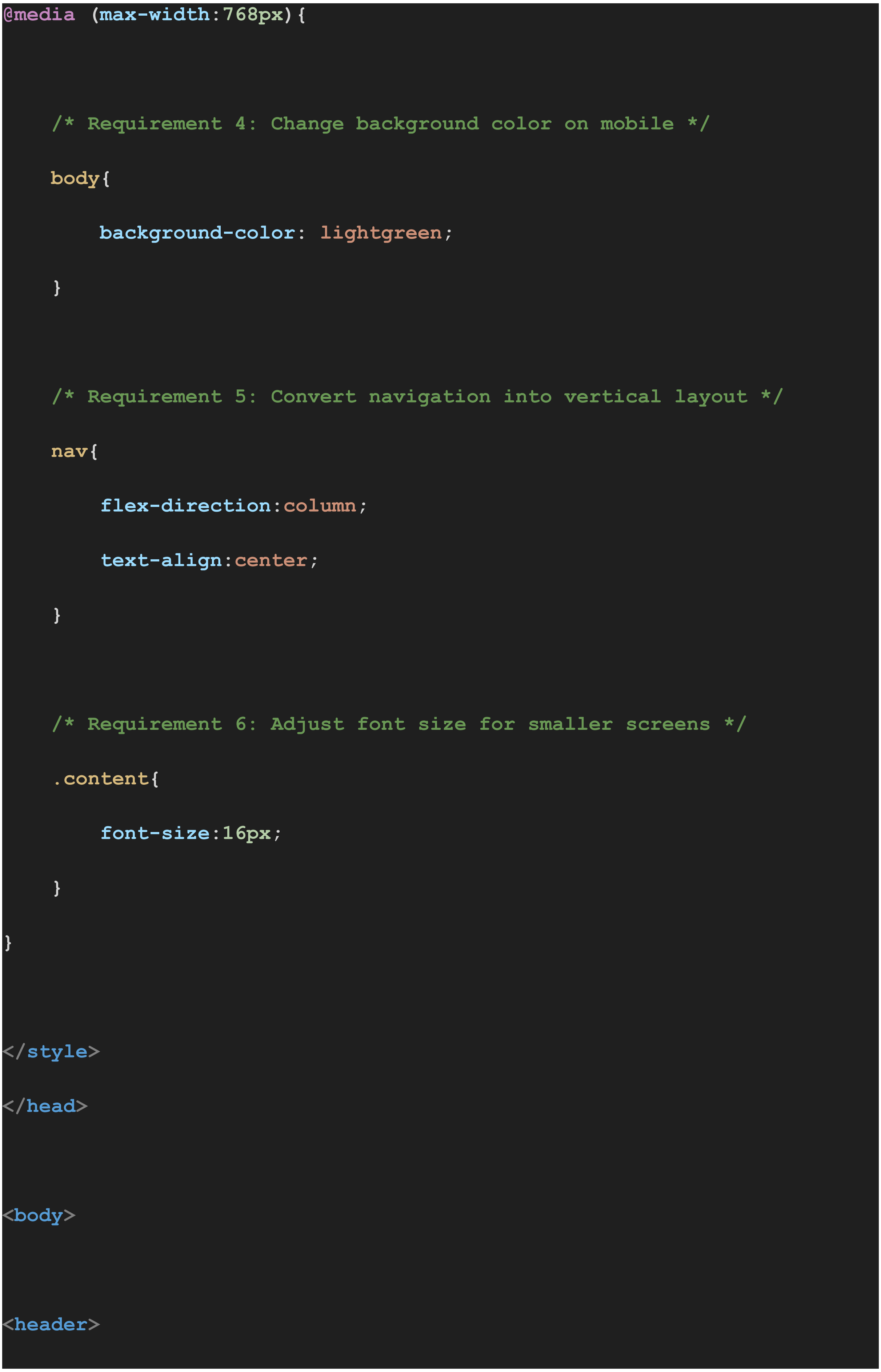
HTML5 – Day-4 Hands On – Chintha Ramakrishna

Problem 1 :-Create a webpage that becomes responsive by adding the viewport meta tag, using media queries to change background color, adjust font size, and convert navigation into a vertical layout on mobile screens.

