JAVA SCRIPT BASIC CODING TEST

- **What is console.log() used for in JavaScript? Provide an example. (2)**

  The `console.log()` method in JavaScript is used to print messages to the console. This is often used for debugging purposes, to display variable values, or to provide information about the execution flow of the program.

- **2. Explain the scope of variables declared with var, let, and const. (3)**

1. `var` is function-scoped and hoisted.
2. `let` is block-scoped and hoisted without initialization.
3. `const` is block-scoped, hoisted without initialization, and must be initialized at declaration. The reference cannot be changed, but the contents of objects or arrays can be modified.

- **3. Can JavaScript identifiers start with numbers? Explain with an example. (2)**

  No, JavaScript identifiers cannot start with numbers. Identifiers in JavaScript must begin with a letter (a-z, A-Z), a dollar sign ($), or an underscore (_). After the first character, identifiers can also contain digits (0-9).

- **4. Explain the concept of dynamic typing in JavaScript. (2)**

  Dynamic typing in JavaScript means that the type of a variable is determined at runtime, not in advance. In JavaScript, you don't need to specify the data type of a variable when you declare it. A variable can hold any type of value and can change types dynamically as the program executes.

- **5. Explain how template literals can be used for concatenation. Provide an example. (2)**
- Template literals in JavaScript provide a more readable and convenient way to concatenate strings compared to the traditional concatenation using the + operator. They allow for embedding expressions within a string using `${expression}` syntax. Template literals are enclosed by backticks (`` ` ``) instead of single or double quotes.

- **6. List and describe the different types of operators in JavaScript. (4)**
- **1. Arithmetic Operators**

❖ Arithmetic operators are used to perform arithmetic on numbers.
❖ + (Addition): Adds two operands.
❖ `javascript`
❖ `Copy code`
❖ `let sum = 5 + 3; // 8`

❖ - (Subtraction): Subtracts the right operand from the left operand.
❖ `javascript`
❖ `Copy code`
❖ `let difference = 10 - 4; // 6`

❖ * (Multiplication): Multiplies two operands.
❖ `javascript`
❖ `Copy code`
❖ `let product = 7 * 6; // 42`

❖ / (Division): Divides the left operand by the right operand.
❖ `javascript`
❖ `Copy code`
❖ `let quotient = 20 / 5; // 4`

❖ % (Modulus): Returns the remainder of division of two operands.
❖ `javascript`
❖ `Copy code`
❖ `let remainder = 10 % 3; // 1`

- ++ (Increment): Increases an integer value by one.
- `javascript`
- `Copy code`

-
  ```
  let a = 5;
  a++; // a is now 6
  ```

- -- (Decrement): Decreases an integer value by one.
- `javascript`
- `Copy code`

-
  ```
  let b = 5;
  b--; // b is now 4
  ```

- **2. Assignment Operators**

- Assignment operators assign values to variables.
- = (Assignment): Assigns the value of the right operand to the left operand.
- `javascript`
- `Copy code`
- `let x = 10;`

- += (Addition assignment): Adds the right operand to the left operand and assigns the result to the left operand.
- `javascript`
- `Copy code`
- `x += 5; // x is now 15`

- -= (Subtraction assignment): Subtracts the right operand from the left operand and assigns the result to the left operand.
- `javascript`
- `Copy code`
- `x -= 3; // x is now 12`

- *= (Multiplication assignment): Multiplies the left operand by the right operand and assigns the result to the left operand.
- `javascript`
- `Copy code`
- `x *= 2; // x is now 24`

- /= (Division assignment): Divides the left operand by the right operand and assigns the result to the left operand.
- `javascript`
- `Copy code`
- `x /= 4; // x is now 6`

- %= (Modulus assignment): Takes the modulus of the left operand by the right operand and assigns the result to the left operand.
- `javascript`
- `Copy code`
- `x %= 5; // x is now 1`

