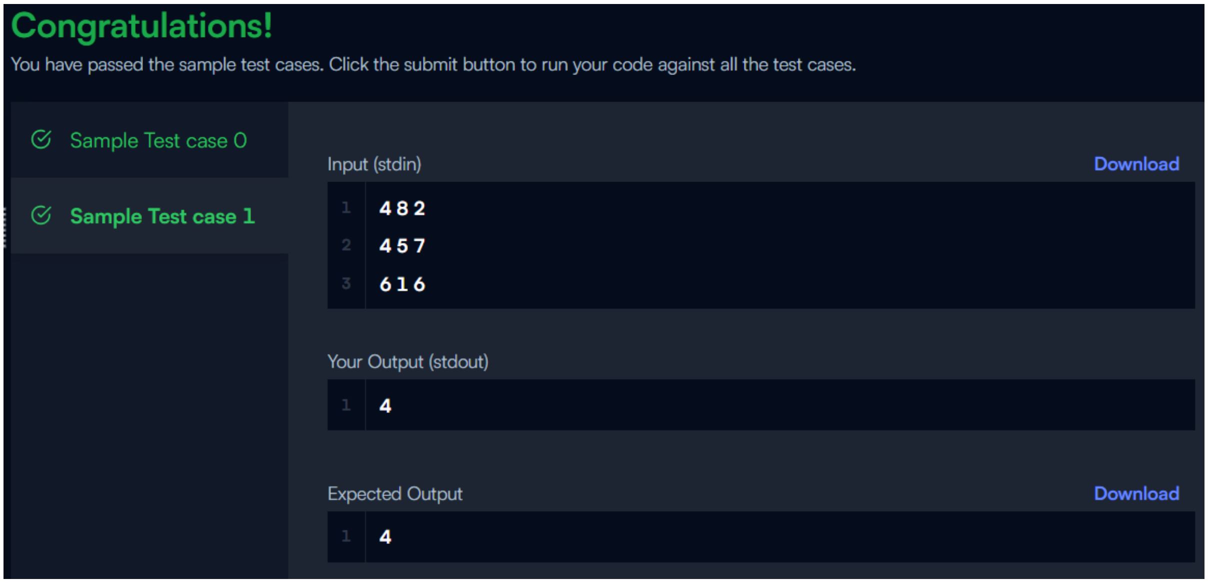
```
4 9 2
3 5 7
8 1 5
```

Your Output (stdout)

```
1
```

Expected Output

```
1
```



Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Input (stdin)

```
4 8 2
4 5 7
6 1 6
```