Code :-

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">

<!-- Requirement 1: Add viewport meta tag -->
<meta name="viewport" content="width=device-width, initial-scale=1.0">

<title>Responsive Website</title>

<style>

/* Desktop default style */
body{
    margin:0;
    font-family: Arial, sans-serif;
    background-color: lightblue; /* Desktop background */
}

header{
```

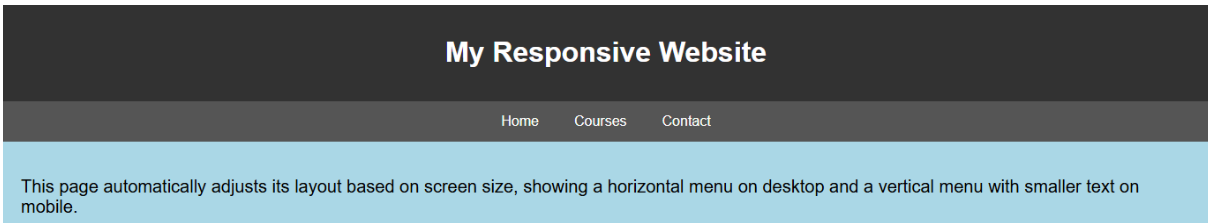
```css
    background:#333;
    color:white;
    padding:15px;
    text-align:center;
}

/* Requirement 2: Navigation horizontal layout for desktop */
nav{
    display:flex;
    justify-content:center;
    background:#555;
}

nav a{
    color:white;
    padding:14px 20px;
    text-decoration:none;
}

.content{
    padding:20px;
    font-size:20px; /* Desktop font size */
}

/* Requirement 3: Use media query for mobile */
```



```
@media (max-width:768px){

    /* Requirement 4: Change background color on mobile */
    body{
        background-color: lightgreen;
    }

    /* Requirement 5: Convert navigation into vertical layout */
    nav{
        flex-direction:column;
        text-align:center;
    }

    /* Requirement 6: Adjust font size for smaller screens */
    .content{
        font-size:16px;
    }
}

</style>
</head>

<body>

<header>
```

```html
<h1>My Responsive Website</h1>

</header>



<nav>

<a href="#">Home</a>

<a href="#">Courses</a>

<a href="#">Contact</a>

</nav>



<div class="content">

<p>This page automatically adjusts its layout based on screen size,
showing a horizontal menu on desktop and a vertical menu with smaller
text on mobile.</p>

</div>



</body>

</html>



    <!-- Hands-on Tasks:

        Add viewport meta tag to the HTML page- Done

        Use media queries to: - Done

        Change background color on mobile screen - Done

        Adjust font size for smaller screens - Done

        Convert navigation into vertical layout on mobile - Done

        Test the page using browser responsive mode. -->
```

Output :-



My Responsive Website

This page automatically adjusts its layout based on screen size, showing a horizontal menu on desktop and a vertical menu with smaller text on mobile.

Explanation :-

- The viewport meta tag helps the webpage fit properly on different screen sizes like mobile and desktop.
- Media queries are used to detect smaller screens and change styles such as background color for better appearance.
- Font size is adjusted on mobile so text remains readable without zooming.
- Navigation is changed from horizontal to vertical layout to save space on small screens.
- Testing in browser responsive mode ensures the page adapts correctly and provides a good user experience on all devices.

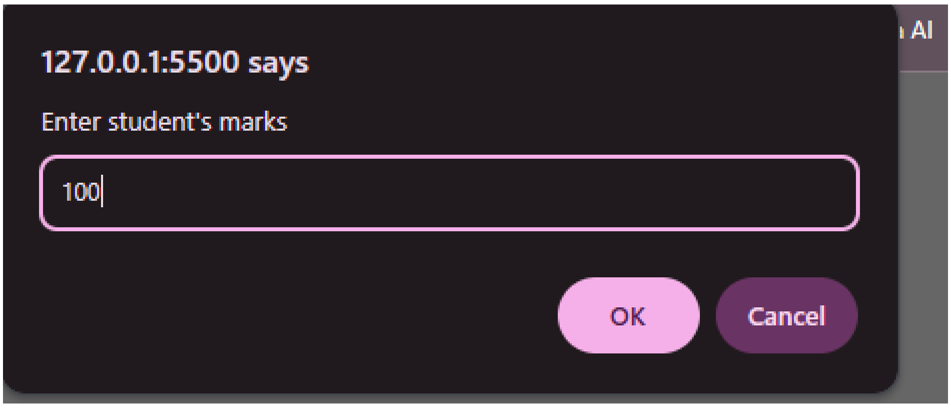
Problem 2 :- Write a JavaScript program that stores a student's marks in a variable, uses if–else conditions to assign a grade (A, B, C, or Fail), and displays the result in the console or web page.

Code :-

