Basic DSA-style questions that use only variables, data types, and conditional statements in JavaScript

Question: 1
Declare a variable `num` with any value. Use an `if-else` condition to check whether the number is positive, negative, or zero. Print the result (with the help of console.log).

Examples:

- Input: `num = 8`
  Output: `"The number is positive."`
- Input: `num = -5`
  Output: `"The number is negative."`
- Input: `num = 0`
  Output: `"The number is zero."`

Question: 2
Create two variables, `a` and `b`, with different numbers. Write a program that compares them and prints which one is greater.

Examples:

- Input: `a = 12, b = 7`
  Output: `"a is greater than b"`
- Input: `a = 4, b = 15`
  Output: `"b is greater than a"`

Question: 3
Store a temperature value in a variable. Use a conditional statement to print:

- "Hot" if the temperature is above 30,
- "Warm" if it's between 15 and 30,

- "Cold" if it's below 15.

Examples:

- Input: `temperature = 35`
  Output: `"Hot"`
- Input: `temperature = 20`
  Output: `"Warm"`
- Input: `temperature = 10`
  Output: `"Cold"`

For Loop Questions:

1. Print Numbers from 1 to 10

- Use a `for` loop to print the numbers from 1 to 10.

2.Sum of First 10 Natural Numbers

- Use a `for` loop to calculate and print the sum of the first 10 natural numbers (1 to 10).

Examples:

- Sum of numbers from 1 to 10:
  - Output: `"The sum is 55"`

3.**Print Even Numbers from 1 to 20**

- Use a `for` loop to print all even numbers between 1 and 20.

4.**Find the Factorial of a Number**

- Use a `for` loop to calculate the factorial of a given number `n`.

Examples:

- Input: `n = 4`
  Output: `"The factorial of 4 is 24"`
- Input: `n = 6`
  Output: `"The factorial of 6 is 720"`

Functions Questions:

1. Create a function to add two numbers.

Write a function called `addNumbers` that takes two numbers as arguments, adds them, and prints the sum using `console.log`.

Examples:

- Input: `addNumbers(5, 3)`
  Output: `"The sum is 8"`
- Input: `addNumbers(10, 15)`
  Output: `"The sum is 25"`

2. Write a function that multiplies a number by itself.

Define a function `squareNumber` that takes a number as an argument and prints its square using `console.log`.

Examples:

- Input: `squareNumber(4)`
  Output: `"The square of 4 is 16"`
- Input: `squareNumber(7)`
  Output: `"The square of 7 is 49"`

3. Create a function to check if a number is even or odd.

Write a function called `checkEvenOdd` that takes a number as input and prints whether it's even or odd.

Examples:

- Input: `checkEvenOdd(8)`
  Output: `"8 is even"`
- Input: `checkEvenOdd(15)`
  Output: `"15 is odd"`

4. Write a function that prints all numbers from 1 to N.

Define a function `printNumbers` that takes a number N as input and prints all numbers from 1 to N using a loop.

5. Multiply Two Numbers
Write a function called `multiplyNumbers` that takes two arguments, multiplies them, and prints the result using `console.log`.

Examples:

- Input: `multiplyNumbers(6, 4)`
  Output: `"The result is 24"`
- Input: `multiplyNumbers(3, 7)`
  Output: `"The result is 21"`

6. Find the Largest of Two Numbers

Create a function called `findLargest` that takes two numbers as arguments and prints the larger number using `console.log`.

Examples:

- Input: `findLargest(12, 5)`
  Output: `"The larger number is 12"`
- Input: `findLargest(8, 15)`
  Output: `"The larger number is 15"`

7. Function to Calculate Factorial

Create a function called `findFactorial` that takes a number as an argument and prints the factorial of that number.

Examples:

- Input: `findFactorial(4)`
  Output: `"The factorial of 4 is 24"`
- Input: `findFactorial(5)`
  Output: `"The factorial of 5 is 120"`

String Questions:

1.**Concatenate Strings:** Write a function that takes two strings as arguments and prints their concatenation.

