JavaScript Test
GROUP – A

Multiple-Choice Questions (MCQs) - (2 Marks Each) Total 20 Marks

1. What will be the output of the following code?

```javascript
console.log(1 + '1');
```

A) 2
B) "11"
C) NaN
D) Error
Answer: B) `"11"`
Explanation: JavaScript converts the number to a string and concatenates it with "1".

2. What is the default value of a variable declared with `let` but not initialized?
A) `null`
B) `undefined`
C) `0`
D) `NaN`
Answer: B) `undefine`
Explanation: let variables that are declared but not initialized are undefined.

3. Which of the following operators checks both value and type?
A) `==`
B) `===`
C) `=`
D) `!`
Answer: B) `===`
Explanation: The `===` operator checks both the value and type, ensuring they match exactly.

4. What will be the output of this code?

```javascript
let x = 10;
let y = 20;
console.log(x == "10" && y == 20);
```

A) true
B) false
C) Error
D) undefined

Answer: A) `true`

Explanation: `x == "10"` is true because the `==` operator performs type coercion, and `y == 20` is true.

5. What is the result of the following expression?

```javascript
let x = 5;
let y = 2;
console.log(x % y);
```

A) 2
B) 1
C) 0
D) NaN
Answer: B) `1`
Explanation: The modulus operator (`%`) returns the remainder of division (`5 % 2` is `1`).

6. What is the output of the following code?

```javascript
const foo = () => { return 5 };
console.log(foo());
```

A) 5
B) undefined
C) Error
D) Na
Answer: A) `5`
Explanation: The arrow function `foo` returns `5`.

7. Which loop will always run at least once?
A) `for` loop
B) `while` loop
C) `do...while` loop
D) `forEach` loop

Answer: C) do...while loop

Explanation: The `do...while` loop guarantees at least one iteration before the condition is checked.

8. Which method removes the last element of an array?
A) `pop()`
B) `shift()`
C) `unshift()`
D) `slice()`

Answer: A) `pop()`
Explanation: The `pop()` method removes the last element from an array.

9. What does `JSON.parse()` do?
A) Converts a JavaScript object into a JSON string
B) Converts a JSON string into a JavaScript object
C) Initializes a JavaScript object
D) Converts an object to a strin
Answer: B) Converts a JSON string into a JavaScript object
Explanation: `JSON.parse()` converts a JSON-formatted string into a JavaScript object.

10. What is the output of the following code?

```javascript
let obj = {a: 1, b: 2};
let arr = Object.values(obj);
console.log(arr);
```

A) `[1, 2]`
B) `{a: 1, b: 2}`
C) `["a", "b"]`
D) undefined

Answer: A) `[1, 2]`

Explanation: `Object.values()` returns an array of the object's values.

End of Group - A

JavaScript Test

Group - B

Easy Coding Questions- (Choose any Two) **(15 Marks Each)** **Total 30 Marks**

1. Add Two Numbers (This problem helps you understand basic arithmetic and how to handle input and output in JavaScript.)

Problem Description

You are given two integers. Your task is to calculate and return their sum.

For example:

- If the input numbers are 5 and 7, their sum is 12.

Example

Input:

```
5 7
```

Output:

```
12
```

Function Description

Complete the function `addNumbers` as described below:

Function Signature:

```javascript
function addNumbers(a, b) {
    // Your code here
}
```

- **Parameters**:
  - `a`: An integer representing the first number.
  - `b`: An integer representing the second number.
- **Returns**:
  An integer, the sum of `a` and `b`.

Input Format

Two space-separated integers, `a` and `b`.

Output Format

A single integer, the sum of the two input numbers.

Constraints

- $-10^6 \leq a, b \leq 10^6$

---

2. Check If a String is Empty (This problem introduces you to string manipulation and basic conditional statements.)

Problem Description

You are given a string, and your task is to check if the string is empty. An empty string is a string with no characters (length 0). Return `true` if the string is empty, otherwise return `false`.

For example:

- Input: `""`
  Output: `true`
- Input: `"hello"`
  Output: `false`

Example

Input:

```
""
```

Output:

```
true
```

Function Description

Complete the function `isEmptyString` as described below:

Function Signature:

```javascript
function isEmptyString(str) {
    // Your code here
}
```

- **Parameters**:
  - `str`: A string that needs to be checked for emptiness.