```
<!DOCTYPE html>

<html>

<head>

<title>Student Grade Evaluator</title>

</head>

<body>



<h2>Student Grade Evaluator</h2>
```

```
<script>

/* Scenario:
Evaluate student performance based on marks */

/* Requirement: Accept student marks as a variable */
let marks = Number(prompt("Enter student's marks"));


/* Requirement: Use if-else to assign grades */
if (isNaN(marks) || marks < 0 || marks > 100) {

    /* Display result */
    document.write("Invalid Input");

}
else if (marks >= 75) {

    /* Marks ≥ 75 → Grade A */
    document.write("Grade A");

}
else if (marks >= 60) {

    /* Marks ≥ 60 → Grade B */

    document.write("Grade B");


}

else if (marks >= 40) {


    /* Marks ≥ 40 → Grade C */

    document.write("Grade C");


}

else {


    /* Marks < 40 → Fail */

    document.write("Fail");


}




</script>


</body>

</html>
```

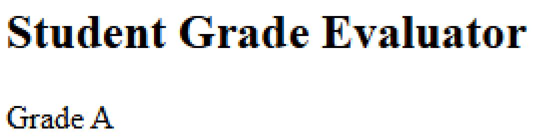
Output :-



127.0.0.1:5500 says

Enter student's marks

100



Student Grade Evaluator

Grade A

Explanation :-

1. This task helps you learn how to declare variables using `let` or `const` and store numeric marks.
2. You use comparison operators (≥, <) and if–else statements to check the mark range and decide the grade.
3. Logical decision-making allows the program to show Grade A, B, C, or Fail based on the student's performance.
4. The result is printed using console.log() or document.write(), showing how output works in JavaScript.
5. Overall, it teaches basic decision-making and variable usage in simple JavaScript programs.

Problem 3 :-

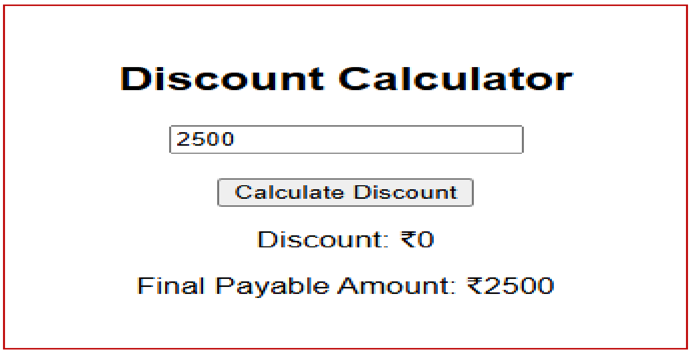
Write a JavaScript program that stores a purchase amount in a variable, applies discount rules using if–else, and displays the discount amount and final payable amount.

Code :-

```
<!DOCTYPE html>
<html>
<head>
<title>Discount Calculator</title>
```

```
<style>
body{
font-family:Arial;
text-align:center;
margin-top:50px;
}
.box{
border:1px solid #c00c0c;
padding:20px;
width:300px;
margin:auto;
}
</style>
</head>

<body>

<div class="box">

<h2>Discount Calculator</h2>

<!-- Requirement: Input purchase amount -->
<input type="number" id="amount" placeholder="Enter amount"><br><br>

<!-- Requirement: Button to calculate discount -->
<button onclick="calculate()">Calculate Discount</button>

<p id="discount"></p>
<p id="final"></p>

</div>

<script>

/* Requirement: Use arithmetic operators and if-else */
function calculate(){

/* Requirement: Get purchase amount */
let amount = Number(document.getElementById("amount").value);

/* Requirement: Primitive variables */
let discount = 0;
let finalAmount = 0;

/* Requirement: Apply discount rules */
if(amount >= 5000){
discount = amount * 0.20;
}
else if(amount >= 3000){
discount = amount * 0.10;
}
else{
discount = 0;
}