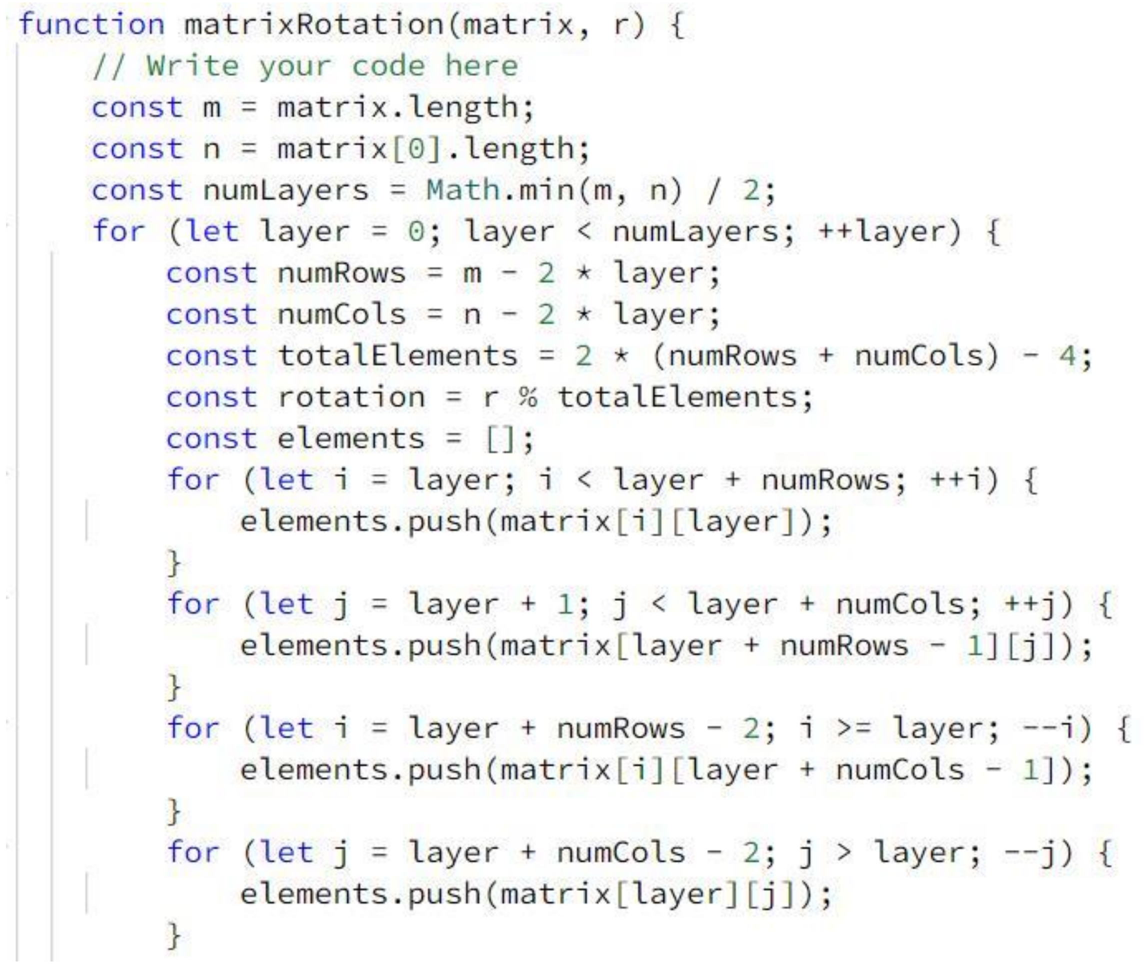
Your Output (stdout)

```
4
```

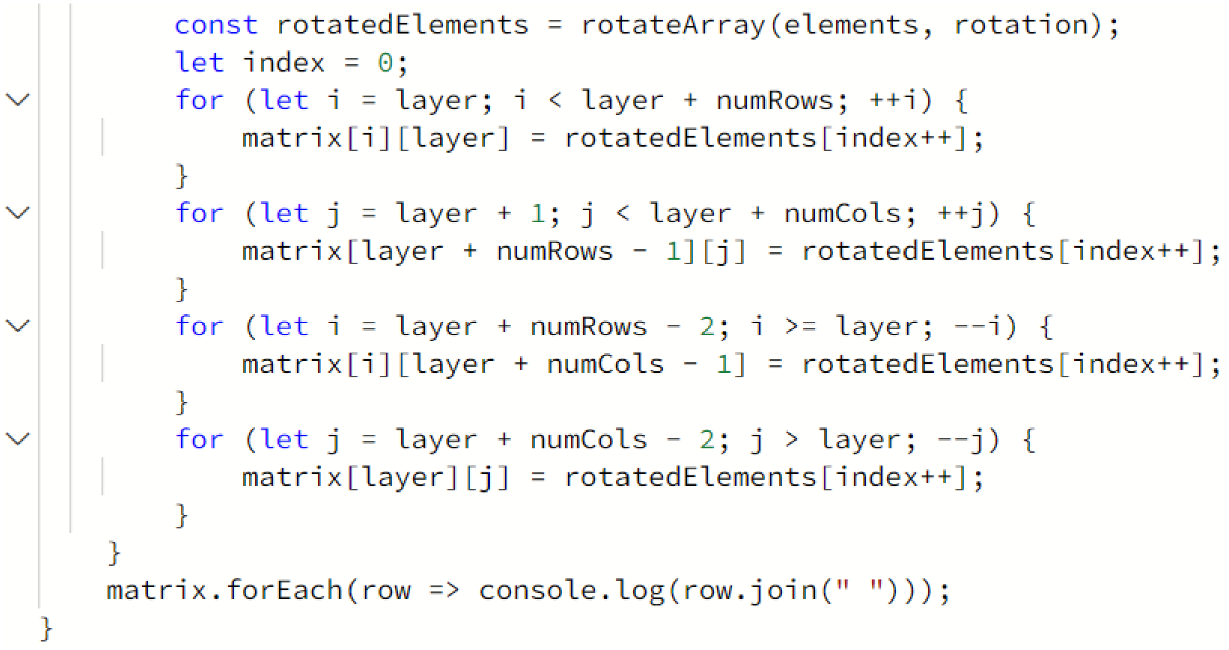
Expected Output

```
4
```

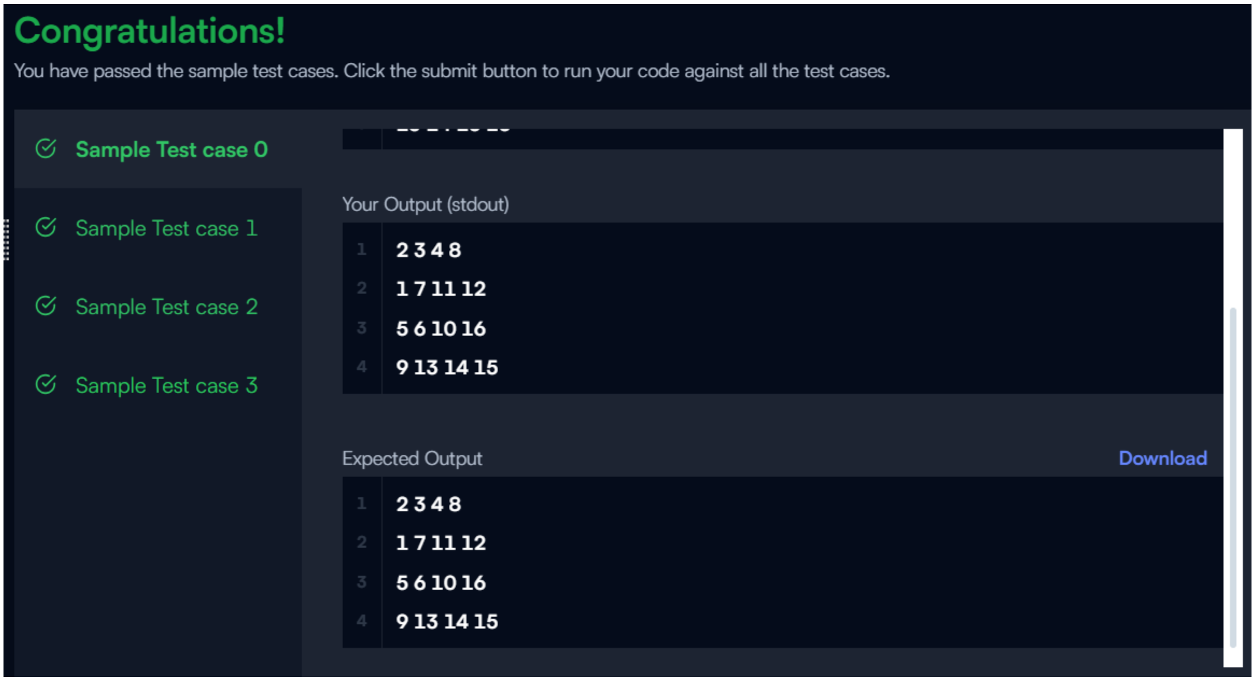
EJERCICIO 7



```
function matrixRotation(matrix, r) {
    // Write your code here
    const m = matrix.length;
    const n = matrix[0].length;
    const numLayers = Math.min(m, n) / 2;
    for (let layer = 0; layer < numLayers; ++layer) {
        const numRows = m - 2 * layer;
        const numCols = n - 2 * layer;
        const totalElements = 2 * (numRows + numCols) - 4;
        const rotation = r % totalElements;
        const elements = [];
        for (let i = layer; i < layer + numRows; ++i) {
            elements.push(matrix[i][layer]);
        }
        for (let j = layer + 1; j < layer + numCols; ++j) {
            elements.push(matrix[layer + numRows - 1][j]);
        }
        for (let i = layer + numRows - 2; i >= layer; --i) {
            elements.push(matrix[i][layer + numCols - 1]);
        }
        for (let j = layer + numCols - 2; j > layer; --j) {
            elements.push(matrix[layer][j]);
        }
```



```
        const rotatedElements = rotateArray(elements, rotation);
        let index = 0;
        for (let i = layer; i < layer + numRows; ++i) {
            matrix[i][layer] = rotatedElements[index++];
        }
        for (let j = layer + 1; j < layer + numCols; ++j) {
            matrix[layer + numRows - 1][j] = rotatedElements[index++];
        }
        for (let i = layer + numRows - 2; i >= layer; --i) {
            matrix[i][layer + numCols - 1] = rotatedElements[index++];
        }
        for (let j = layer + numCols - 2; j > layer; --j) {
            matrix[layer][j] = rotatedElements[index++];
        }
    }
    matrix.forEach(row => console.log(row.join(" ")));
}
```



