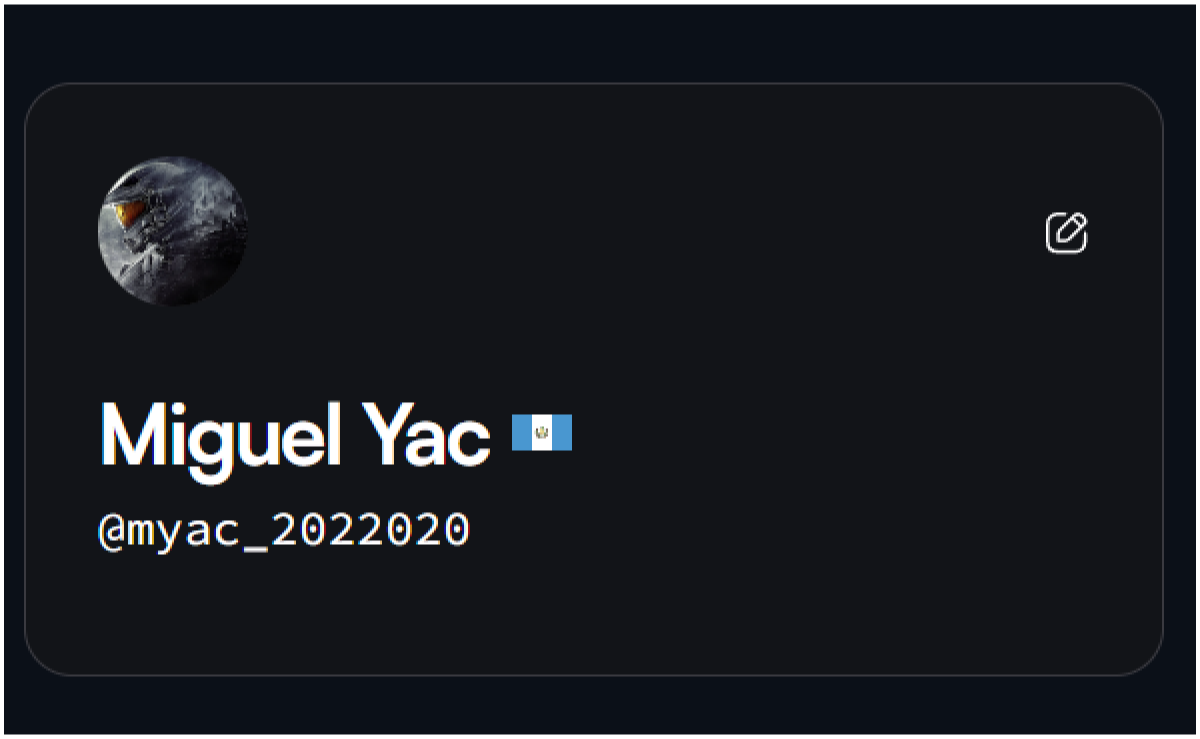