3. Comparison Operators

- Comparison operators compare two values and return a Boolean value.
- == (Equal to): Returns `true` if the operands are equal.
- `javascript`
- `Copy code`
- `5 == '5'; // true`

- === (Strict equal to): Returns `true` if the operands are equal and of the same type.
- `javascript`
- `Copy code`
- `5 === '5'; // false`

- != (Not equal to): Returns `true` if the operands are not equal.
- `javascript`
- `Copy code`
- `5 != '5'; // false`

- !== (Strict not equal to): Returns `true` if the operands are not equal or not of the same type.
- `javascript`
- `Copy code`
- `5 !== '5'; // true`

- > (Greater than): Returns `true` if the left operand is greater than the right operand.
- `javascript`

- Copy code
- `10 > 6; // true`

- < (Less than): Returns `true` if the left operand is less than the right operand.
- javascript
- Copy code
- `3 < 7; // true`

- >= (Greater than or equal to): Returns `true` if the left operand is greater than or equal to the right operand.
- javascript
- Copy code
- `4 >= 4; // true`

- <= (Less than or equal to): Returns `true` if the left operand is less than or equal to the right operand.
- javascript
- Copy code
- `5 <= 8; // true`

4. Logical Operators

- Logical operators are used to combine or invert Boolean values.
- && (Logical AND): Returns `true` if both operands are `true`.
- javascript
- Copy code
- `true && false; // false`

- || (Logical OR): Returns `true` if at least one of the operands is `true`.
- javascript
- Copy code
- `true || false; // true`

- ! (Logical NOT): Inverts the Boolean value of the operand.
- javascript
- Copy code

- `!true; // false`

- **5. Bitwise Operators**

- Bitwise operators perform bitwise operations on binary representations of numbers.
- & (Bitwise AND)
- | (Bitwise OR)
- ^ (Bitwise XOR)
- ~ (Bitwise NOT)
- << (Left shift)
- >> (Right shift)
- >>> (Zero-fill right shift)
- `javascript`
- `Copy code`
-
```
let a = 5;  // 0101 in binary
let b = 3;  // 0011 in binary
let result = a & b; // 0001 (1 in decimal)
```

- **6. String Operators**

- String operators are used to concatenate strings.
- + (Concatenation): Concatenates two strings.
- `javascript`
- `Copy code`
- `let str = "Hello" + " " + "World"; // "Hello World"`

- **7. Conditional (Ternary) Operator**

- The conditional operator assigns a value based on a condition.
- `? :` (Ternary): A shorthand for an `if-else` statement.
- `javascript`
- `Copy code`

-
  ```
  let age = 18;
  let canVote = (age >= 18) ? "Yes" : "No"; // "Yes"
  ```

- **8. Type Operators**

- Type operators are used to determine the type of a variable.
- `typeof`: Returns the type of a variable.
- `javascript`
- `Copy code`
- `typeof 42; // "number"`

- `instanceof`: Checks if an object is an instance of a particular class or constructor function.
- `javascript`
- `Copy code`

-
  ```
  let date = new Date();
  console.log(date instanceof Date); // true
  ```

- **9. Comma Operator**

- The comma operator evaluates each of its operands (from left to right) and returns the value of the last operand.
- `javascript`
- `Copy code`

-
  ```
  let a = (1, 2, 3);
  console.log(a); // 3
  ```

- These operators form the core of JavaScript operations, enabling manipulation and interaction with data in various ways.

- **7. What are decision-making statements in JavaScript? Provide examples. (3)**