Congratulations!

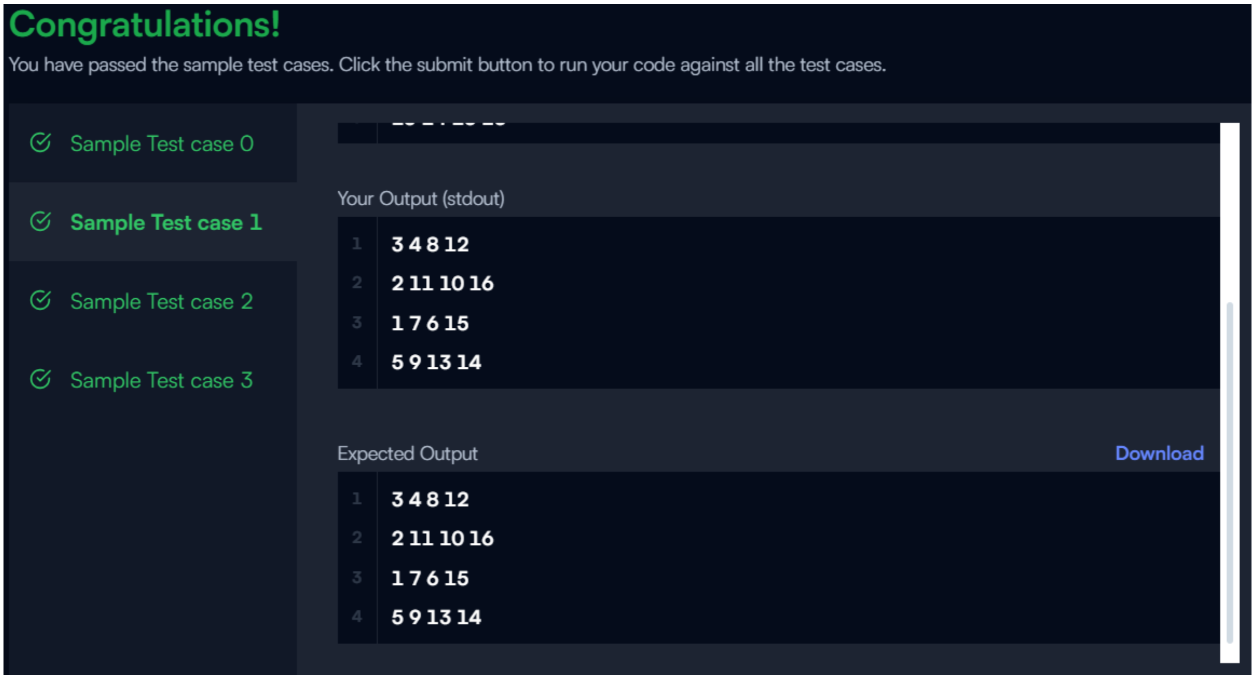
You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Your Output (stdout)

```
2 3 4 8
1 7 11 12
5 6 10 16
9 13 14 15
```

Expected Output

```
2 3 4 8
1 7 11 12
5 6 10 16
9 13 14 15
```



Congratulations!