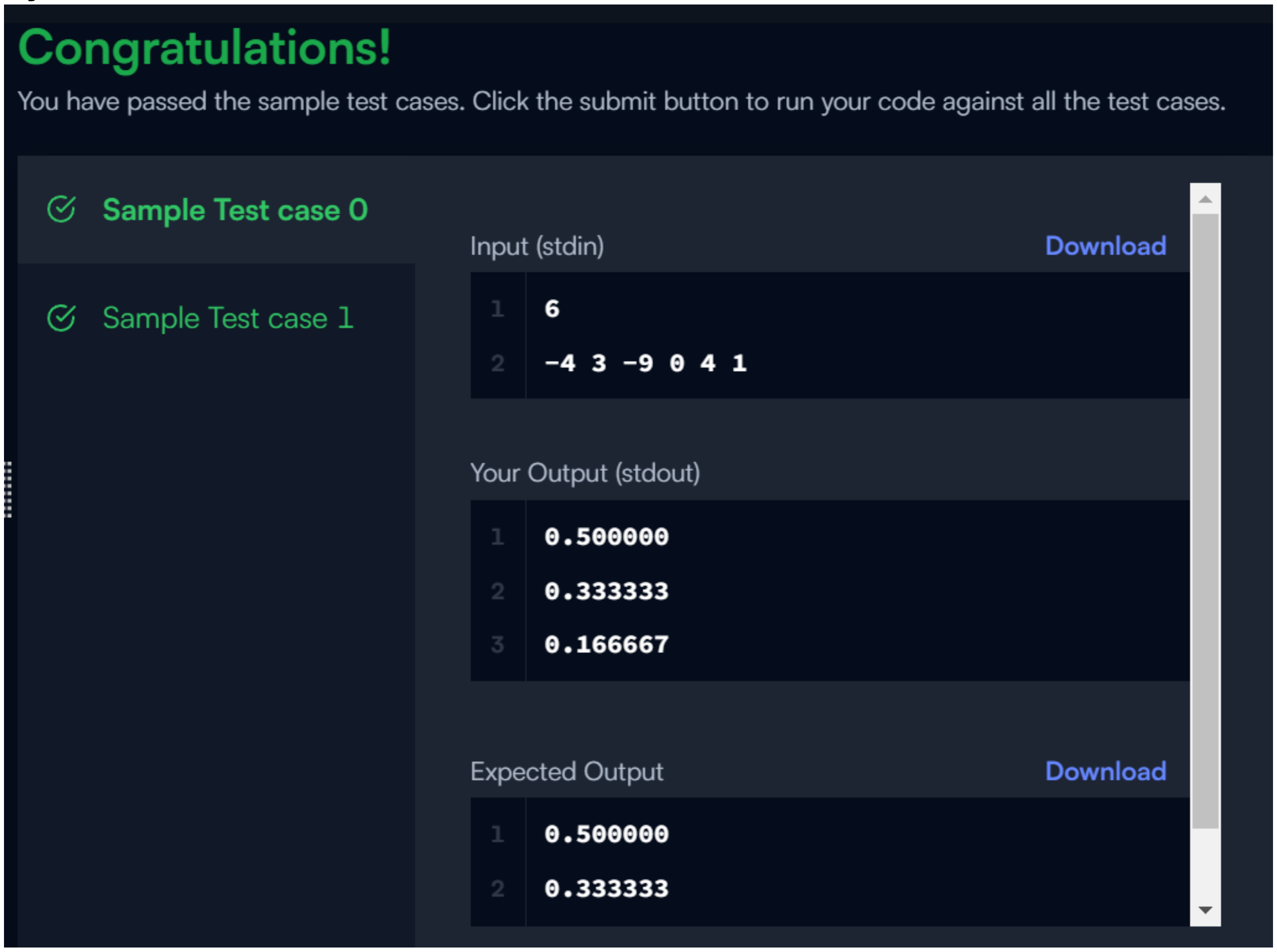