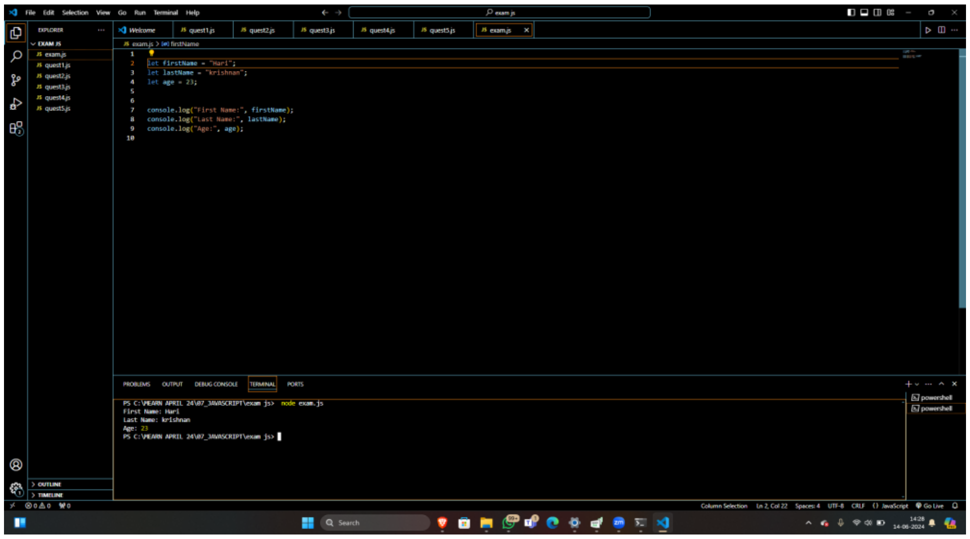
1. **`if`**: Executes code block if condition is true.
2. **`if...else`**: Executes one block if condition is true, another if false.
3. **`else if`**: Specifies new conditions if previous ones are false.
4. **`switch`**: Evaluates an expression, executes matching case block.

- ### 8. Can you reassign a const variable? (2

  No, you cannot reassign a `const` variable in JavaScript. Once a variable is declared with `const`, its value cannot be changed through reassignment. This immutability applies only to the variable binding itself, not the contents of the value it holds if it's an object or array.

Part 2: Coding Questions (30 marks)

1. Declare three variables: firstName, lastName, and age. Assign them appropriate values and print each variable's value to the console. (3)

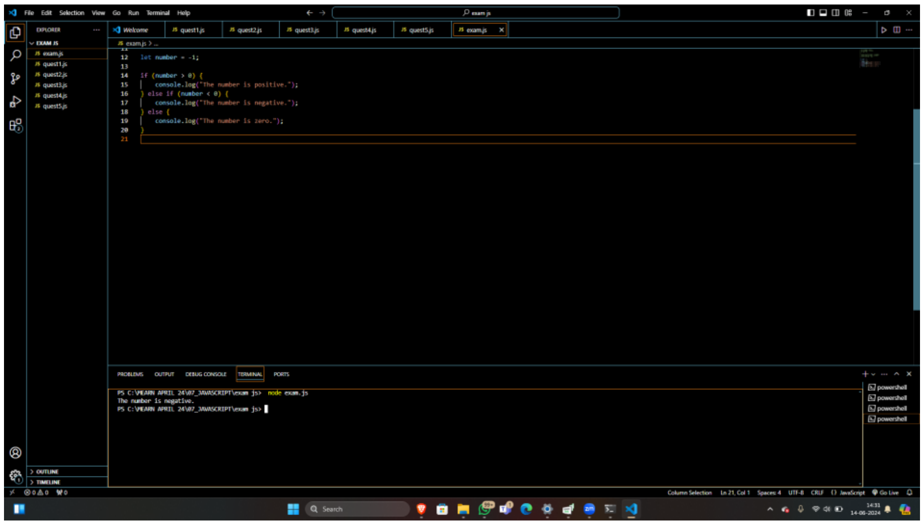


```
let firstName = "Hari";
let lastName = "krishnan";
let age = 23;


console.log("First Name:", firstName);
console.log("Last Name:", lastName);
console.log("Age:", age);
```

```
PS C:\MEARN APRIL 24\07_JAVASCRIPT\exam js>  node exam.js
First Name: Hari
Last Name: krishnan
Age: 23
PS C:\MEARN APRIL 24\07_JAVASCRIPT\exam js>
```