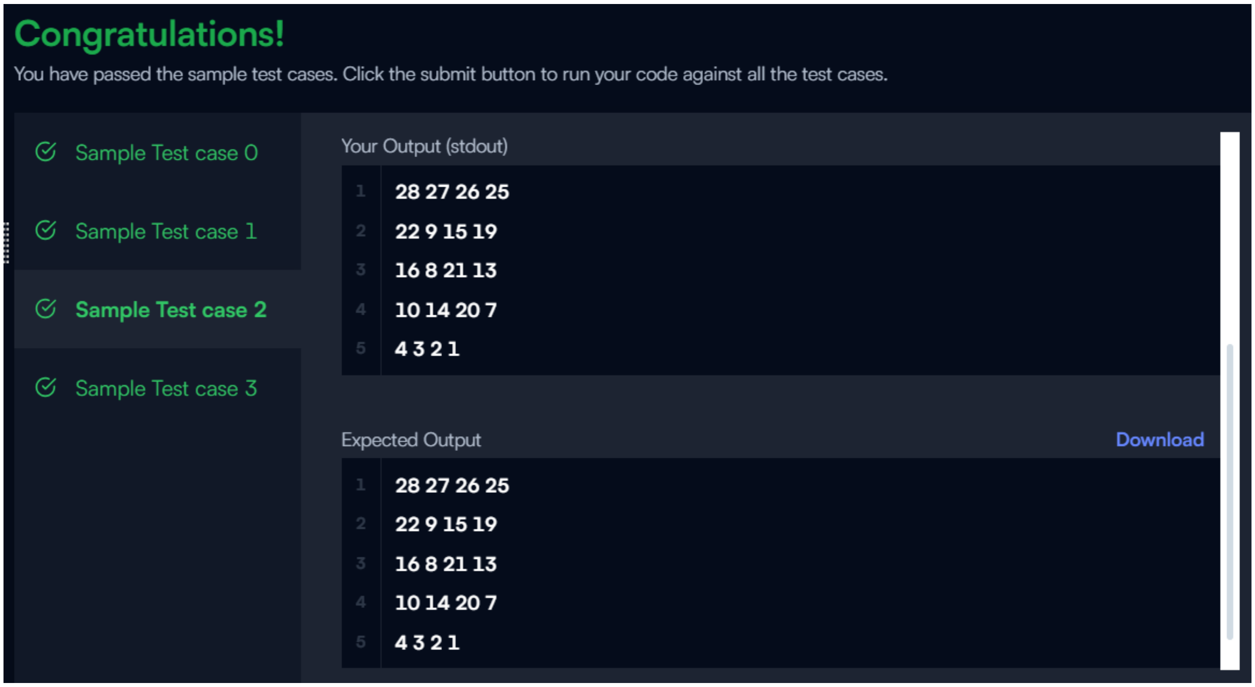
You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Your Output (stdout)

```
3 4 8 12
2 11 10 16
1 7 6 15
5 9 13 14
```

Expected Output

```
3 4 8 12
2 11 10 16
1 7 6 15
5 9 13 14
```



Congratulations!