/* Requirement: Calculate final amount */
finalAmount = amount - discount;

/* Requirement: Display result on webpage */
document.getElementById("discount").innerHTML =
"Discount: ₹" + discount;

document.getElementById("final").innerHTML =
"Final Payable Amount: ₹" + finalAmount;

}

</script>

</body>
</html>
```

Output :-



Discount Calculator

Discount: ₹0

Final Payable Amount: ₹2500

Explanation :-

1. This task teaches how to store numeric values and perform calculations using arithmetic operators like multiplication and subtraction.
2. Using if–else conditions, the program decides whether the customer gets 20%, 10%, or no discount based on the amount.
3. You calculate the discount amount first and then subtract it from the total to get the final payable amount.
4. Hardcoded values help focus on logic without taking user input.
5. Overall, it shows how basic JavaScript concepts can be used to create real-world features like an online store discount system.

Problem 4:-

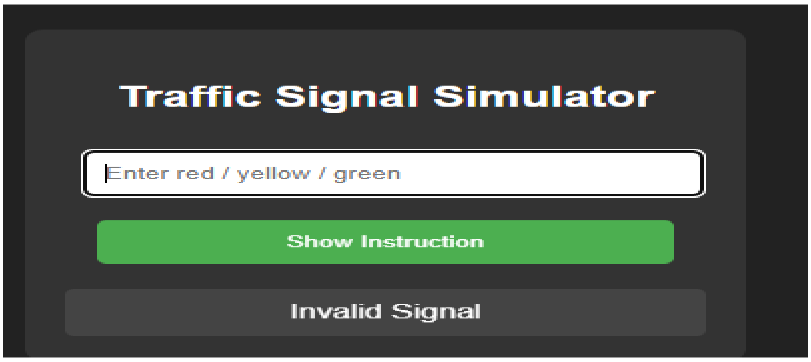
Write a JavaScript program that stores a traffic signal color in a variable and uses a switch–case statement to display the instruction (Stop, Get Ready, Go) in the console, while handling invalid input.

Code :-

```
<!DOCTYPE html>
<html>
<head>
<title>Traffic Signal Simulator</title>

<style>
body{
font-family:Arial;
background:#222;
color:white;
text-align:center;
margin-top:40px;
}
.card{
background:#333;
width:320px;
margin:auto;
padding:20px;
border-radius:10px;
}
input,button{
width:90%;
padding:10px;
margin:10px;
border:none;
border-radius:5px;
}
button{
```

```
background:#4caf50;
color:white;
}
.result{
background:#444;
padding:10px;
margin-top:10px;
border-radius:5px;
}
</style>
</head>

<body>

<div class="card">

<h2>Traffic Signal Simulator</h2>

<!-- Requirement: Store signal color using string input -->
<input type="text" id="color" placeholder="Enter red / yellow / green">

<button onclick="runSignal()">Show Instruction</button>

<div id="display" class="result"></div>

</div>

<script>

/* Requirement: Must use switch-case, string data type, no if-else */
function runSignal(){

let signal = document.getElementById("color").value.toLowerCase();

/* Requirement: Use switch statement */
switch(signal){

case "red":
/* Red → Stop */
console.log("Stop");
document.getElementById("display").innerHTML="Stop";
break;
case "yellow":
/* Yellow → Get Ready */
console.log("Get Ready");
document.getElementById("display").innerHTML="Get Ready";
break;

case "green":
/* Green → Go */
console.log("Go");
document.getElementById("display").innerHTML="Go";
break;

/* Handle invalid input gracefully */
default:
console.log("Invalid Signal");
document.getElementById("display").innerHTML="Invalid Signal";

}

/* Learning Outcome:
- Used switch effectively
- Compared string values
- Handled multiple conditions
- Understood alternative to if-else */