2. Given a number, write a code snippet to check if the number is positive, negative, or zero. Print the result to the console. (3)

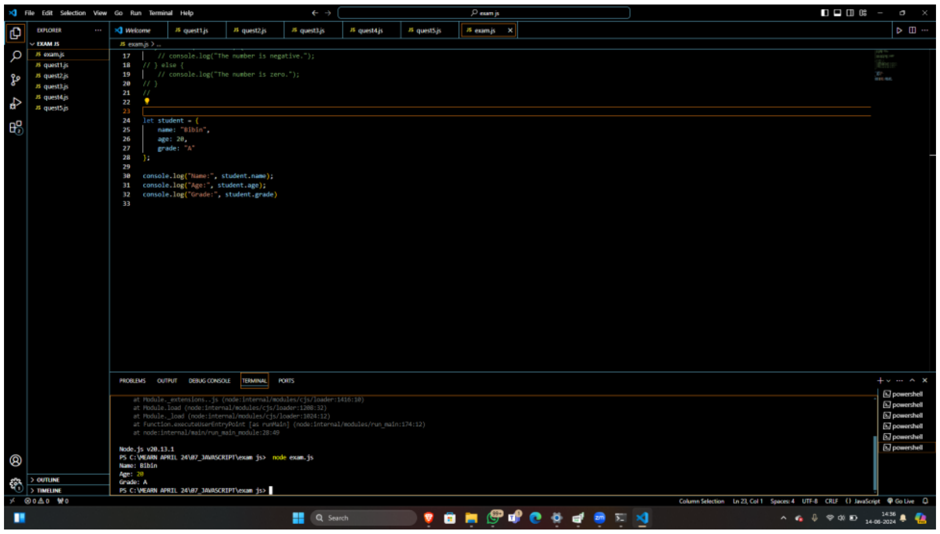
3. Write a for loop that prints the even numbers from 1 to 20 to the console. (3)



```javascript
let number = -1;

if (number > 0) {
    console.log("The number is positive.");
} else if (number < 0) {
    console.log("The number is negative.");
} else {
    console.log("The number is zero.");
}
```

```
PS C:\MEARN APRIL 24\07_JAVASCRIPT\exam js>  node exam.js
The number is negative.
PS C:\MEARN APRIL 24\07_JAVASCRIPT\exam js>
```

4. Create an object representing a student with properties name, age, and grade. Print each property to the console. (4)

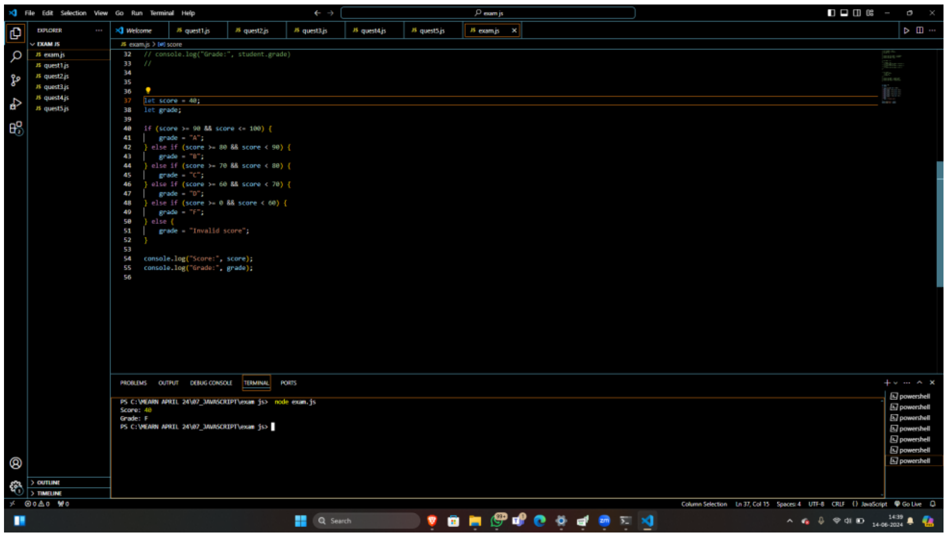


```javascript
    // console.log("The number is negative.");
// } else {
    // console.log("The number is zero.");
// }
//


let student = {
    name: "Bibin",
    age: 20,
    grade: "A"
};

console.log("Name:", student.name);
console.log("Age:", student.age);
console.log("Grade:", student.grade)
```

```
    at Module._extensions..js (node:internal/modules/cjs/loader:1416:10)
    at Module.load (node:internal/modules/cjs/loader:1208:32)
    at Module._load (node:internal/modules/cjs/loader:1024:12)
    at Function.executeUserEntryPoint [as runMain] (node:internal/modules/run_main:174:12)
    at node:internal/main/run_main_module:28:49

Node.js v20.13.1
PS C:\MEARN APRIL 24\07_JAVASCRIPT\exam js>  node exam.js
Name: Bibin
Age: 20
Grade: A
PS C:\MEARN APRIL 24\07_JAVASCRIPT\exam js>
```