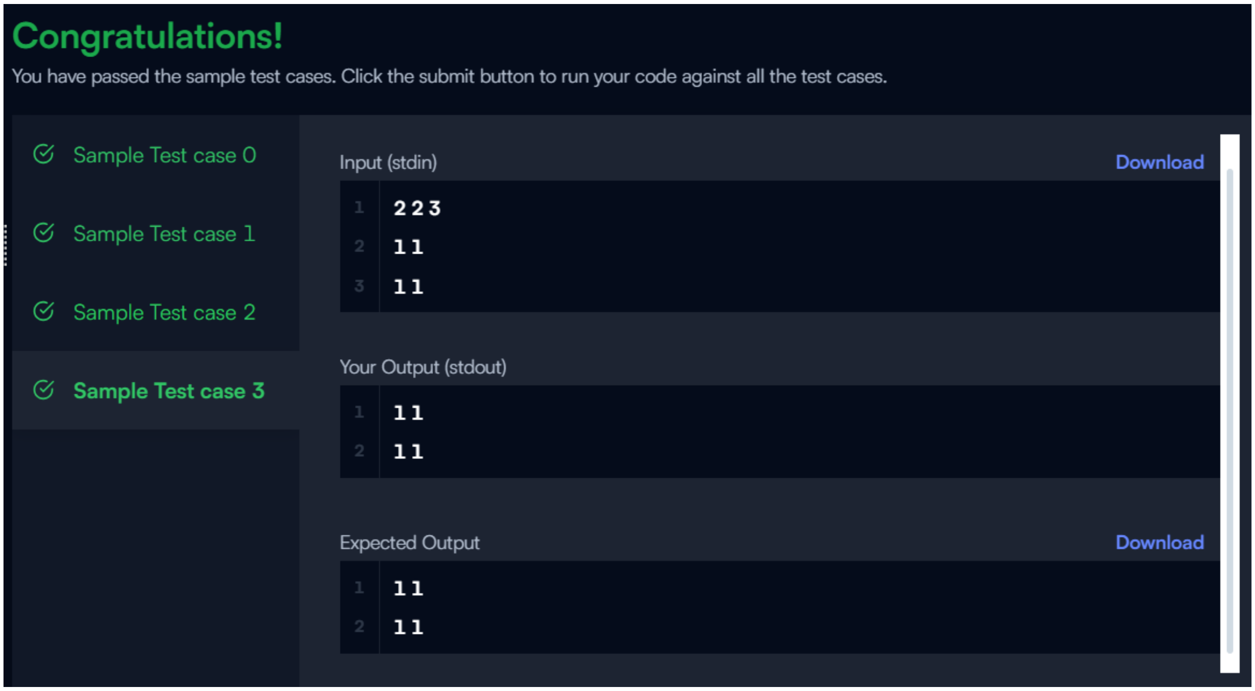
You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Your Output (stdout)

```
28 27 26 25
22 9 15 19
16 8 21 13
10 14 20 7
4 3 2 1
```

Expected Output

```
28 27 26 25
22 9 15 19
16 8 21 13
10 14 20 7
4 3 2 1
```



Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Input (stdin)

```
2 2 3
1 1
1 1
```

