# Day 0: Hello, World!



'use strict';

process.stdin.resume();

process.stdin.setEncoding('utf-8');

let inputString = '';

let currentLine = 0;

process.stdin.on('data', inputStdin => {

    inputString += inputStdin;

});

process.stdin.on('end', \_ => {

    inputString = inputString.trim().split('\n').map(string => {

        return string.trim();

    });

    main();

});

function readLine() {

    return inputString[currentLine++];

}

/\*\*

\*   A line of code that prints "Hello, World!" on a new line is provided in the editor.

\*   Write a second line of code that prints the contents of 'parameterVariable' on a new line.

\*

\*   Parameter:

\*   parameterVariable - A string of text.

\*\*/

function greeting(parameterVariable) {

    // This line prints 'Hello, World!' to the console:

    console.log('Hello, World!');

    // Write a line of code that prints parameterVariable to stdout using console.log:

    console.log(parameterVariable);

}

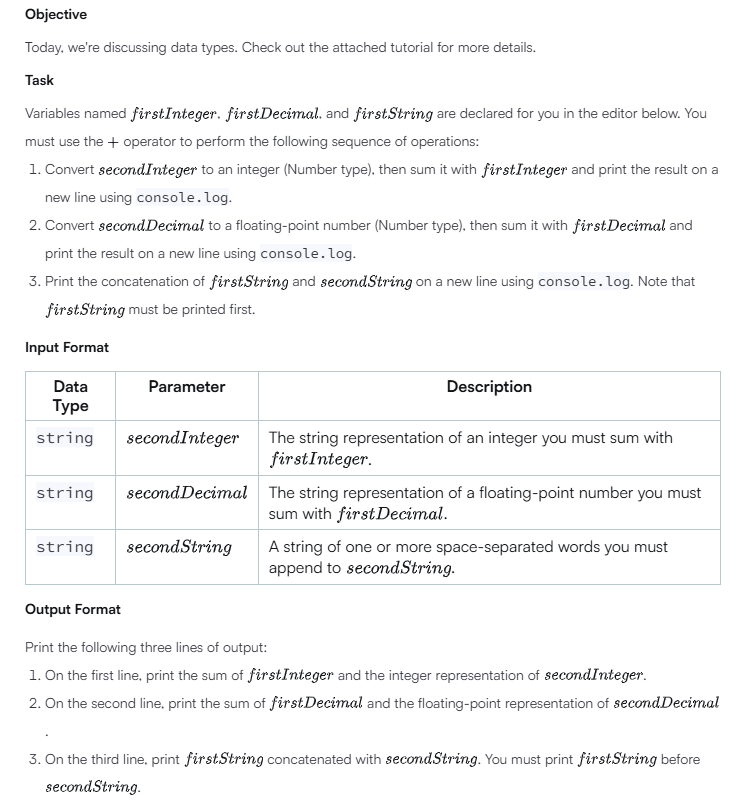
function main() {

    const parameterVariable = readLine();

    greeting(parameterVariable);

}

# Day 0: Data Types

'use strict';

process.stdin.resume();

process.stdin.setEncoding('utf-8');

let inputString = '';

let currentLine = 0;

process.stdin.on('data', inputStdin => {

    inputString += inputStdin;

});

process.stdin.on('end', \_ => {

    inputString = inputString.trim().split('\n').map(string => {

        return string.trim();

    });

    main();

});

function readLine() {

    return inputString[currentLine++];

}

/\*\*

\*   The variables 'firstInteger', 'firstDecimal', and 'firstString' are declared for you -- do not modify them.

\*   Print three lines:

\*   1. The sum of 'firstInteger' and the Number representation of 'secondInteger'.

\*   2. The sum of 'firstDecimal' and the Number representation of 'secondDecimal'.

\*   3. The concatenation of 'firstString' and 'secondString' ('firstString' must be first).

\*

\*   Parameter(s):

\*   secondInteger - The string representation of an integer.

\*   secondDecimal - The string representation of a floating-point number.

\*   secondString - A string consisting of one or more space-separated words.