5. Given a number score, write a code snippet to assign a grade based on the score. The grading criteria are as follows: (5)

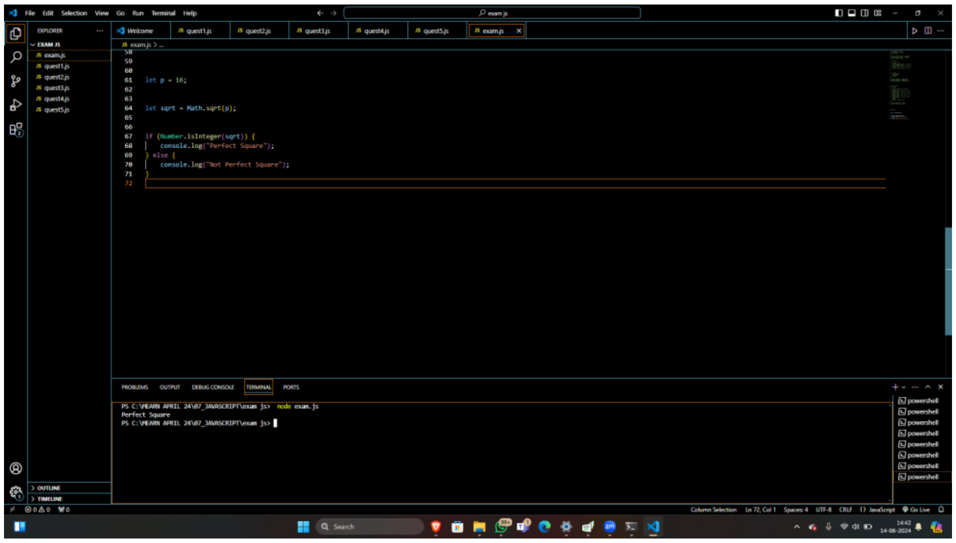
A: 90-100   B: 80-89   C: 70-79   D: 60-69   F: 0-59



```javascript
// console.log("Grade:", student.grade)
//


let score = 40;
let grade;

if (score >= 90 && score <= 100) {
    grade = "A";
} else if (score >= 80 && score < 90) {
    grade = "B";
} else if (score >= 70 && score < 80) {
    grade = "C";
} else if (score >= 60 && score < 70) {
    grade = "D";
} else if (score >= 0 && score < 60) {
    grade = "F";
} else {
    grade = "Invalid score";
}

console.log("Score:", score);
console.log("Grade:", grade);
```

```
PS C:\MEARN APRIL 24\07_JAVASCRIPT\exam js>  node exam.js
Score: 40
Grade: F
PS C:\MEARN APRIL 24\07_JAVASCRIPT\exam js>
```

