Q1. Create a simple webpage using HTML & CSS that includes a form with input fields for name, email, and message. Use CSS to style the form.

index.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Contact Form</title>
    <style>
        body{font-family: Arial, sans-serif; text-align: center;}
        form{width: 300px; margin: auto; padding: 20px; border: 1px solid #000;}
        input, textarea{display: block; width: 100%; margin-bottom: 10px; padding: 5px;}
    </style>
</head>
<body>
    <h2>Contact Us</h2>
    <form action="submit.php" method="post">
        <input type="text" name="name" placeholder="Name" required>
        <input type="email" name="email" placeholder="Email" required>
        <textarea name="message" placeholder="Message" required></textarea>
        <input type="submit" value="Send">
    </form>
</body>
</html>
```

submit.php

```
<?php
echo "Name: " . htmlspecialchars($_POST["name"]) . "<br>";
echo "Email: " . htmlspecialchars($_POST["email"]) . "<br>";
echo "Message: " . htmlspecialchars($_POST["message"]);
?>
```

---

Q2. Design an HTML form to accept two strings and provide three options using radio buttons. Write a PHP script to perform the selected operations.

index.html

```
<form action="process.php" method="post">
    <input type="text" name="str1" placeholder="Enter first string" required>
    <input type="text" name="str2" placeholder="Enter second string" required>
    <input type="radio" name="operation" value="compare" required> Compare
```

```
    <input type="radio" name="operation" value="uppercase"> Convert to Uppercase
    <input type="radio" name="operation" value="lowercase"> Convert to Lowercase
    <input type="submit" value="Submit">
</form>
```

process.php

```
<?php
$str1 = $_POST["str1"];
$str2 = $_POST["str2"];
switch ($_POST["operation"]) {
    case "compare": echo ($str1 === $str2) ? "Strings are same" : "Strings are different"; break;
    case "uppercase": echo strtoupper($str1); break;
    case "lowercase": echo strtolower($str1); break;
}
?>
```

---

Q3. Write an HTML code to create a table displaying book details.

table.html

```
<table border="1">
    <tr><th>Bno</th><th>Bname</th><th>Price</th></tr>
    <tr><td>101</td><td>DBMS</td><td>200.50</td></tr>
    <tr><td>102</td><td>C-Prog</td><td>150.75</td></tr>
    <tr><td>103</td><td>Java</td><td>300.00</td></tr>
    <tr><td>104</td><td>PHP</td><td>250.50</td></tr>
</table>
```

---

Q4. Design an HTML form to accept book details and store them in a MySQL database.

index.html

```
<form action="insert.php" method="post">
    <input type="text" name="bno" placeholder="Book No" required>
    <input type="text" name="bname" placeholder="Book Name" required>
    <input type="text" name="price" placeholder="Price" required>
    <input type="submit" value="Submit">
</form>
```

insert.php

```
<?php
$conn = new mysqli("localhost", "root", "", "test");
if ($conn->connect_error) die("Connection failed");

$sql = "INSERT INTO books (bno, bname, price) VALUES ('$_POST[bno]', '$_POST[bname]', '$_POST[price]')";
echo $conn->query($sql) ? "Record Added" : "Error";
$conn->close();
?>
```

---

Q5. Design an HTML form to accept a number and write a PHP script to calculate its factorial using a function.

factorial.php

```
<?php
function factorial($n) {return ($n <= 1) ? 1 : $n * factorial($n - 1);}
echo "Factorial: " . factorial($_POST["num"]);
?>
.
```

---

Q6. Develop a PHP script that extracts multiple values from an associative array and prints them in key-value format.

array.php

```
<?php
$student = ["Name" => "John", "Age" => 20, "Course" => "IT"];
foreach ($student as $key => $value) {
    echo "$key: $value <br>";
}
?>
```

---

Q7. Create an HTML page with different text styles (bold, italic, underlined) and state the style of each line in text.

styles.html

