*<!-- Problem Stmt1 : Perform the following  -->*

<?php

*// 1. Create an indexed array with a list of friend names*

    $friends = ["Mrudula", "Swati", "Neha", "Samruddhi", "Shubhangi"];

*// 2. Traverse the created indexed array using a foreach loop*

    echo "Traversing the array using foreach loop:<br>";

    foreach ($friends as $friend) {

        echo $friend . "<br>";

    }

    echo "<br>";

*// 3. Retrieve a particular element of an array*

    echo "Retrieving the second element (index 1) of the array:<br>";

    echo $friends[1];

    echo "<br><br>";

*// 4. Retrieve a range of elements from an array using array\_slice()*

    echo "Retrieving a range of elements from the array using array\_slice():<br>";

    $slice = array\_slice($friends, 1, 3); *// Retrieves elements from index 1 to 3*

    print\_r($slice);

    echo "<br><br>";

*// 5. Insert an additional element into the array*

    echo "Inserting an additional element 'Frank' into the array:<br>";

    array\_push($friends, "Shruti");

    print\_r($friends);

    echo "<br><br>";

*// 6. Display the entire array element using print\_r() and var\_dump()*

    echo "Displaying the entire array element using print\_r():<br>";

    print\_r($friends);

    echo "<br><br>";

    echo "Displaying the entire array element using var\_dump():<br>";

    var\_dump($friends);

    echo "<br><br>";

*// 7. Count the array elements using count() and sizeof()*

    echo "Counting the array elements using count() and sizeof():<br>";

    echo "Count: " . count($friends) . "<br>"; *// Using count()*

    echo "Sizeof: " . sizeof($friends) . "<br>"; *// Using sizeof()*

    echo "<br>";

*// 8. Find the occurrence of each value in an array using array\_count\_values()*

    echo "Finding the occurrence of each value in the array using array\_count\_values():<br>";

    $friendCounts = array\_count\_values($friends);

    print\_r($friendCounts);

    echo "<br><br>";

*// 9. Sort the array elements using sort(), asort(), and rsort()*

    echo "Sorting the array elements using sort():<br>";

    sort($friends);

    print\_r($friends);

    echo "<br><br>";

*// Resetting array to original order for demonstration*

    $friends = ["Mrudula", "Swati", "Neha", "Samruddhi", "Shubhangi"];

    echo "Sorting the array elements using asort() (maintains key association):<br>";

    asort($friends);

    print\_r($friends);

    echo "<br><br>";

*// Resetting array to original order for demonstration*

    $friends = ["Mrudula", "Swati", "Neha", "Samruddhi", "Shubhangi"];

    echo "Sorting the array elements using rsort() (reverse sort):<br>";

    rsort($friends);

    print\_r($friends);

    echo "<br><br>";

*// Resetting array to original order for demonstration*

    $friends = ["Mrudula", "Swati", "Neha", "Samruddhi", "Shubhangi"];

*// 10. Move the pointer through the array using current(), next(), prev(), end(), and reset()*

    echo "Moving the pointer through the array:<br>";

    echo "Current: " . current($friends) . "<br>";

    echo "Next: " . next($friends) . "<br>";

    echo "Previous: " . prev($friends) . "<br>";

    echo "End: " . end($friends) . "<br>";

    echo "Reset: " . reset($friends) . "<br>";

?>

*<!-- Problem Stmt2 : Perform the following -->*

<?php

*// 1. Create the Associative array with list of month numbers and month names up to the 11th month*

    $months = [

        1 => "January",

        2 => "February",

        3 => "March",

        4 => "April",

        5 => "May",

        6 => "June",

        7 => "July",

        8 => "August",

        9 => "September",

        10 => "October",

        11 => "November"

    ];

*// 2. Insert the 12th month in the above array*

    $months[12] = "December";

*// 3. Retrieve a particular element of an array based on key*

    echo "The 5th month is: " . $months[5]; *// Outputs: May*

    echo "<br><br>";