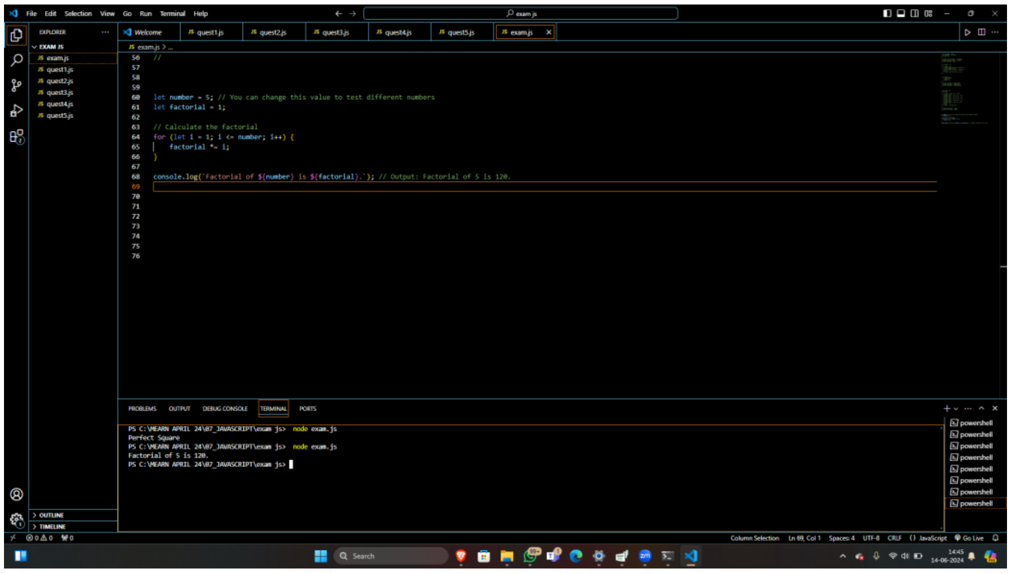
6. Check if a number p is a perfect square and print "Perfect Square" or "Not Perfect Square". (4)



```javascript
let p = 16;


let sqrt = Math.sqrt(p);


if (Number.isInteger(sqrt)) {
    console.log("Perfect Square");
} else {
    console.log("Not Perfect Square");
}
```

```
PS C:\MEARN APRIL 24\07_JAVASCRIPT\exam js>  node exam.js
Perfect Square
PS C:\MEARN APRIL 24\07_JAVASCRIPT\exam js>
```

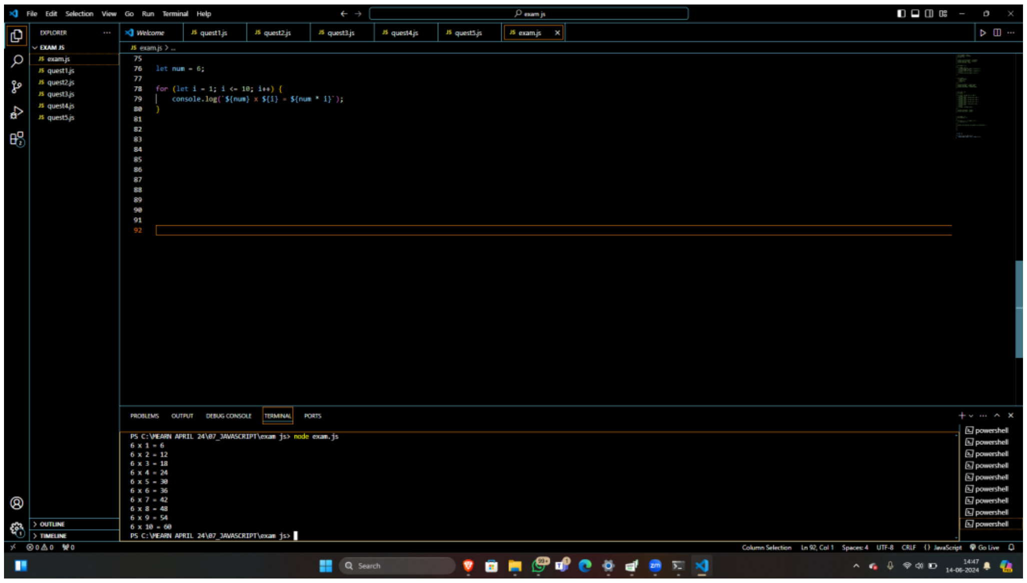
7. Write a program that calculates the factorial of a given number and logs the result to the console. (3)



```javascript
//



let number = 5; // You can change this value to test different numbers
let factorial = 1;

// Calculate the factorial
for (let i = 1; i <= number; i++) {
    factorial *= i;
}

console.log(`Factorial of ${number} is ${factorial}.`); // Output: Factorial of 5 is 120.
```

```
PS C:\MEARN APRIL 24\07_JAVASCRIPT\exam js>  node exam.js
Perfect Square
PS C:\MEARN APRIL 24\07_JAVASCRIPT\exam js>  node exam.js
Factorial of 5 is 120.
PS C:\MEARN APRIL 24\07_JAVASCRIPT\exam js>
```