Miguel Yac 🇬🇹

@myac_2022020

Ejercicios Fáciles.



Congratulations!

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

Input (stdin)

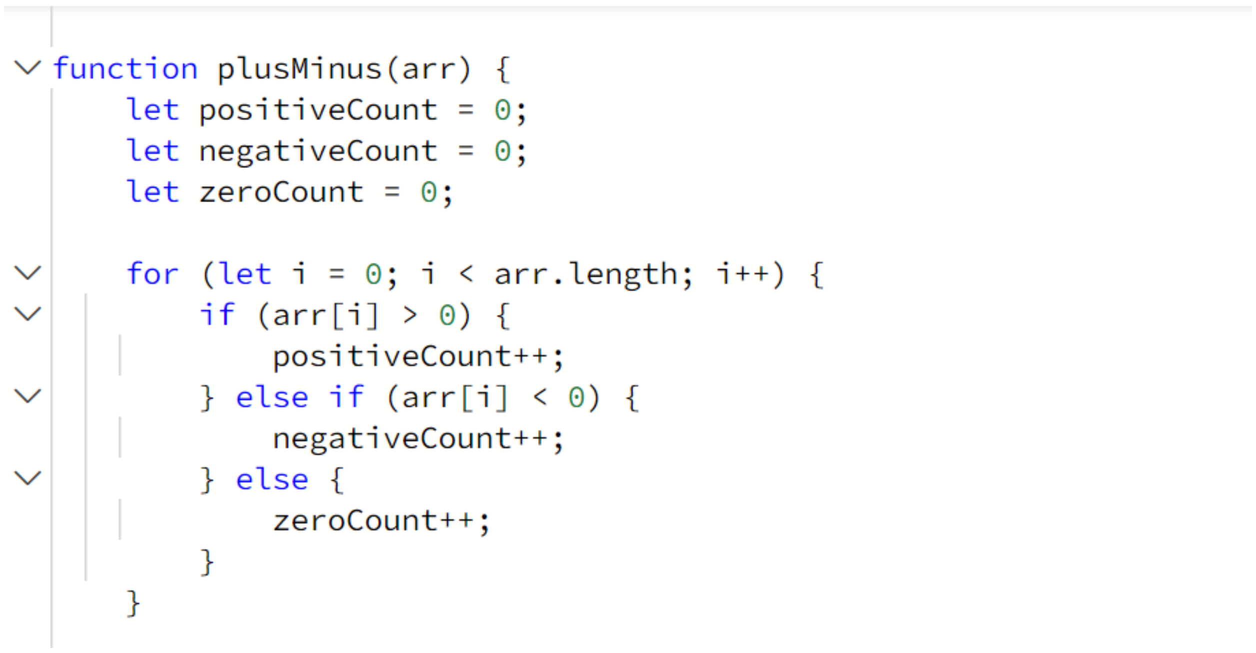
```
6
-4 3 -9 0 4 1
```

Your Output (stdout)

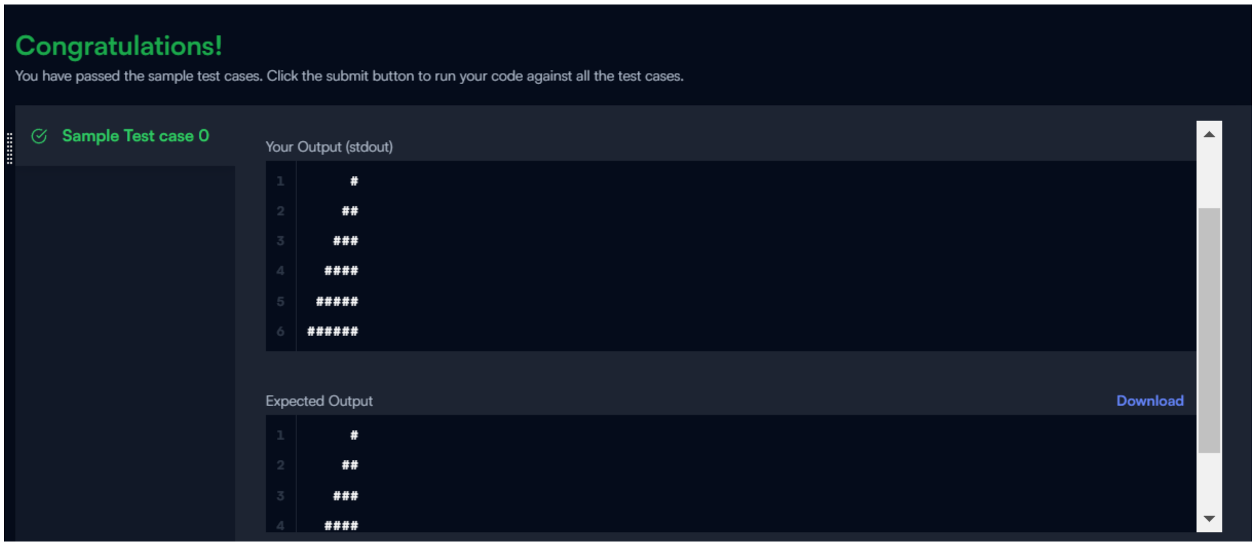
```
0.500000
0.333333
0.166667
```

Expected Output

```
0.500000
0.333333
```



```
function plusMinus(arr) {
    let positiveCount = 0;
    let negativeCount = 0;
    let zeroCount = 0;

    for (let i = 0; i < arr.length; i++) {
        if (arr[i] > 0) {
            positiveCount++;
        } else if (arr[i] < 0) {
            negativeCount++;
        } else {
            zeroCount++;
        }
    }
```