*// 4. Display the entire array element using print\_r() and var\_dump()*

    echo "Using print\_r():<br>";

    print\_r($months);

    echo "<br><br>";

    echo "Using var\_dump():<br>";

    var\_dump($months);

    echo "<br><br>";

*// 5. Looping through an associative array using each() and list()*

*// The each() function is deprecated as of PHP 7.2.0, but for demonstration:*

*// echo "Looping through the array using each() and list():<br>";*

*// reset($months); // Reset the array pointer*

*// while (list($key, $value) = each($months)) {*

*//     echo "Key: $key; Value: $value<br>";*

*// }*

*// echo "<br>";*

*// 5. Loop through an associative array using foreach*

    echo "Looping through the array using foreach():<br>";

    foreach ($months as $key => $value) {

    echo "Key: $key; Value: $value<br>";

    }

    echo "<br>";

*// 6. Swap the keys and values in the array using array\_flip()*

    $flippedMonths = array\_flip($months);

    echo "Swapped keys and values using array\_flip():<br>";

    print\_r($flippedMonths);

    echo "<br><br>";

*// 7. Check the existence of any value and key using in\_array() and array\_key\_exists()*

*// Check if the value "May" exists in the array*

    if (in\_array("May", $months)) {

        echo "Value 'May' exists in the array.<br>";

    } else {

        echo "Value 'May' does not exist in the array.<br>";

    }

*// Check if the key 6 exists in the array*

    if (array\_key\_exists(6, $months)) {

        echo "Key 6 exists in the array.<br>";

    } else {

        echo "Key 6 does not exist in the array.<br>";

    }

    echo "<br>";

*// 8. Retrieve the key based on the value of any element using array\_search()*

    $key = array\_search("August", $months);

    echo "The key for the value 'August' is: $key";

    echo "<br><br>";

?>

*<!-- Problem Stmt3: Perform the following -->*

<?php

*// 1. Create the multidimensional array (Nested arrays)*

    $students = [

        [

            "Roll" => 1,

            "Name" => "ABC",

            "Class" => "FY",

            "Marks" => 97

        ],

        [

            "Roll" => 2,

            "Name" => "PQR",

            "Class" => "SY",

            "Marks" => 89

        ],

        [

            "Roll" => 3,

            "Name" => "XYZ",

            "Class" => "TY",

            "Marks" => 85

        ]

    ];

*// 2. Access a particular array element*

    echo "Accessing the name of the first student: " . $students[0]["Name"];

    echo "<br><br>";

*// 3. Access all the array elements using a nested for loop*

    echo "Accessing all the array elements using a nested for loop:<br>";

    for ($i = 0; $i < count($students); $i++) {

        foreach ($students[$i] as $key => $value) {

            echo "$key: $value<br>";

        }

        echo "<br>";

    }

*// 4. Displaying array elements using print\_r() and var\_dump()*

    echo "Displaying the array using print\_r():<br>";

    print\_r($students);

    echo "<br><br>";

    echo "Displaying the array using var\_dump():<br>";

    var\_dump($students);

    echo "<br><br>";

?>

*<!-- Problem Stmt4: Queue functionality of Array (First in-first out manner) -->*

<?php

*// 1. Create an array with 10 numbers*

    $queue = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

*// Display the original array*

    echo "Original array:<br>";

    print\_r($queue);

    echo "<br><br>";

*// 2. Remove the first element from the array using array\_shift()*

    $removedElement = array\_shift($queue);

    echo "Removed element: $removedElement<br>";

    echo "Array after removing the first element:<br>";

    print\_r($queue);

    echo "<br><br>";

*// 3. Add an element at the beginning of the array using array\_unshift()*

    array\_unshift($queue, 0);

    echo "Array after adding element at the beginning:<br>";

    print\_r($queue);

    echo "<br><br>";

?>

*<!-- Problem Stmt5: Stack functionality of Array (Last in First out Manner) -->*

<?php

*// 1. Create an array with 10 numbers*

    $stack = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

*// Display the original array*

    echo "Original array:<br>";

    print\_r($stack);

    echo "<br><br>";