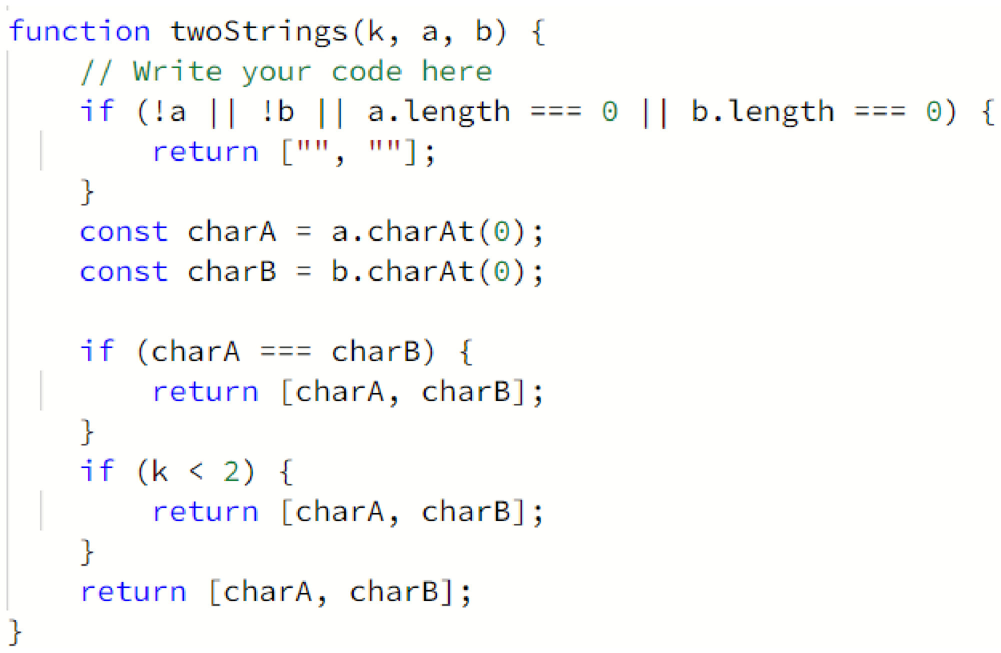
Your Output (stdout)

```
1 1
1 1
```

Expected Output

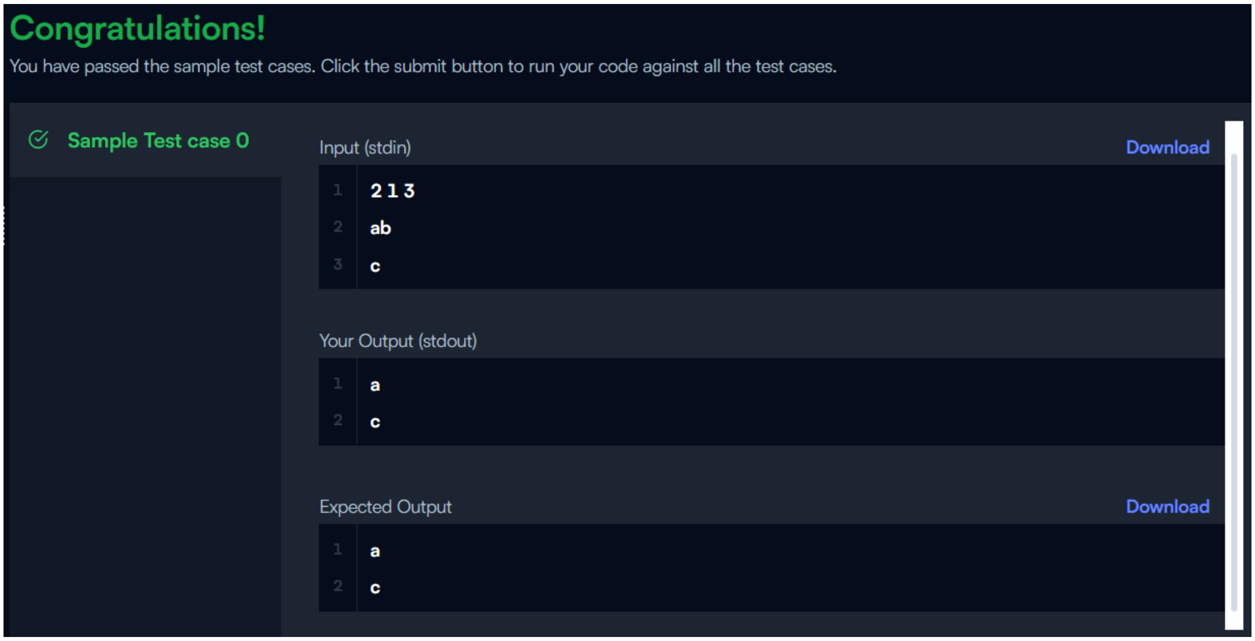
```
1 1
1 1
```

EJERCICIO 8



```
function twoStrings(k, a, b) {
    // Write your code here
    if (!a || !b || a.length === 0 || b.length === 0) {
        return ["", ""];
    }
    const charA = a.charAt(0);
    const charB = b.charAt(0);

    if (charA === charB) {
        return [charA, charB];
    }
    if (k < 2) {
        return [charA, charB];
    }
    return [charA, charB];
}
```



Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Input (stdin)

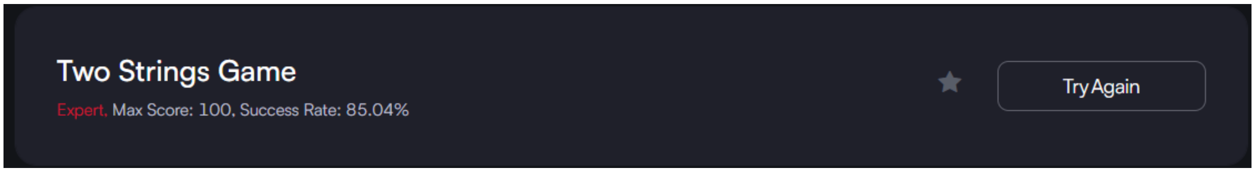
```
213
ab
c
```

Your Output (stdout)

```
a
c
```

Expected Output

```
a
c
```



Two Strings Game

Expert, Max Score: 100, Success Rate: 85.04%