\*\*/

function performOperation(secondInteger, secondDecimal, secondString) {

    // Declare a variable named 'firstInteger' and initialize with integer value 4.

    const firstInteger = 4;

    // Declare a variable named 'firstDecimal' and initialize with floating-point value 4.0.

    const firstDecimal = 4.0;

    // Declare a variable named 'firstString' and initialize with the string "HackerRank".

    const firstString = 'HackerRank ';

    // Print the sum of 'firstInteger' and 'secondInteger' (converted to a Number type)

    console.log(firstInteger + Number(secondInteger));

    // Parse the secondDecimal and check the number of decimal places

    const secondDecimalValue = parseFloat(secondDecimal);

    const decimalPlaces = secondDecimal.split(".")[1]?.length || 0;

    // Dynamically format the sum based on the number of decimal places

    if (decimalPlaces > 0) {

        console.log((firstDecimal + secondDecimalValue).toFixed(decimalPlaces));

    } else {

        console.log(firstDecimal + secondDecimalValue);

    }

    // Print the concatenation of 'firstString' and 'secondString' on a new line

    console.log(firstString + secondString);

}

function main() {

    const secondInteger = readLine();

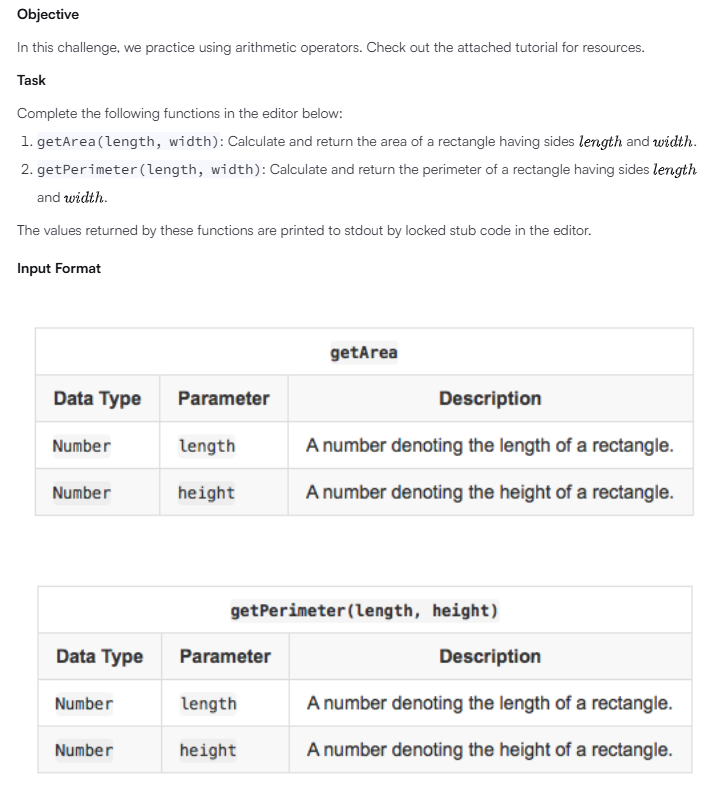
    const secondDecimal = readLine();

    const secondString = readLine();

    performOperation(secondInteger, secondDecimal, secondString);

}

# Day 1: Arithmetic Operators



'use strict';

process.stdin.resume();

process.stdin.setEncoding('utf-8');

let inputString = '';

let currentLine = 0;

process.stdin.on('data', inputStdin => {

    inputString += inputStdin;

});

process.stdin.on('end', \_ => {

    inputString = inputString.trim().split('\n').map(string => {

        return string.trim();

    });

    main();

});

function readLine() {

    return inputString[currentLine++];

}

/\*\*

\*   Calculate the area of a rectangle.

\*

\*   length: The length of the rectangle.

\*   width: The width of the rectangle.

\*

\*   Return a number denoting the rectangle's area.

\*\*/

function getArea(length, width) {

    let area;

    // Write your code here

    area = length \* width;

    return area;

}

/\*\*

\*   Calculate the perimeter of a rectangle.

\*

\*   length: The length of the rectangle.

\*   width: The width of the rectangle.

\*

\*   Return a number denoting the perimeter of a rectangle.

\*\*/

function getPerimeter(length, width) {

    let perimeter;

    // Write your code here

    perimeter = 2 \* (length + width);

    return perimeter;

}

function main() {

    const length = +(readLine());

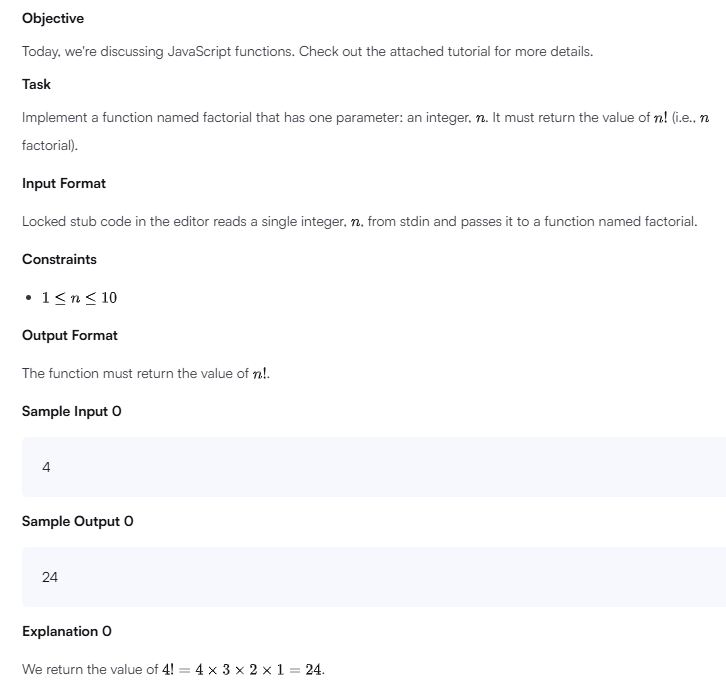
    const width = +(readLine());

