



WEB DESIGN AND WEB DEVELOPMENT

ASSIGNMENT 2 - UNIT 2

NAME: Jeswanthika K

REGISTER NUMBER: 23BCS038

CLASS: III BSC CS “A”

SUBMISSION DATE: 17.08.2025

31. Remove the first element from an array in PHP using array functions.

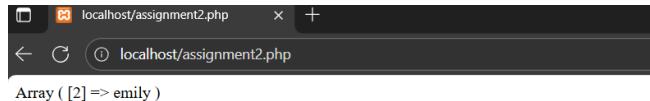
```
<?php  
$fruits = array("Apple", "Banana", "Cherry", "Mango");  
array_shift($fruits);  
print_r($fruits);  
?>
```

OUTPUT:

32. OUTPUT: a)



b)



33. Construct a PHP code to create a multidimensional array representing a matrix and display the value in the second row and third column.

```
<?php  
$matrix = array(  
    array(1, 2, 3),  
    array(4, 5, 6),  
    array(7, 8, 9)  );  
  
$value = $matrix[1][2];  
  
echo "The value in the second row and third column is: $value";  
?>
```

OUTPUT:



34. Replace all occurrences of a specific word with another word in a string using regular expressions in PHP.

```
<?php  
$text = "PHP is great. I love PHP because PHP is powerful.";  
$pattern = "/\bPHP\b/";  
$replacement = "Java";  
$result = preg_replace($pattern, $replacement, $text);  
echo $result;  
?>
```

OUTPUT:

A screenshot of a web browser window. The address bar at the top shows 'localhost/assignment2.php'. The main content area of the browser displays the text 'Java is great. I love Java because Java is powerful.'

35. Write a PHP script using an array that checks if a string contains another string and displays the result.

```
<?php  
$strings = array(  
    "Hello, welcome to PHP programming.",  
    "PHP is great for web development.",  
    "Learning PHP is fun!"  
);  
  
$search = "PHP";  
  
foreach ($strings as $str) {  
    if (strpos($str, $search) !== false) {  
        echo "The string '$str' contains '$search'.<br>";  
    } else {  
        echo "The string '$str' does NOT contain '$search'.<br>";  
    }  
}  
?>
```

OUTPUT:

```
The string 'Hello, welcome to PHP programming.' contains 'PHP'.
The string 'PHP is good for web development' contains 'PHP'.
The string 'Learning PHP is fun!' contains 'PHP'.
```

36. Create an array of fruits in PHP and display the third element.

```
<?php
$fruits = array("Apple", "Banana", "Cherry", "Mango", "Orange");
echo "The third fruit is: " . $fruits[2];
?>
```

OUTPUT:

```
The third fruit is: Cherry
```

39. A school wants to automate the calculation of student grades. Design a system that allows teachers to input student scores, calculates their grades, and generates a summary report. How would you utilize arrays and array functions to store and process the student data effectively?

```
<?php
session_start();
if (!isset($_SESSION['students'])) $_SESSION['students'] = [];
if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $name = $_POST['student_name'];
    $score = (float)$_POST['student_score'];
    $grade =
        ($score >= 90 ? "A+" : ($score >= 80 ? "A" : ($score >= 70 ? "B" : ($score >= 60 ? "C" : ($score >= 50 ? "D" : "F")))));
    $_SESSION['students'][] = ["name" => $name, "score" => $score, "grade" => $grade];
}
?>
<html>
<head><title>Student Grade Calculator</title></head>
<body>
<h2>Student Grade Input</h2>
<form method="post">
    Name: <input type="text" name="student_name" required>
```

```

Score: <input type="number" name="student_score" min="0" max="100" required>
<button type="submit">Add Student</button>
</form>

<?php if ($_SESSION['students']) { ?>
<h2>Summary Report</h2>
<table border="1" cellpadding="5">
<tr><th>Name</th><th>Score</th><th>Grade</th></tr>
<?php foreach ($_SESSION['students'] as $s) echo
"<tr><td>{$s['name']}</td><td>{$s['score']}</td><td>{$s['grade']}</td></tr>"; ?>
</table>
<?php } ?>
</body>
</html>

```

OUTPUT:



40. Write a PHP script to remove all characters from a string except a-z A-Z 0-9 or " " using an array.

```

<?php
$text = "Hello@123! Welcome_to PHP#2025.";
$chars = str_split($text);
$allowed = array_merge(range('a', 'z'), range('A', 'Z'), range('0', '9'), array(' '));
$result = array_filter($chars, function($char) use ($allowed) {
    return in_array($char, $allowed);
});
$cleanString = implode("", $result);
echo "Original: $text<br>";
echo "Cleaned: $cleanString";
?>

```

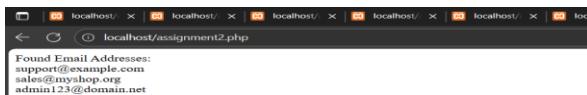
OUTPUT:



41. How can you use regular expressions to extract all email addresses from a given string using an array in PHP?

```
<?php  
  
$text = "Contact us at support@example.com or sales@myshop.org. You can also reach  
admin123@domain.net.>";  
  
$pattern = "/[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-z]{2,}/";  
  
preg_match_all($pattern, $text, $matches);  
  
$emails = $matches[0];  
  
echo "Found Email Addresses:<br>";  
  
foreach ($emails as $email) {  
  
    echo $email . "<br>";  
  
}  
  
?>
```