Examples:

- Input: `concatenateStrings("Hello", "World")`
  Output: `"HelloWorld"`
- Input: `concatenateStrings("Good", "Morning")`
  Output: `"GoodMorning"`

2.**Count the Number of Characters:** Write a function that accepts a string and counts how many characters (excluding spaces) it contains.

Examples:

- Input: `countCharacters("Hello World")`
  Output: `"Number of characters: 10"`
- Input: `countCharacters("OpenAI rocks")`
  Output: `"Number of characters: 10"`

3.**Check if Two Strings Are Equal:** Write a function that compares two strings and checks if they are equal (case-sensitive comparison).

Examples:

- Input: `checkStringsEqual("Hello", "hello")`
  Output: `"Strings are not equal"`
- Input: `checkStringsEqual("Test", "Test")`
  Output: `"Strings are equal"`

4.**Print Each Character of a String:** Write a function that takes a string and prints each character of the string on a new line.

5. Reverse Words in a String

Write a function that takes a string and returns the words in reverse order.

Examples:

- **Input:** `"Hello World"`
  - **Output:** `"World Hello"`
- **Input:** `"OpenAI is great"`
  - **Output:** `"great is OpenAI"`

6. Longest Common Prefix

Write a function that takes an array of strings and returns the longest common prefix.

Examples:

- **Input:** `["flower", "flow", "flight"]`
  - **Output:** `"fl"`
- **Input:** `["dog", "racecar", "car"]`
  - **Output:** `""` **(no common prefix)**

7. Valid Anagram

Write a function that checks if two strings are anagrams of each other (i.e., contain the same characters in the same frequency, ignoring the order).

Examples:

- **Input:** `("listen", "silent")`
  - **Output:** `true`
- **Input:** `("hello", "world")`
  - **Output:** `false`

8. Valid Palindrome

Write a function that checks if a given string is a palindrome (i.e., reads the same backward as forward, ignoring spaces and punctuation).

Examples:

- **Input:** `"A man, a plan, a canal, Panama"`
  - **Output:** `true`
- **Input:** `"race a car"`
  - **Output:** `false`

Break and continue statement

1.Check if a String Contains a Specific Character: Write a function that checks if a string contains a specific character.

Examples:

- Input: `containsCharacter("Hello World", "o")`
  Output: `"Character found"`
- Input: `containsCharacter("JavaScript", "z")`
  Output: `"Character not found"`

2.**Find First Occurrence of a Character (Use `break`):** Write a function that searches for the first occurrence of a specific character in a string and stops the search as soon as the character is found.

Examples:

- Input: `findFirstOccurrence("Hello World", "o")`
  Output: `"First occurrence at index: 4"`

- Input: `findFirstOccurrence("OpenAI", "p")`
  Output: `"First occurrence at index: 1"`

3.**Skip Vowels in a String (Use `continue`):** Write a function that prints all characters of a string except the vowels. Use `continue` to skip the vowels.

Examples:

- Input: `skipVowels("Hello World")`
  Output: `"Hll Wrld"`
- Input: `skipVowels("JavaScript")`
  Output: `"JvScrpt"`

Array Questions:

1.**Sum of All Elements:** Write a function that takes an array of numbers and calculates the sum of all its elements.

Examples:

- Input: `sumOfElements([1, 2, 3, 4, 5])`
  Output: `15`
- Input: `sumOfElements([10, 20, 30])`
  Output: `60`

2.Find Maximum Element: Write a function that finds the maximum number in an array and logs it to the console.

Examples:

- Input: `findMaximum([1, 5, 3, 9, 2])`
  Output: `9`
- Input: `findMaximum([-10, -5, -1, -20])`
  Output: `-1`

3.**Count Odd and Even Numbers:** Write a function that counts how many odd and even numbers are in an array and prints the counts.

Examples:

- Input: `countOddEven([1, 2, 3, 4, 5, 6])`
  Output: `"Odd: 3, Even: 3"`
- Input: `countOddEven([10, 15, 20, 25])`
  Output: `"Odd: 2, Even: 2"`

4.**Reverse the Array:** Write a function that reverses the elements of an array and prints the reversed array.