}

</script>

</body>
</html>
```

Output :-



Traffic Signal Simulator

Invalid Signal

Explanation :-

1. This task helps you store string values like "red", "yellow", and "green" in a variable.
2. Using a switch statement, the program checks the signal color and prints the correct instruction for each case.
3. A default case handles invalid inputs and displays a proper message instead of breaking the program.
4. The output is shown using console.log(), helping you understand console-based results.
5. Overall, it teaches how switch–case works as an alternative to if–else for multiple conditions.

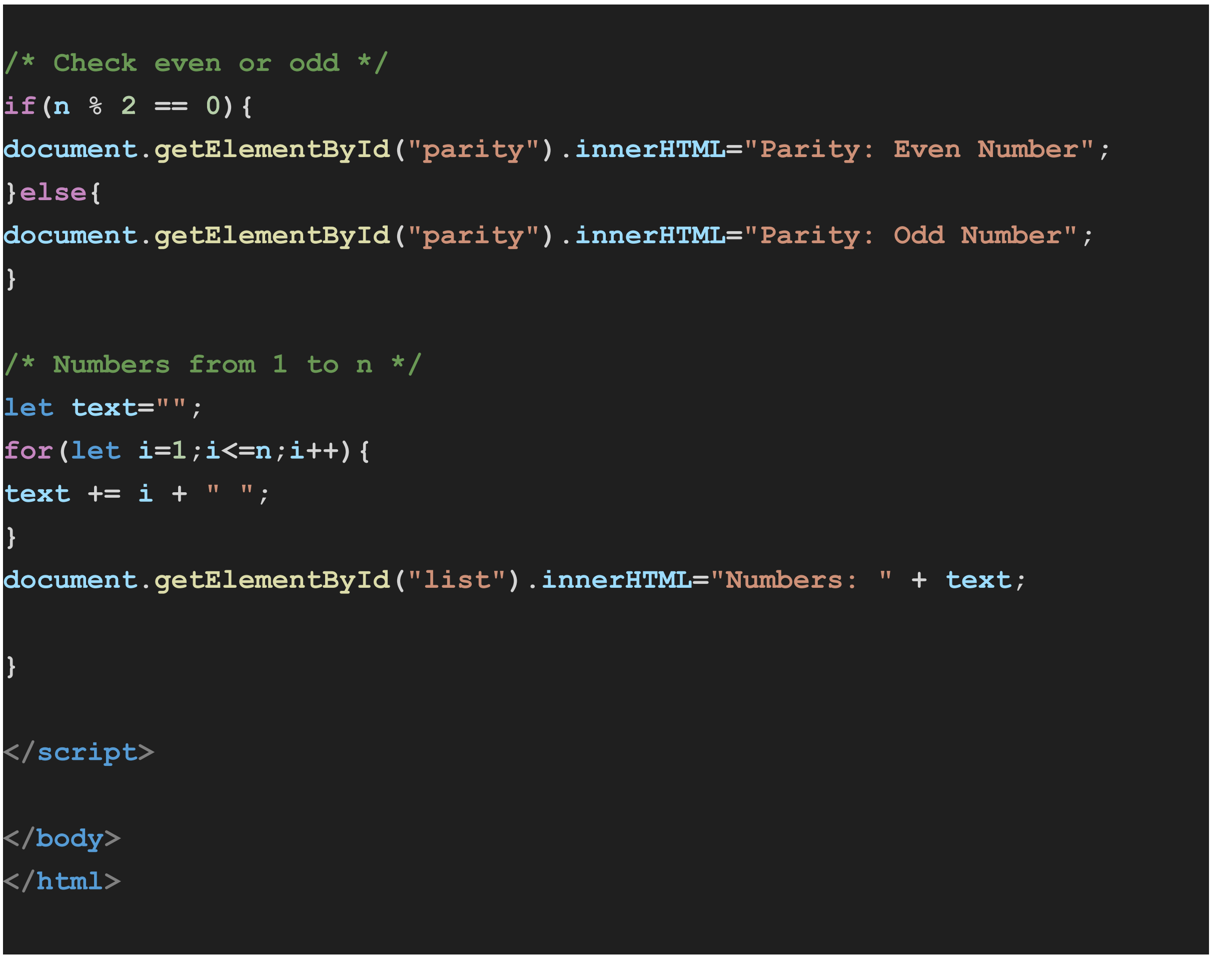
Problem 5 :-

Write a JavaScript program that stores a number in a variable, checks whether it is positive or negative using the ternary operator, checks even or odd using if–else, and prints numbers from 1 to the given number using a loop.

Code:-

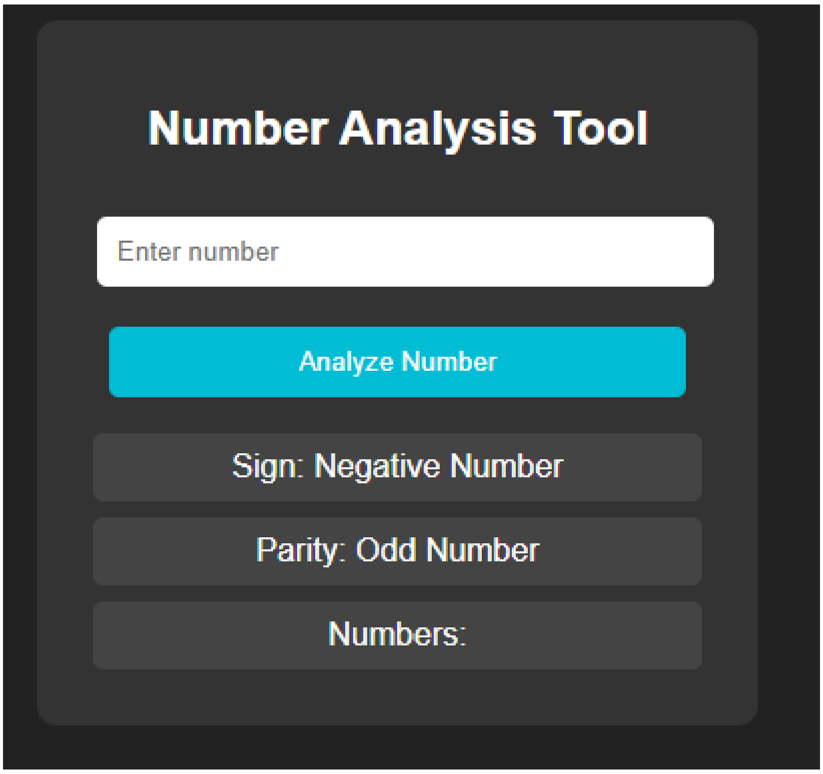
```
<!DOCTYPE html>
<html>
<head>
<title>Number Analysis Tool</title>

