Javascript Assignment

Level - Medium

1. Age Category Checker

Description:

Given an integer age, determine whether the person is Minor (< 18), Adult (18–60), or Senior (> 60).

Input Format: A single integer age.

Output Format: A string among Minor, Adult, or Senior.

Example:

Input: 45

Output: Adult

2. Introduction Formatter

Description:

Given a person's name, age, and favorite hobby, output a single formatted string using template literals: "My name is {name}, I am {age} years old, and I enjoy {hobby}.".

Input Format: Three lines containing name (string), age (integer), hobby (string).

Output Format: A single string with formatted output.

Example:

Input: Alice, 25, Reading

Output: My name is Alice, I am 25 years old, and I enjoy Reading.

3. Sign of Number

Description:

Given an integer n, print Positive if n > 0, Negative if n < 0, or Zero if n == 0.

Input Format: An integer n.

Output Format: A string.

Example:

Input: 0

Output: Zero

4. FizzBuzz Enhancement

Description:

Create a standard/normal function which accepts an array of Integers. For each integer in range, output:

- Fizz if divisible by 3,
- Buzz if divisible by 5,
- FizzBuzz if divisible by both,
- the integer itself otherwise.

Input Format: Array of integers: [1, 2, 34, 5, 6, 23].

Output Format: The sequence, each on a new line.

Example:

Input: [1, 2, 3, 4, 15, 25]

Output:

1

2

Fizz

4

FizzBuzz

Buzz

5. Marks Analyzer

Description:

Create a function expression which takes two parameters,N (Number of students) and an array containing marks of students. Given marks of N students, compute total marks, average, highest and lowest score.

Input Format: N, [76, 64, 67, 94, 87, …N]

Output Format:

Total marks

Average marks (float or integer)

Highest mark

Lowest mark

Example:

Input:

5

80 90 75 60 95

Output:

Total: 400

Average: 80

Highest: 95

Lowest: 60

6. Reverse Array

Description:

Given an array of integers, return a new array with elements in reverse order (Use of inbuilt array method **reverse()** is not allowed)

Input Format: Array of N integers.

Output Format: N space-separated integers in reverse.

Example:

Input: [1, 3, 2, 5]

Output: 5 2 3 1

7. Sum using Rest Parameters

Description:

Given an unspecified number of integers, compute their sum using a function that accepts rest parameters.

Input Format: A single line with integers separated by spaces.

Output Format: Sum of integers.

Example:

Input: 10 20 30

Output: 60

8. Value vs Reference Demo

Description:

Write a program to demonstrate the difference between passing an array by value (copy) vs reference (original). Modify the array inside a function.

Input: Any sample array.

Output: Show original array before and after function calls.

9. Array Transformation using HOFs

Description:

Given an array of integers, use map() to square each number, then filter() to keep those greater than 25.

Input Format:

First line: integer N

Second line: N integers.

Output Format: Space-separated integers after transformation.

Example:

Input:

5

2 4 6 1 5

Output: 36 25

10. Custom Calculation with Callback

Description:

Write a function calculate(arr, callback, operation) that processes a numeric array based on a callback (arr, operation).

Input Format: array, callback, operation(sum/product/average)

Output Format: Result of the calculation.

Example:

Input:

4

2 4 6 8

sum

Output: 20