    console.log(getArea(length, width));

    console.log(getPerimeter(length, width));

}

# Day 1: Functions



'use strict';

process.stdin.resume();

process.stdin.setEncoding('utf-8');

let inputString = '';

let currentLine = 0;

process.stdin.on('data', inputStdin => {

    inputString += inputStdin;

});

process.stdin.on('end', \_ => {

    inputString = inputString.trim().split('\n').map(string => {

        return string.trim();

    });

    main();

});

function readLine() {

    return inputString[currentLine++];

}

/\*

 \* Create the function factorial here

 \*/

function factorial(n) {

    if (n === 0 || n === 1) {

        return 1;

    }

    return n \* factorial(n - 1);  // Recursively calculate the factorial

}

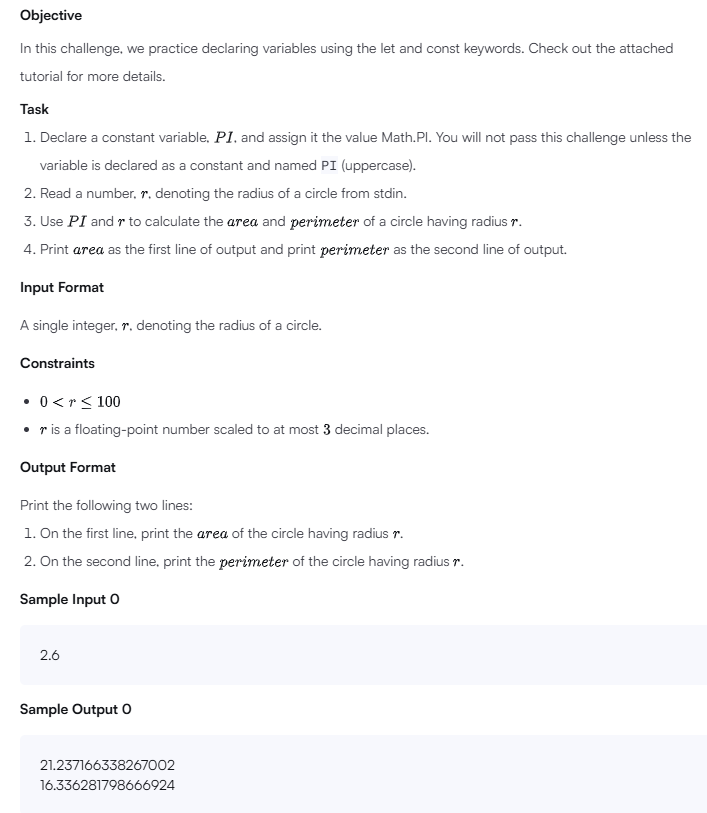
function main() {

    const n = +(readLine());

    console.log(factorial(n));

}

# Day 1: Let and Const



'use strict';

process.stdin.resume();

process.stdin.setEncoding('utf-8');

let inputString = '';

let currentLine = 0;

process.stdin.on('data', inputStdin => {

    inputString += inputStdin;

});

process.stdin.on('end', \_ => {

    inputString = inputString.trim().split('\n').map(string => {

        return string.trim();

    });

    main();

});

function readLine() {

    return inputString[currentLine++];

}

function main() {

    // Write your code here. Read input using 'readLine()' and print output using 'console.log()'.

    const PI = Math.PI;  // Declare PI as a constant with the value of Math.PI

    const r = parseFloat(readLine());  // Read the radius as a floating-point number

    // Calculate the area and perimeter

    const area = PI \* r \* r;

    const perimeter = 2 \* PI \* r;

    // Print the area and perimeter

    console.log(area);

    console.log(perimeter);

    // Print the area of the circle:

    // Print the perimeter of the circle:

    try {

        // Attempt to redefine the value of constant variable PI

        PI = 0;

        // Attempt to print the value of PI

        console.log(PI);

    } catch(error) {

        console.error("You correctly declared 'PI' as a constant.");

    }

}