```
<!DOCTYPE html>
```

```
<html>
<head><title>Text Styles</title></head>
<body>
    <p><b>Bold Text</b></p>
    <p><i>Italic Text</i></p>
    <p><u>Underlined Text</u></p>
    <p><b><i>Bold & Italic</i></b></p>
</body>
</html>
```

---

Q8. Write a PHP script to set a cookie for a username and retrieve its value on another page.

setcookie.php

```
<?php
setcookie("username", "JohnDoe", time() + 3600);
echo "Cookie Set!";
?>
```

getcookie.php

```
<?php
echo isset($_COOKIE["username"]) ? "Hello, " . $_COOKIE["username"] : "No Cookie Found";
?>
```

---

Q9. Write a PHP script to create and display a numeric array containing 5 values.

numeric_array.php

```
<?php
$arr = [10, 20, 30, 40, 50];
foreach ($arr as $value) {
    echo $value . "<br>";
}
?>
```

---

Q10. Create a MySQL table Login (username, password) and design an HTML form for authentication.

login.html

```
<form action="login.php" method="post">
    <input type="text" name="username" placeholder="Username" required>
    <input type="password" name="password" placeholder="Password" required>
    <input type="submit" value="Login">
</form>
```

login.php

```
<?php
$conn = new mysqli("localhost", "root", "", "test");
$sql = "SELECT * FROM login WHERE username='$_POST[username]' AND
password='$_POST[password]'";
$result = $conn->query($sql);
echo ($result->num_rows > 0) ? "Login Successful" : "Login Failed";
$conn->close();
?>
```

---

Q11. Write a PHP script to count the number of elements in an array and display the maximum and minimum values.

array_count.php

```
<?php
$arr = [5,10,15,20,25];
echo "Count: " . count($arr) . "<br>Max: " . max($arr) . "<br>Min: " . min($arr);
?>
```

---

Q12. Create an HTML form to accept a number and write a PHP script to sum all digits of the input number.

sum_digits.php

```
<?php
$num = $_POST["num"];
$sum = array_sum(str_split($num));
echo "Sum of Digits: $sum";
?>
```

---

Q13. Write a PHP script to count the number of times a web page has been accessed.

counter.php

```
<?php
session_start();
$_SESSION["count"] = isset($_SESSION["count"]) ? $_SESSION["count"] + 1 : 1;
echo "Page visits: " . $_SESSION["count"];
?>
```

---

Q14. Write a PHP script to display all records of MySQL emp table in tabular format.

display_emp.php

```
<?php
$conn = new mysqli("localhost", "root", "", "test");
$result = $conn->query("SELECT * FROM emp");
echo "<table border='1'>";
echo "<tr><th>ID</th><th>Name</th><th>Salary</th></tr>";
while ($row = $result->fetch_assoc()) {
    echo "<tr><td>{$row['id']}</td><td>{$row['name']}</td><td>{$row['salary']}</td></tr>";
}
echo "</table>";
$conn->close();
?>
```

---

Q15. Write a PHP script to merge two different arrays.

merge_arrays.php

```
<?php
$arr1 = [1, 2, 3];
$arr2 = [4, 5, 6];
$merged = array_merge($arr1, $arr2);
print_r($merged);
?>
```

---

Q16. Write a PHP script to insert a new item into an array at any position.

insert_array.php

```
<?php
$arr = [1, 2, 3, 4, 5];
array_splice($arr, 2, 0, 99);
print_r($arr);
?>
```

---

Q17. Write a PHP script to delete a specific employee record from the emp table.

delete_emp.php

```
<?php
$conn = new mysqli("localhost", "root", "", "test");
$sql = "DELETE FROM emp WHERE id='$_POST[id]'";
echo $conn->query($sql) ? "Record Deleted" : "Error";
$conn->close();
?>
```

---

Q18. Develop a form with radio buttons for gender selection. Use PHP to display the selected option.

gender.php

```
<form method="post">
    <input type="radio" name="gender" value="Male">Male
    <input type="radio" name="gender" value="Female">Female
    <input type="submit" value="Submit">
</form>
<?php if(isset($_POST["gender"])) echo "Selected: ".$_POST["gender"];?>
```

---

Q19. Create a class Dept with data members dno and dname. Derive class Emp from Dept and add ename, salary.

class_emp.php