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

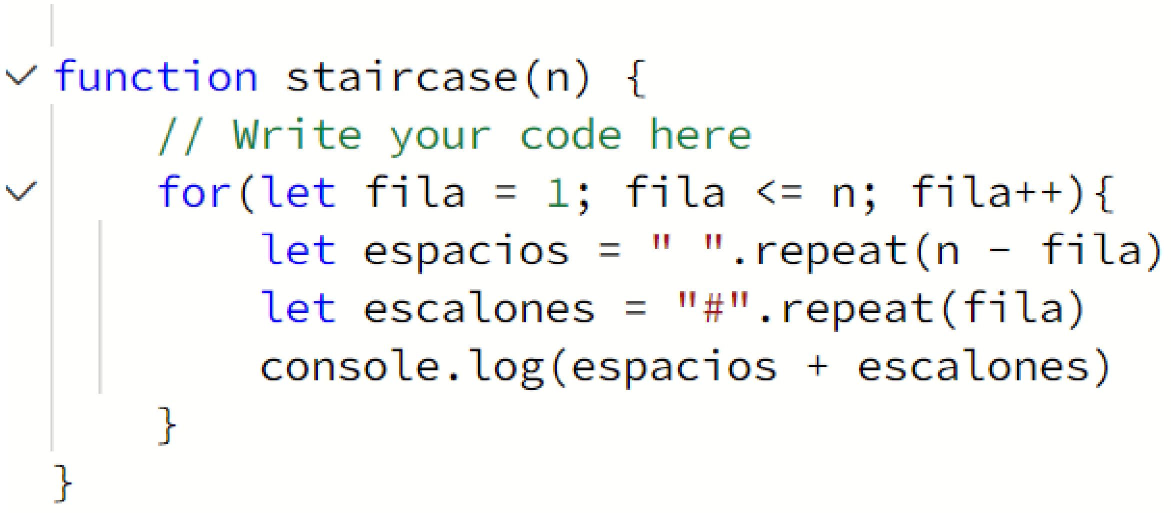
Your Output (stdout)

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

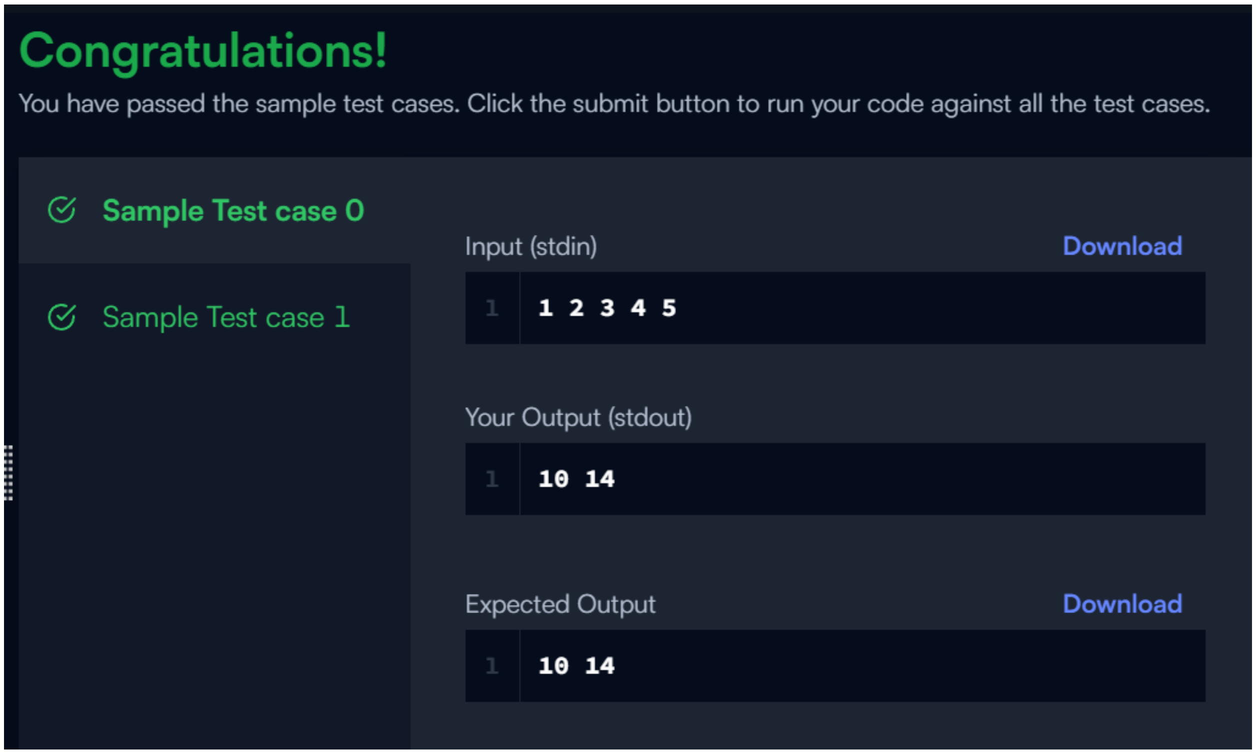
Expected Output

Download

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



```
function staircase(n) {
    // Write your code here
    for(let fila = 1; fila <= n; fila++){
        let espacios = " ".repeat(n - fila)
        let escalones = "#".repeat(fila)
        console.log(espacios + escalones)
    }
}
```