<style>
body{
font-family:Arial;
background:#222;
color:white;
text-align:center;
margin-top:40px;
}
.card{
background:#333;
width:320px;
margin:auto;
padding:20px;
border-radius:10px;
}
input,button{
width:90%;
padding:10px;
margin:10px;
border:none;
border-radius:5px;
}
button{
```

```
background:#00bcd4;
color:white;
}
.result{
background:#444;
padding:8px;
margin:8px;
border-radius:5px;
}
</style>
</head>

<body>

<div class="card">

<h2>Number Analysis Tool</h2>

<!-- Requirement: Enter number -->
<input type="number" id="num" placeholder="Enter number">

<!-- Requirement: Button to analyze -->
<button onclick="analyze()">Analyze Number</button>

<div id="sign" class="result"></div>
<div id="parity" class="result"></div>
<div id="list" class="result"></div>

</div>

<script>

/* Requirement: Use variables and conditional logic */
function analyze(){

let n = Number(document.getElementById("num").value);

/* Check positive or negative */
if(n >= 0){
document.getElementById("sign").innerHTML="Sign: Positive Number";
}else{
document.getElementById("sign").innerHTML="Sign: Negative Number";
}
```



```
/* Check even or odd */
if(n % 2 == 0){
document.getElementById("parity").innerHTML="Parity: Even Number";
}else{
document.getElementById("parity").innerHTML="Parity: Odd Number";
}

/* Numbers from 1 to n */
let text="";
for(let i=1;i<=n;i++){
text += i + " ";
}
document.getElementById("list").innerHTML="Numbers: " + text;

}

</script>

</body>
</html>
```

Output :-



Sign: Negative Number

Parity: Odd Number

Numbers:

Explanation :-

1. This task teaches how to store a number and apply different control flow techniques in one program.
2. The ternary operator quickly checks if the number is positive or negative and displays the result.
3. An if–else statement determines whether the number is even or odd by checking divisibility by 2.
4. A loop (for or while) prints all numbers from 1 up to the given number to show iteration.
5. Overall, it helps you combine conditions, operators, and loops to create a multi-step logical program in JavaScript.