- **Returns**:
  A boolean value: `true` if the string is empty, otherwise `false`.

Input Format

A single string `str`.

Output Format

A single boolean value (`true` or `false`).

Constraints

- $0 \leq \text{length of } str \leq 1000$

---

3. Print Even Numbers in Range (This problem introduces the concept of loops and conditional checks.)

Problem Description

Given an integer `n`, print all even numbers from `1` to `n`.

For example:

- If `n = 10`, the even numbers are `2, 4, 6, 8, 10`.

Example

Input:

```
10
```

Output:

```
2 4 6 8 10
```

Function Description

Complete the function `printEvenNumbers` as described below:

Function Signature:

```javascript
function printEvenNumbers(n) {
    // Your code here
}
```

- **Parameters**:
  - `n`: An integer representing the upper limit of the range.
- **Returns**:
  A string containing space-separated even numbers from `1` to `n`.

Input Format

A single integer `n`.

Output Format

A single line containing space-separated even numbers.

Constraints

- $1 \leq n \leq 1000$

---

4. Square of a Number (This problem focuses on using arithmetic operations in JavaScript.)

Problem Description

You are given an integer. Your task is to calculate and return its square.

For example:

- Input: `4`
  Output: `16`

This problem focuses on using arithmetic operations in JavaScript.

Example

Input:

```
4
```

Output:

```
16
```

Function Description

Complete the function `squareNumber` as described below:

Function Signature:

```javascript
function squareNumber(num) {
    // Your code here
}
```

- **Parameters**:
  - `num`: An integer representing the number to be squared.
- **Returns**:
  An integer, the square of the given number.

Input Format

A single integer `num`.

Output Format

A single integer, the square of the input number.

Constraints

- $-10^4 \leq num \leq 10^4$

---

5. Convert Celsius to Fahrenheit (This problem helps you practice arithmetic and working with floating-point numbers.)

Problem Description

You are given a temperature in Celsius. Your task is to convert it to Fahrenheit using the formula:

$$F = C \times \frac{9}{5} + 32$$

For example:

- Input: `25`
  Output: `77`

Example

Input:

```
25
```

Output:

```
77
```

Function Description

Complete the function `celsiusToFahrenheit` as described below:

Function Signature:

```
function celsiusToFahrenheit(celsius) {
    // Your code here
}
```

- **Parameters**:
  - `celsius`: A floating-point number representing the temperature in Celsius.
- **Returns**:
  A floating-point number, the equivalent temperature in Fahrenheit.

Input Format

A single floating-point number `celsius`.

Output Format

A single floating-point number, the temperature in Fahrenheit.

Constraints

- $-10^3 \leq celsius \leq 10^3$

---

6. Check Leap Year

Problem Description

You are given a year. Your task is to determine whether it is a leap year.

A year is a leap year if:

1. It is divisible by `4`.
2. However, if the year is divisible by `100`, it is not a leap year. Unless it is also divisible by `400`.

For example:

- Input: `2024`
  Output: `true`
- Input: `1900`
  Output: `false`

Example

Input:

```
2024
```

Output:

```
true
```

Function Description

Complete the function `isLeapYear` as described below:

Function Signature:

```javascript
function isLeapYear(year) {
    // Your code here
}
```

- **Parameters**:
    - `year`: An integer representing the year.
- **Returns**:
  A boolean value: `true` if the year is a leap year, otherwise `false`.

Input Format

A single integer `year`.

Output Format

A single boolean value (`true` or `false`).

Constraints

- $1 \leq year \leq 10^6$

End of Group - B

JavaScript Test

Group - C

Medium Coding Questions (Choose any two) **(25 Marks Each)** **Total 50 Marks**

---

1. Reverse a String (This problem helps you understand string manipulation and how to use loops effectively in JavaScript.)

Problem Description

You are given a string. Your task is to reverse the string and return it.

For example:

- Input: `"hello"`
  Output: `"olleh"`

This problem helps you understand string manipulation and how to use loops effectively in JavaScript.

Example

Input:

```
hello
```

Output:

```
olleh
```

Function Description

Complete the function `reverseString` as described below:

Function Signature:

```javascript
function reverseString(str) {
    // Your code here
}
```

- **Parameters**:
  - `str`: A string that needs to be reversed.
- **Returns**:
  A string, which is the reverse of the input.