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
```

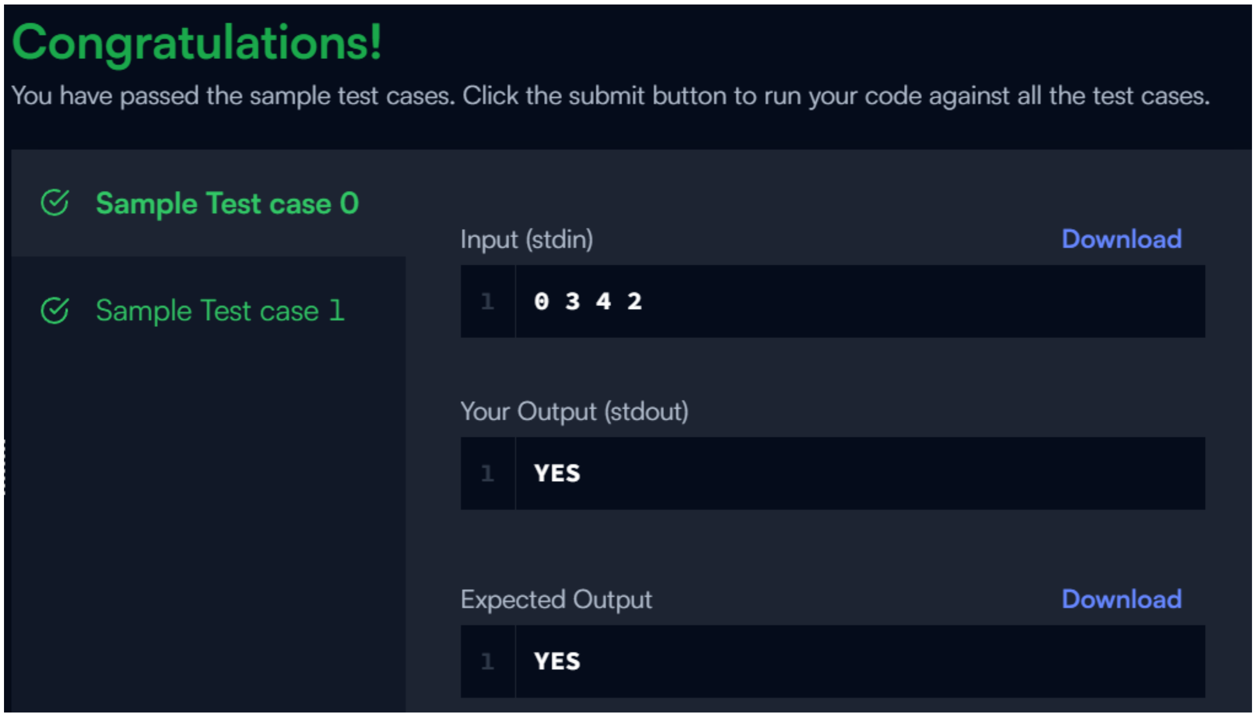
Your Output (stdout)

```
10 14
```

Expected Output

```
10 14
```

```
function miniMaxSum(arr) {
    // Write your code here
    arr.sort((a, b) => a - b)
    let minSum = arr.slice(0, 4).reduce((acc, val) => acc+val, 0)
    let maxSum = arr.slice(1).reduce((acc, val) => acc+val, 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.

Sample Test case 0

Sample Test case 1

Input (stdin)