Examples:

- Input: `reverseArray([1, 2, 3, 4, 5])`
  Output: `[5, 4, 3, 2, 1]`
- Input: `reverseArray(['a', 'b', 'c'])`
  Output: `['c', 'b', 'a']`

5.**Check for Duplicates:** Write a function that checks if an array contains any duplicate elements. Print `true` if duplicates are found, otherwise print `false`.

Examples:

- Input: `checkDuplicates([1, 2, 3, 4, 1])`
  Output: `true`
- Input: `checkDuplicates([5, 6, 7, 8])`
  Output: `false`

6.**Find Element Index:** Write a function that takes an array and a number, then finds and logs the index of that number in the array. If the number is not found, print `-1`.

Examples:

- Input: `findIndex([10, 20, 30, 40], 30)`
  Output: `2`
- Input: `findIndex([1, 2, 3], 5)`
  Output: `-1`

7.**Merge Two Arrays:** Write a function that merges two arrays into one and logs the new array.

Examples:

- Input: `mergeArrays([1, 2], [3, 4])`
  Output: `[1, 2, 3, 4]`
- Input: `mergeArrays(['a', 'b'], ['c', 'd'])`
  Output: `['a', 'b', 'c', 'd']`

8.**Find Minimum Element:** Write a function that finds the minimum number in an array and logs it to the console.

Examples:

- Input: `findMinimum([10, 5, 20, 1])`
  Output: `1`

- Input: `findMinimum([-3, -1, -7, -2])`
  Output: `-7`

9.**Count the Number of Elements:** Write a function that counts the total number of elements in an array and prints the count.

Examples:

- Input: `countElements([1, 2, 3, 4, 5])`
  Output: `5`
- Input: `countElements(['apple', 'banana', 'cherry'])`
  Output: `3`

10.**Find the Average of Elements:** Write a function that calculates and prints the average of the numbers in an array.

Examples:

- Input: `findAverage([10, 20, 30, 40])`
  Output: `25`
- Input: `findAverage([5, 10, 15])`
  Output: `10`

Mock Interview 1

Programming Questions

Count Number of Even and Odd Elements in an Array

Write a function that takes an array of numbers and counts how many even and odd numbers it contains.

Examples:
Input: countEvenOdd([1, 3, 5, 8, 10]) Output: "Odd: 3, Even: 2"

Input: countEvenOdd([2, 4, 6, 7]) Output: "Odd: 1, Even: 3"

Program for Average of an Array

Write a function that calculates the average of the elements in an array.

Examples:
Input: averageOfArray([2, 4, 6, 8]) Output: 5

Input: averageOfArray([10, 20, 30, 40]) Output: 25

Program to Print the Given Digit in Words

Write a function that prints the given digit in words, but only if the digit is between 0 and 9. If the input is greater than 9, return an appropriate message indicating that only single digits are allowed.

Examples:
Input: digitToWord(5)

Output: "Five"

Input: digitToWord(3) Output: "Three"

Check if a Number is Divisible by 6 or Not

Write a function that checks if a given large number (as a string) is divisible by 6.

Examples:
Input: isDivisibleBySix('123456') Output: true

Input: isDivisibleBySix('12345') Output: false

Check if a Number is a Palindrome

Write a function that checks if a given number is a palindrome.

Examples:
Input: isPalindrome(121) Output: true

Input: isPalindrome(123) Output: false

Program to Count Vowels in a String

Write a function that counts the number of vowels in a given string.

Examples:
Input: countVowels("Hello World") Output: 3

Input: countVowels("Javascript") Output: 3

Check if Given Number is a Perfect Square

Write a function to check if the given number is a perfect square.

Examples:
Input: isPerfectSquare(16) Output: true

Input: isPerfectSquare(20) Output: false

Program to Find the Maximum Element in a Matrix

Write a function that finds the maximum element in a given 2D matrix.

Examples:
Input: findMaxInMatrix([[1, 2, 3], [4, 5, 6], [7, 8, 9]]) Output: 9

Input: findMaxInMatrix([[5, 1], [8, 3], [2, 7]]) Output: 8

Program for Sum of the Digits of a Given Number

Write a function that finds the sum of the digits of a given number.

