**ĐỀ TEST 13**

**Môn: WEB2062**

**Thời gian: 60 phút**

**Câu 1: DATA STRUCTURES (1 điểm)**

Cho một mảng số nguyên:

const numbers = [1, 2, 3, 4, 5];

Hãy viết mã JavaScript để in ra một mảng mới chứa các phần tử trong mảng `numbers` giai thừa.

**Câu 2: MODERN OPERATORS (1 điểm)**

Cho đoạn mã JavaScript sau:

const a = 5;

const b = 10;

const result = a \*\* b;

Giá trị của biến `**result**` là gì? Giải thích về toán tử **`\*\*`** trong JavaScript.

**Câu 3: FUNCTIONS (1 điểm)**

Viết một hàm JavaScript có tên là `**calculateAverage**` nhận vào một mảng số nguyên và trả về giá trị trung bình của các phần tử trong mảng.

**Câu 4: ARRAYS VÀ LÀM VIỆC VỚI ARRAYS (1 điểm)**

Cho mảng `**fruits**`:

const fruits = ['apple', 'banana', 'orange', 'apple', 'mango', 'banana', 'apple'];

Hãy viết mã JavaScript để đếm số lần xuất hiện của từng loại trái cây trong mảng `**fruits**` và lưu kết quả vào một đối tượng.

**Câu 5: ADVANCED DOM and EVENTS (1 điểm)**

Giả sử bạn có một đoạn mã HTML sau:

<button id="myButton">Click me</button>

<div id=”thongbao”><div>

Hãy viết mã JavaScript để thêm một sự kiện "**click**" vào nút có id là "**myButton**" và khi click vào nút, in ra thông báo "Xin chúc mừng bạn**!**" trong thẻ div.

**Câu 6: OOP (1 điểm)**

Viết một lớp JavaScript có tên là `**Rectangle**` để mô phỏng một hình chữ nhật. Lớp này có các thuộc tính là chiều dài và chiều rộng, và các phương thức tính diện tích và chu vi của hình chữ nhật.

**Câu 7: ASYNCHRONOUS VÀ AJAX (1 điểm)**

Hãy viết mã JavaScript sử dụng `**fetch**` để gửi một yêu cầu GET đến một API <https://jsonplaceholder.typicode.com/todos> chỉ lấy các công việc đã hoàn thành tức "completed: true” và dùng DOM để hiển thị “title” ra màn hình

**Câu 8: PROMISES (1 điểm)**

Viết một hàm JavaScript có tên là **delayedSum** nhận vào hai tham số là a và b. Hàm này sẽ trả về một **Promise** sau khoảng thời gian chờ 1 giây. **Promise** sẽ giải quyết với kết quả là tổng của a và b.

**Câu 9: ERROR HANDLING (1 điểm)**

Viết một đoạn mã JavaScript sử dụng khối **try-catch** để bắt lỗi khi thực hiện phép chia cho 0. Trong khối try, hãy thực hiện phép chia 10 / 0. Nếu có lỗi xảy ra, hãy in ra thông báo lỗi "**Cannot divide by zero!**".