Download

```
0 3 4 2
```

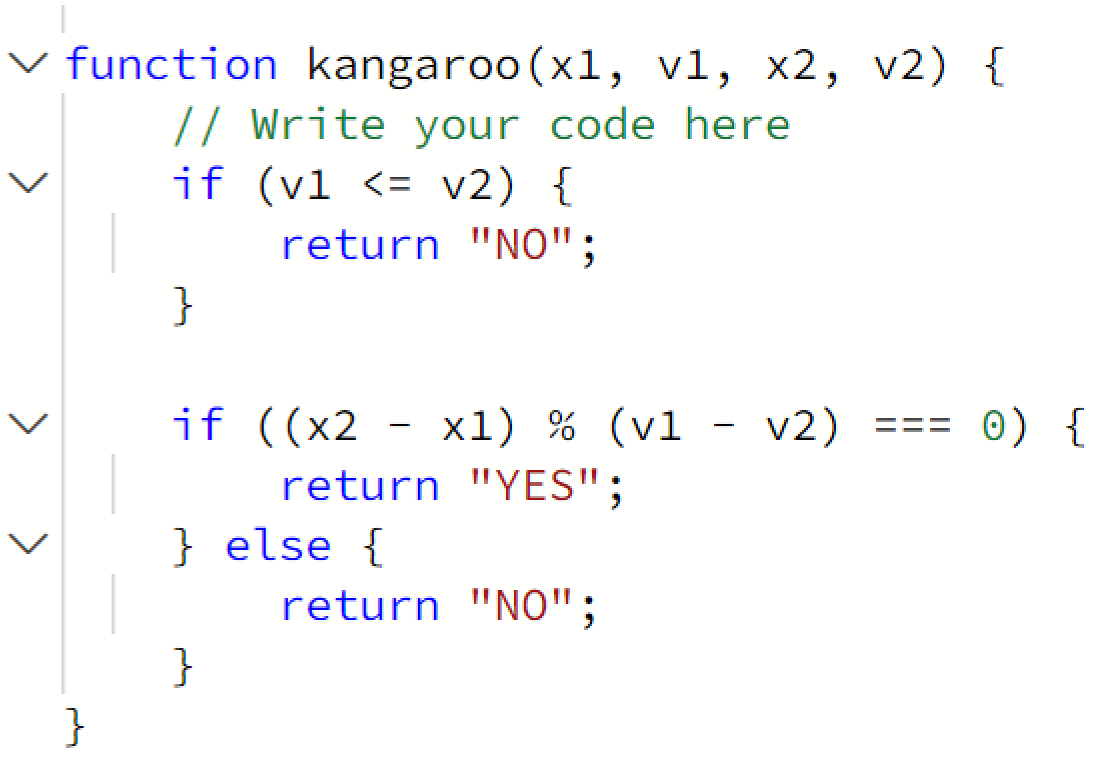
Your Output (stdout)

```
YES
```

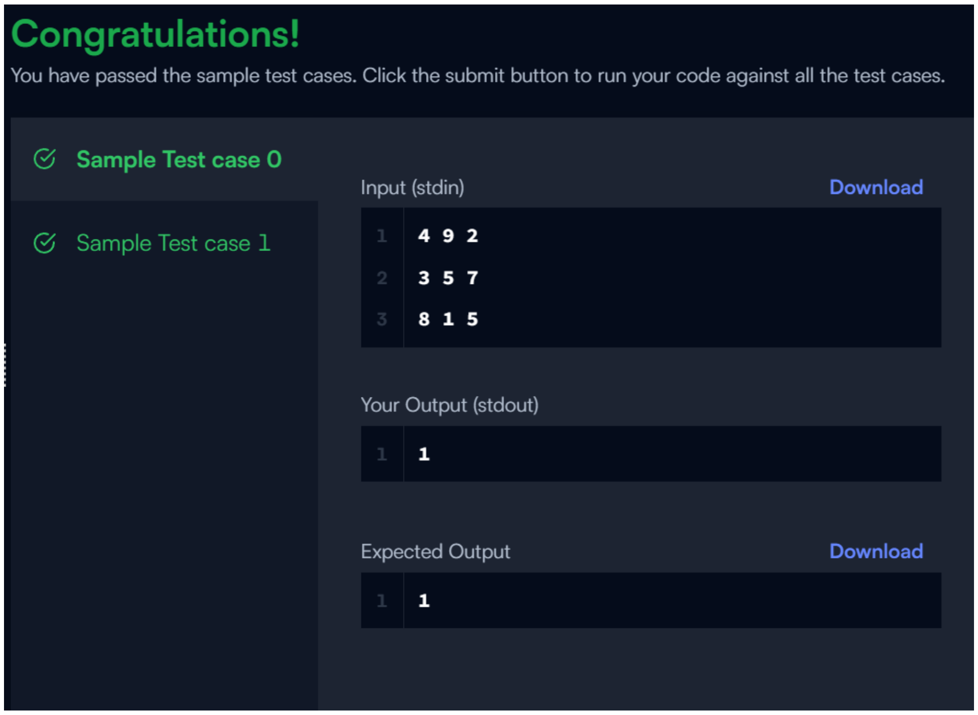
Expected Output

Download

```
YES
```


```
function kangaroo(x1, v1, x2, v2) {
    // Write your code here
    if (v1 <= v2) {
        return "NO";
    }

    if ((x2 - x1) % (v1 - v2) === 0) {
        return "YES";
    } else {
        return "NO";
    }
}
```

Intermedio.