```
<?php
class Dept {
    public $dno, $dname;
    function __construct($dno, $dname) { $this->dno = $dno; $this->dname = $dname; }
```

```
}

class Emp extends Dept {
    public $ename, $salary;
    function __construct($dno, $dname, $ename, $salary) {
        parent::__construct($dno, $dname);
        $this->ename = $ename;
        $this->salary = $salary;
    }
    function display() {
            echo "Dept No: $this->dno, Dept Name: $this->dname, Employee: $this->ename, Salary:
$this->salary";
    }
}

$e = new Emp(101, "IT", "John", 50000);
$e->display();
?>


—
--
```

Q20. Create a class Fruit with data members name and color. Derive class Apple from Fruit with data member price.

fruit.php

```
<?php
class Fruit {
    public $name, $color;
    function __construct($name, $color) {
        $this->name = $name;
        $this->color = $color;
    }
}

class Apple extends Fruit {
    public $price;
    function __construct($name, $color, $price) {
        parent::__construct($name, $color);
        $this->price = $price;
    }
    function display() {
        echo "Fruit: $this->name, Color: $this->color, Price: $this->price";
```

```
    }
}

$a = new Apple("Apple", "Red", 120);
$a->display();
?>
```

---

Q21. Create a class Shape with data members x and y. Derive classes Square and Circle to calculate the area.

shape.php

```
<?php
class Shape {
    public $x, $y;
    function __construct($x, $y) {
        $this->x = $x;
        $this->y = $y;
    }
}

class Square extends Shape {
    function area() { return $this->x * $this->x; }
}

class Circle extends Shape {
    function area() { return 3.14 * $this->x * $this->x; }
}

$sq = new Square(4, 0);
echo "Square Area: " . $sq->area();

$cir = new Circle(5, 0);
echo "<br>Circle Area: " . $cir->area();
?>
```

---

Q22. Create a PHP script to find the occurrences of a given element in an array.

count_occurrences.php

```
<?php
$arr = [1, 2, 3, 1, 2, 1];
```

```
$element = 1;
$count = array_count_values($arr)[$element];
echo "$element occurs $count times.";
?>
```

---

Q23. Create an HTML form with a checkbox group for selecting multiple hobbies. Write a PHP script to display the selected values.

index.html

```
<form action="hobbies.php" method="post">
    <input type="checkbox" name="hobbies[]" value="Reading"> Reading
    <input type="checkbox" name="hobbies[]" value="Sports"> Sports
    <input type="checkbox" name="hobbies[]" value="Music"> Music
    <input type="submit" value="Submit">
</form>
```

hobbies.php

```
<?php
echo "Selected: " . implode(",",$_POST["hobbies"]);
?>
```

---

Q24. Write a PHP script to accept employee details (name, address) and earning details (basic, DA, HRA). Display them properly.

employee.php

```
<?php
$name = $_POST["name"];
$address = $_POST["address"];
$basic = $_POST["basic"];
$da = $_POST["da"];
$hra = $_POST["hra"];
$salary = $basic + $da + $hra;

echo "Employee: $name <br> Address: $address <br> Total Salary: $salary";
?>
```

---

Q25. Write a PHP script to sort an integer array in ascending and descending order.

sort_array.php

```
<?php
$arr = [5,3, 9, 1, 7];
sort($arr);echo "Ascending: " . implode(", ", $arr) . "<br>";
rsort($arr); echo "Descending: " . implode(",",$arr);
?>
```

---

Q26. Write a PHP script to create a class Vegetable that contains data members name, color, and price. Include a member function to accept and display details.

vegetable.php

```
<?php
class Vegetable {
    public $name, $color, $price;
    function __construct($name, $color, $price) {
        $this->name = $name;
        $this->color = $color;
        $this->price = $price;
    }
    function display(){
        echo "Vegetable: $this->name, Color: $this->color, Price: $this->price";
    }
}

$v = new Vegetable("Carrot", "Orange", 40);
$v->display();
?>
```

---