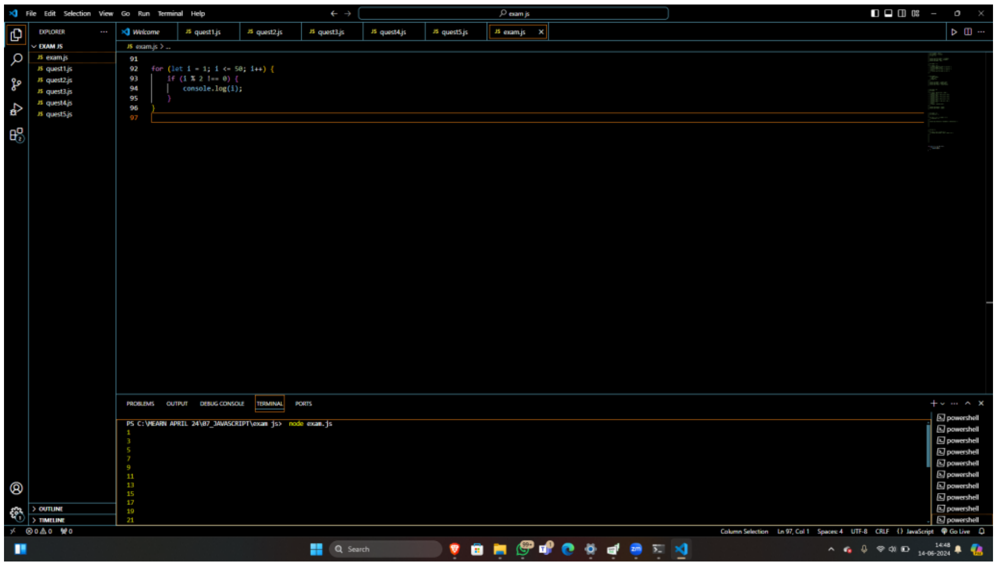
8. Write a program that logs the multiplication table of 6 (up to 10) to the console. (3)



```javascript
let num = 6;

for (let i = 1; i <= 10; i++) {
    console.log(`${num} x ${i} = ${num * i}`);
}
```

```
PS C:\MEARN APRIL 24\07_JAVASCRIPT\exam js> node exam.js
6 x 1 = 6
6 x 2 = 12
6 x 3 = 18
6 x 4 = 24
6 x 5 = 30
6 x 6 = 36
6 x 7 = 42
6 x 8 = 48
6 x 9 = 54
6 x 10 = 60
PS C:\MEARN APRIL 24\07_JAVASCRIPT\exam js>
```

9. Write a program that logs all odd numbers between 1 and 50 to the console. (2)



```javascript
for (let i = 1; i <= 50; i++) {
    if (i % 2 !== 0) {
        console.log(i);
    }
}
```

```
PS C:\MEARN APRIL 24\07_JAVASCRIPT\exam js>  node exam.js
1
3
5
7
9
11
13
15
17
19
21
```