

PB2 : Develop a PHP program that facilitates the addition, multiplication of two matrices. Utilize HTML for the user interface and PHP for the backend logic. Dynamically generate the required number of textboxes based on the specified number of rows and columns.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Matrix Operations</title>
    <style>
        table {
            border-collapse: collapse;
            margin-bottom: 20px;
        }
        th, td {
            padding: 8px;
            text-align: center;
            border: 1px solid #ddd;
        }
    </style>
</head>
<body>
    <h2>Matrix Operations</h2>
    <form action="" method="post">
        <label for="rows1">Rows (Matrix 1):</label>
        <input type="number" id="rows1" name="rows1" min="1" required>
        <label for="cols1">Columns (Matrix 1):</label>
        <input type="number" id="cols1" name="cols1" min="1" required><br><br>

        <label for="rows2">Rows (Matrix 2):</label>
        <input type="number" id="rows2" name="rows2" min="1" required>
        <label for="cols2">Columns (Matrix 2):</label>
        <input type="number" id="cols2" name="cols2" min="1" required><br><br>

        <input type="submit" name="generate" value="Generate Matrices">
```

```
</form>
```

```
<?php
```

```
if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST["generate"])) {
```

```
    $rows1 = $_POST["rows1"];
```

```
    $cols1 = $_POST["cols1"];
```

```
    $rows2 = $_POST["rows2"];
```

```
    $cols2 = $_POST["cols2"];
```

```
    echo '<form action="" method="post">';
```

```
    echo "<h3>Matrix 1:</h3>";
```

```
    generateMatrixInput("matrix1", $rows1, $cols1);
```

```
    echo "<h3>Matrix 2:</h3>";
```

```
    generateMatrixInput("matrix2", $rows2, $cols2);
```

```
    echo '<br><button type="submit" name="addition">Add</button>';
```

```
    if ($cols1 == $rows2) { // Check compatibility for multiplication
```

```
        echo '<button type="submit" name="multiplication">Multiply</button>';
```

```
    } else {
```

```
        echo '<button type="button" disabled>Multiply</button>';
```

```
        echo "<p>Matrices are not compatible for multiplication. Number of columns in  
Matrix 1 must match number of rows in Matrix 2.</p>";
```

```
    }
```

```
    echo '</form>';
```

```
}
```

```
function generateMatrixInput($matrixName, $rows, $cols) {
```

```
    echo "<table>";
```

```
    for ($i = 0; $i < $rows; $i++) {
```

```
        echo "<tr>";
```

```
        for ($j = 0; $j < $cols; $j++) {
```

```

        echo '<td><input type="number" name="" . $matrixName . '[' . $i . ']' . $j . ']'
required></td>';
    }
    echo "</tr>";
}
echo "</table>";
}

if ($_SERVER["REQUEST_METHOD"] == "POST" && isset($_POST["addition"])) {
    $matrix1 = $_POST["matrix1"];
    $matrix2 = $_POST["matrix2"];
    $result = matadd($matrix1, $matrix2);

    echo "<h3>Result (Addition):</h3>";
    displayMatrix($result);
}

if ($_SERVER["REQUEST_METHOD"] == "POST" &&
isset($_POST["multiplication"])) {
    $matrix1 = $_POST["matrix1"];
    $matrix2 = $_POST["matrix2"];
    $result = matmul($matrix1, $matrix2);

    echo "<h3>Result (Multiplication):</h3>";
    displayMatrix($result);
}

function matadd($matrix1, $matrix2) {
    $result = array();
    for ($i = 0; $i < count($matrix1); $i++) {
        for ($j = 0; $j < count($matrix1[0]); $j++) {
            $result[$i][$j] = $matrix1[$i][$j] + $matrix2[$i][$j];
        }
    }
}

```

```
    return $result;
}
```

```
function matmul($matrix1, $matrix2) {
    $result = array();
    $cols1 = count($matrix1[0]);
    $rows2 = count($matrix2);

    for ($i = 0; $i < count($matrix1); $i++) {
        for ($j = 0; $j < count($matrix2[0]); $j++) {
            $result[$i][$j] = 0;
            for ($k = 0; $k < count($matrix2); $k++) {
                $result[$i][$j] += $matrix1[$i][$k] * $matrix2[$k][$j];
            }
        }
    }
    return $result;
}
```

```
function displayMatrix($matrix) {
    echo "<table>";
    foreach ($matrix as $row) {
        echo "<tr>";
        foreach ($row as $value) {
            echo "<td>$value</td>";
        }
        echo "</tr>";
    }
    echo "</table>";
}

?>
</body>
</html>
```

PB3. Write a PHP program that implements a class to add and find the difference of two distance values given in feet and inches. The user inputs are collected through an HTML form.

PB3.html

```
<!DOCTYPE html>
<html lang="en">
<head>

    <title>Distance Calculator</title>
</head>
<body>
    <h2>Distance Calculator</h2>
    <form action="PB3.php" method="post">
        <label for="feet1">Feet 1:</label>
        <input type="number" id="feet1" name="feet1" required> feet
        <input type="number" id="inches1" name="inches1" required> inches<br><br>

        <label for="feet2">Feet 2:</label>
        <input type="number" id="feet2" name="feet2" required> feet
        <input type="number" id="inches2" name="inches2" required> inches<br><br>

        <input type="submit" value="Calculate">
    </form>
</body>
</html>
```

PB3.php

```
<?php

class Distance {
    public $feet;
    public $inches;
```

```

public function __construct($feet, $inches) {
    $this->feet = $feet;
    $this->inches = $inches;
}

public function add($feet2, $inches2) {
    $ft = $this->feet + $feet2;
    $in = $this->inches + $inches2;

    if ($in >= 12) {
        $ft += floor($in / 12);
        $in = $in % 12;
    }

    return new Distance($ft, $in);
}

public function subtract($feet2, $inches2) {
    $ft = $this->feet - $feet2;
    $in = $this->inches - $inches2;

    if ($in < 0) {
        $ft--; // Reduce a foot
        $in += 12; // Add 12 inches
    }

    return new Distance($ft, $in);
}

public function getDistance() {
    return $this->feet . " feet " . $this->inches . " inches";
}
}

```

```
if ($_SERVER["REQUEST_METHOD"] == "POST") {  
    $feet1 = $_POST["feet1"];  
    $inches1 = $_POST["inches1"];  
    $feet2 = $_POST["feet2"];  
    $inches2 = $_POST["inches2"];  
  
    $distance1 = new Distance($feet1, $inches1);  
    $distance2 = new Distance($feet2, $inches2);  
  
    // Adding distances  
    $sum = $distance1->add($feet2, $inches2);  
  
    // Finding difference  
    $difference = $distance1->subtract($feet2, $inches2);  
  
    echo "<h2>Result</h2>";  
    echo "<p>Sum: " . $sum->getDistance() . "</p>";  
    echo "<p>Difference: " . $difference->getDistance() . "</p>";  
}  
?>
```