Input Format

A single string `str`.

Output Format

A single string, the reversed input string.

Constraints

- $1 \leq \text{length of } str \leq 1000$

---

2. Find Prime Numbers in Range (This problem helps you practice working with loops and functions to determine prime numbers.)

Problem Description

You are given an integer `n`. Your task is to find all prime numbers up to `n` (inclusive).

For example:

- Input: `10`
  Output: `2, 3, 5, 7`

This problem helps you practice working with loops and functions to determine prime numbers.

Example

Input:

```
10
```

Output:

```
2 3 5 7
```

Function Description

Complete the function `findPrimes` as described below:

Function Signature:

```javascript
function findPrimes(n) {
    // Your code here
}
```

- **Parameters**:
  - `n`: An integer representing the upper limit of the range.
- **Returns**:
  A string of space-separated prime numbers from `1` to `n`.

Input Format

A single integer `n`.

Output Format

A single line containing space-separated prime numbers.

Constraints

- $2 \leq n \leq 1000$

---

3. Capitalize Each Word in a String (This problem helps you understand string manipulation and the use of functions in JavaScript.)

Problem Description

You are given a string. Your task is to capitalize the first letter of each word in the string.

For example:

- Input: `"hello world"`
  Output: `"Hello World"`

Example

Input:

```
hello world
```

Output:

```
Hello World
```

Function Description

Complete the function `capitalizeWords` as described below:

Function Signature:

```javascript
function capitalizeWords(sentence) {
    // Your code here
}
```

- **Parameters**:
  - sentence: A string containing multiple words.
- **Returns**:
  A string with the first letter of each word capitalized.

Input Format

A single string `sentence`.

Output Format

A single string with each word capitalized.

Constraints

- $1 \leq \text{length of } sentence \leq 1000$

---

4. Sum of Digits in a Number (This problem helps you understand how to extract digits from a number using loops.)

Problem Description

You are given an integer. Your task is to calculate the sum of its digits.

For example:

- Input: `123`
  Output: `6`

This problem helps you understand how to extract digits from a number using loops.

Example

Input:

```
123
```

Output:

```
6
```

Function Description

Complete the function `sumDigits` as described below:

Function Signature:

```javascript
function sumDigits(num) {
    // Your code here
}
```

- **Parameters**:
  - `num`: An integer whose digits need to be summed.
- **Returns**:
  An integer, the sum of the digits of the given number.

Input Format

A single integer num.

Output Format

A single integer, the sum of its digits.

Constraints

- $0 \leq num \leq 10^6$

---

5. Sort an Array (This problem helps you understand arrays and basic sorting algorithms or built-in methods in JavaScript.)

Problem Description

You are given an array of integers. Your task is to sort the array in ascending order.

For example:

- Input: `[4, 2, 9, 1]`
  Output: `[1, 2, 4, 9]`

Example

Input:

```
[4, 2, 9, 1]
```

Output:

```
[1, 2, 4, 9]
```

Function Description

Complete the function `sortArray` as described below:

Function Signature:

```javascript
function sortArray(arr) {
    // Your code here
}
```

- **Parameters**:
  - `arr`: An array of integers to be sorted.
- **Returns**:
  An array of integers sorted in ascending order.

Input Format

A single line containing space-separated integers representing the array.

Output Format

A single line containing the sorted array.

Constraints

- $1 \leq \text{length of } arr \leq 1000$
- $-10^3 \leq \text{each element of } arr \leq 10^3$

---

6. Count Occurrences of a Character (This problem helps you understand how to work with strings and loops in JavaScript.)

Problem Description

You are given a string and a character. Your task is to count how many times the character appears in the string.

For example:

- Input: `"hello", "l"`
  Output: `2`

Example

Input:

```
hello l
```

Output:

```
2
```

Function Description

Complete the function `countCharacter` as described below:

Function Signature:

```javascript
function countCharacter(str, char) {
    // Your code here
}
```

- **Parameters**:
  - `str`: A string to search within.
  - `char`: A character whose occurrences need to be counted.
- **Returns**:
  An integer, the count of the character in the string.

Input Format

Two space-separated strings:

1. `str`: The main string.

2. `char`: The character to count.

Output Format

A single integer, the count of the character in the string.

Constraints

- $1 \leq \text{length of } str \leq 1000$
- The character char will always be a single character.

End of Group - C