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

Input (stdin)

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

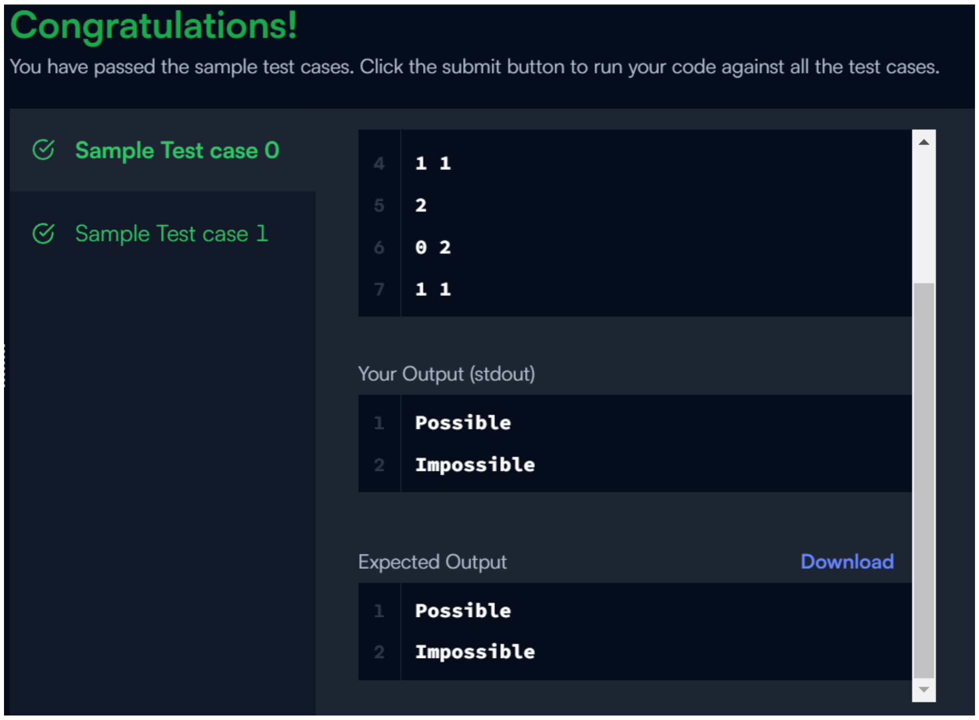
Your Output (stdout)

```
1
```

Expected Output

```
1
```

```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 = Infinity;

    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.

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

Your Output (stdout)

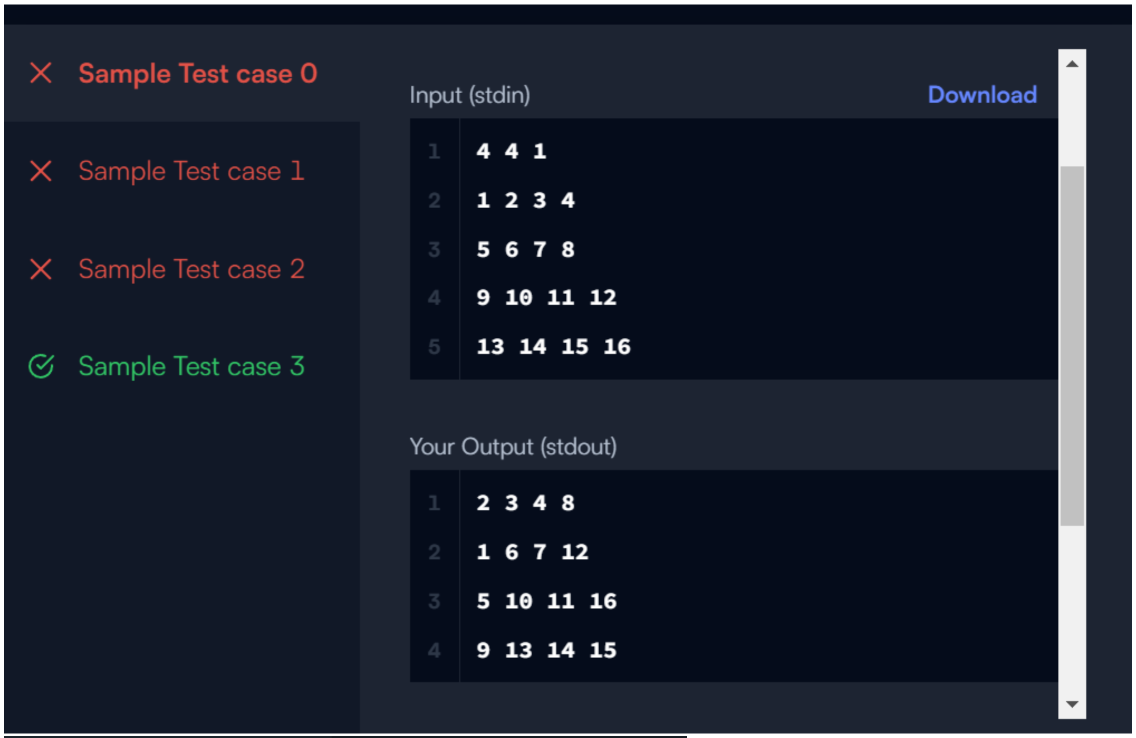
```
Possible
Impossible
```

Expected Output