Examples:
Input: sumOfDigits(123) Output: 6

Input: sumOfDigits(456) Output: 15

Difference Between Sum of the Squares of First n Natural Numbers and Square of Su

Write a function that finds the difference between the sum of the squares of the first n natural numbers and the square of the sum.

Examples:
Input: difference(3) Output: 22

Input: difference(5) Output: 170

Program for Armstrong Numbers

Write a function that checks if a given number is an Armstrong number.

Examples:
Input: isArmstrong(153)

Output: true

Input: isArmstrong(123)

Output: false

Find the Factorial of a Number Iteratively and Recursively

Write two functions (iterative and recursive) to find the factorial of a given number n.

Examples:
Input: factorial(5) Output: 120

Input: factorial(0) Output: 1

Program to Check if an Array is Sorted or Not

Write a function that checks if an array is sorted in ascending order.

Examples:
Input: isSorted([1, 2, 3, 4]) Output: true

Input: isSorted([1, 3, 2, 4]) Output: false

Program to Print Multiplication Table of a Number

Write a function that prints the multiplication table of a given number up to 10.

Examples:
Input: multiplicationTable(5) Output:

5 x 1 =5
5 x 2 = 10 ...
5 x 10 = 50

Check if a Word is Present in a Sentence

Write a function that checks if a given word is present in a sentence.

Examples:
Input: isWordPresent("Hello world", "world") Output: true

Input: isWordPresent("Hello world", "hi") Output: false

Segregate 0s and 1s in an Array

Write a function that segregates 0s and 1s in an array.

Examples:
Input: segregateZerosAndOnes([0, 1, 0, 1, 1, 0]) Output: [0, 0, 0, 1, 1, 1]

Input: segregateZerosAndOnes([1, 1, 0, 0]) Output: [0, 0, 1, 1]

Write a Program to Reverse Digits of a Number

Write a function that reverses the digits of a given number.

Examples:
Input: reverseDigits(1234) Output: 4321

Input: reverseDigits(987) Output: 789

Find Second Largest Element in an Array

Write a function that finds the second largest element in an array.

Examples:
Input: secondLargest([1, 2, 3, 4, 5]) Output: 4

Input: secondLargest([10, 20, 10, 5]) Output: 10

Stack Questions:

Task 1: Create an Empty Stack and Push an Item

- Create an empty stack using an array, push the number 5 onto the stack, and log the updated stack to the console.

Task 2: Pop an Item from the Stack

- Using the stack created in Task 1, pop the top item from the stack and log the popped item as well as the updated stack.

Task 3: Check the Top Item of the Stack

- After Task 2, check what the top item of the stack is without removing it. Log the top item to the console.

Task 4: Check if the Stack is Empty

- Write a condition to check if the stack is empty and log a message indicating whether it is empty or not.

Task 5: Push Multiple Items onto the Stack

- Create an array with numbers `[10, 20, 30]`, then push each of these numbers onto the stack one by one, logging the stack after each push.

Task 6: Pop All Items from the Stack

- Using a loop, pop all items from the stack one by one until it is empty, logging each popped item.

Task 7: Implement a Function to Check if a Stack is Full

- Write a function that checks if the stack has reached its maximum capacity (for this task, consider a maximum capacity of 5 items). Log a message indicating if the stack is full or not.

Queue Questions:

Task: Implement a Circular Queue

Description:
Create a circular queue using an array with a fixed size. Your task is to implement the following operations:

1. **Enqueue:** Add an element to the rear of the queue.
2. **Dequeue:** Remove an element from the front of the queue.
3. **isEmpty:** Check if the queue is empty.
4. **isFull:** Check if the queue is full.

Make sure to handle the circular nature of the queue when adding or removing elements. Use `console.log` to display messages for each operation.

Taks for DOM

Task 5: Toggle Button Text and Style

Task 1: Change Text and Style of an Element

Task 3: Toggle Visibility of an Element

Task 4: Create a Dynamic Counter

Task 6: Show/Hide Password

Task 2: Add and Remove Elements

Task 7: Expand/Collapse a Section