*// 2. Add an element to the end of the array using array\_push()*

    array\_push($stack, 11);

    echo "Array after adding an element at the end using array\_push():<br>";

    print\_r($stack);

    echo "<br><br>";

*// 3. Remove the last element from the array using array\_pop()*

    $removedElement = array\_pop($stack);

    echo "Removed element: $removedElement<br>";

    echo "Array after removing the last element using array\_pop():<br>";

    print\_r($stack);

    echo "<br><br>";

?>

*<!-- Problem Stmt 6: min() and max() function in array -->*

<?php

*// 1. Initialize arrays with marks for 3 subjects for 10 students*

    $physics = [56, 78, 90, 67, 89, 45, 76, 88, 92, 81];

    $chemistry = [66, 88, 70, 77, 99, 55, 86, 98, 72, 91];

    $maths = [76, 68, 80, 87, 79, 65, 96, 78, 82, 71];

*// 2. Merge all arrays to find the maximum and minimum marks*

    $allMarks = array\_merge($physics, $chemistry, $maths);

*// 3. Find the maximum marks among all subjects*

    $maxMarks = max($allMarks);

    echo "The maximum marks among all subjects are: $maxMarks<br>";

*// 4. Find the minimum marks among all subjects*

    $minMarks = min($allMarks);

    echo "The minimum marks among all subjects are: $minMarks<br>";

?>

*<!-- Problem Stmt7: Write the PHP code for user defined function*

*o   Take one email\_id  as input from user.*

*o   Call user defined check\_email function with parameter to check whether the entered email is valid or not.*

*o   Inside function body use call standard library function preg\_match() to check whether entered string is in format of email.*

*o   Print final decision Valid/Invalid*

*-->*

 <html>

    <head>

        <title>Palindrome</title>

    </head>

    <body>

        <form action="" method="POST">

            <h2>Enter your email:</h2>

            <input type="text" name="e1" required><br><br>

            <input type="submit" name="submit" value="Validate Email">

        </form>

        <?php

*// Define the function to check if an email is valid*

            function check\_email($email) {

*// Regular expression for validating an email address*

            $pattern = '/^[\w\.-]+@[\w\.-]+\.\w+$/';

*// Use preg\_match to validate the email address*

            return preg\_match($pattern, $email);

            }

        if (isset($\_POST["submit"])) {

            $email = $\_POST['e1'];

            if (check\_email($email)) {

                echo "$email is a valid email address.<br>";

            } else {

                echo "$email is not a valid email address.<br>";

            }

        }

        ?>

    </body>

</html>

*<!-- Problem Stmt9: Find the Factorial program in PHP using recursive function -->*

<html>

    <head>

        <title>Palindrome</title>

    </head>

    <body>

        <form action="" method="POST">

            <h2>Enter a number :</h2>

            <input type="number" name="n1" required><br><br>

            <input type="submit" name="submit" value="Calculate Factorial">

        </form>

        <?php

*// Define a recursive function to calculate factorial*

            function factorial($n) {

*// Base case: factorial of 0 is 1*

            if ($n <= 1) {

                return 1;

            }

*// Recursive case: n \* factorial of (n-1)*

            return $n \* factorial($n - 1);

            }

            if (isset($\_POST["submit"])) {

                $number = $\_POST['n1'];

                if ($number >= 0) {

                    $result = factorial($number);

                    echo "The factorial of $number is: $result<br>";

                } else {

                    echo "Please enter a non-negative integer.<br>";

                }

            }

        ?>

    </body>

</html>

*<!-- Problem Stmt1 : Perform the following  -->*

<?php

*// 1. Create the array with specified elements*

$numbers = [23, 4, 5, 4, 5, 23, 5, 67, 23];

*// 2. Remove duplicate elements using array\_unique()*

$uniqueNumbers = array\_unique($numbers);

*// 3. Display the array with duplicates removed*

echo "<h2>Array with duplicates removed using array\_unique():<br></h2>";

print\_r($uniqueNumbers);

?>