```
Possible
Impossible
```

```javascript
function organizingContainers(container) {
    const n = container.length;
    const m = container[0].length;

    const ballsTotal = Array(n).fill(0);
    const capacityTotal = Array(m).fill(0);

    for (let i = 0; i < n; i++) {
        for (let j = 0; j < m; j++) {
            ballsTotal[i] += container[i][j];
            capacityTotal[j] += container[i][j];
        }
    }

    for (let i = 0; i < n; i++) {
        let found = false;
        for (let j = 0; j < m; j++) {
            if (ballsTotal[i] === capacityTotal[j]) {
                found = true;
                break;
            }
        }
        if (!found) {
            return "Impossible";
        }
    }

    return "Possible";
}
```

Difícil.



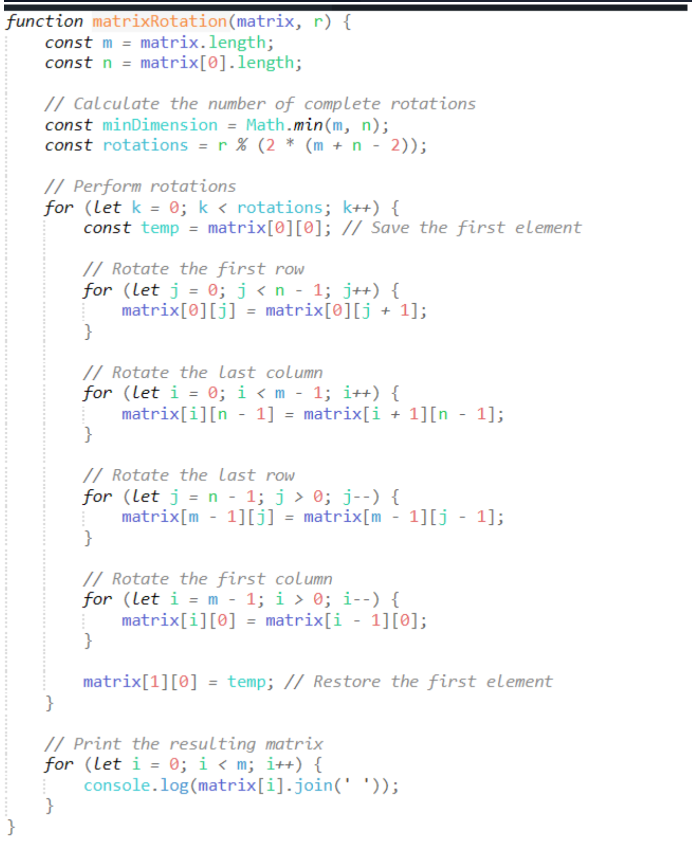
Sample Test case 0
Sample Test case 1
Sample Test case 2
Sample Test case 3

Input (stdin)

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

Your Output (stdout)

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



```javascript
function matrixRotation(matrix, r) {
    const m = matrix.length;
    const n = matrix[0].length;

    // Calculate the number of complete rotations
    const minDimension = Math.min(m, n);
    const rotations = r % (2 * (m + n - 2));

    // Perform rotations
    for (let k = 0; k < rotations; k++) {
        const temp = matrix[0][0]; // Save the first element

        // Rotate the first row
        for (let j = 0; j < n - 1; j++) {
            matrix[0][j] = matrix[0][j + 1];
        }

        // Rotate the last column
        for (let i = 0; i < m - 1; i++) {
            matrix[i][n - 1] = matrix[i + 1][n - 1];
        }

        // Rotate the last row
        for (let j = n - 1; j > 0; j--) {
            matrix[m - 1][j] = matrix[m - 1][j - 1];
        }

        // Rotate the first column
        for (let i = m - 1; i > 0; i--) {
            matrix[i][0] = matrix[i - 1][0];
        }

        matrix[1][0] = temp; // Restore the first element
    }

    // Print the resulting matrix
    for (let i = 0; i < m; i++) {
        console.log(matrix[i].join(' '));
    }
}
```