OUTPUT:



42. Write a PHP script to find the maximum and minimum marks from the following set of arrays \$marks1 = array(360,310,310,330,313,375,456,111,256);

```
$marks2 = array(350,340,356,330,321);  
  
$marks3 = array(630,340,570,635,434,255,298);
```

```
<?php  
  
$marks1 = array(360,310,310,330,313,375,456,111,256);  
  
$marks2 = array(350,340,356,330,321);  
  
$marks3 = array(630,340,570,635,434,255,298);  
  
$allMarks = array_merge($marks1, $marks2, $marks3);  
  
$maxMark = max($allMarks);  
  
$minMark = min($allMarks);  
  
echo "Maximum Marks: " . $maxMark . "<br>";  
  
echo "Minimum Marks: " . $minMark;?>
```

OUTPUT:



43. Develop a regular expression pattern that validates a password based on the following criteria: at least 8 characters long, contains at least one uppercase letter, one lowercase letter, one digit, and one special character.

```
<?php  
$passwords = array(  
    "Abc123!@",  
    "password",  
    "PASS123@",  
    "Passw1@",  
    "ValidPass123#"  
);  
  
$pattern = "/^(?=.*[A-Z])(?=.*[a-z])(?=.*\d)(?=.*[^A-Za-z0-9]).{8,}$/";  
  
foreach ($passwords as $pwd) {  
    if (preg_match($pattern, $pwd)) {  
        echo "$pwd is a valid password.<br>";  
    } else {  
        echo "$pwd is NOT valid.<br>";  
    }  
}  
?>
```

OUTPUT:



44. Develop a music playlist management system for a streaming service. The system should allow users to create, modify, and organize playlists. How would you use arrays and array functions to store and manipulate the song data and playlist information efficiently?

```
<?php  
session_start();  
  
$pls = &$SESSION['playlists']; $pls ??= [];
```

```

if ($_POST) {
    $a=$_POST['action']; $pl=$_POST['playlist']??""; $t=$_POST['title']??""; $sel=$pl;
    if($a=="create") $pls[$_POST['playlist_name']] ??= [];
    if($a=="add")
        $pls[$pl][]=["title"=>$t,"artist"=>$_POST['artist'],"duration"=>$_POST['duration']];
    if($a=="remove") $pls[$pl]=array_values(array_filter($pls[$pl],fn($s)=>$s['title']!=$t));
    if($a=="sort") usort($pls[$pl],fn($x,$y)=>strcmp($x['title'],$y['title']));
    if($a=="view") $sel=$pl;
}
?>

<html><body>

<h2>🎵 Playlist Manager</h2>

<form method="post">

<select name="action" onchange="this.form.submit()">
<option>--Action--
</option><option>create</option><option>add</option><option>remove</option><option>
sort</option><option>view</option>
</select>

<select name="playlist"><option value="">--select--</option>
<?php foreach($pls as $p=>$_){$s=$p==($sel??")?"selected":"";echo"<option
$s>$p</option>";}?>
</select>

<input name="playlist_name" placeholder="Playlist">
<input name="title" placeholder="Title">
<input name="artist" placeholder="Artist">
<input name="duration" placeholder="Duration">
<button>Go</button>
</form><hr>

<?php if(!empty($sel)&&isset($pls[$sel])){echo"<h3>$sel</h3>";foreach($pls[$sel] as
$s)echo"{$s['title']} - {$s['artist']} ({${$s['duration']}})<br>";}?>
</body></html>

```

OUTPUT:



45. Write a PHP function to compare two multidimensional arrays and return the difference.

```
<?php

function arrayDiffMulti($array1, $array2) {

    $result = array();

    foreach ($array1 as $key => $value) {

        if (is_array($value)) {

            if (!isset($array2[$key]) || !is_array($array2[$key])) {

                $result[$key] = $value;

            } else {

                $diff = arrayDiffMulti($value, $array2[$key]);

                if (!empty($diff)) {

                    $result[$key] = $diff;

                }

            }

        } else {

            // Compare scalar values

            if (!in_array($value, $array2)) {

                $result[$key] = $value;

            }

        }

    }

    return $result;
}

$array1 = array(
    "fruits" => array("apple", "banana", "mango"),
    "numbers" => array(1, 2, 3, 4),
    "colors" => array("red", "green")
)
```

```

);

$array2 = array(
    "fruits" => array("apple", "banana"),
    "numbers" => array(2, 3, 4, 5),
    "colors" => array("green", "blue")
);

$difference = arrayDiffMulti($array1, $array2);

echo "<pre>";
print_r($difference);
echo "</pre>";
?>

```

OUTPUT:

```

Array
(
    [fruits] => Array
        (
            [0] => mango
        )

    [numbers] => Array
        (
            [0] => 4
        )

    [colors] => Array
        (
            [0] => red
        )
)

```

46. Write a PHP program to find the index of a specific value in an array.

```

<?php

$fruits = array("apple", "banana", "mango", "orange", "grape");

$searchValue = "mango";

)index = array_search($searchValue, $fruits);

if ($index !== false) {
    echo "The index of '$searchValue' is: " . $index;
} else {
    echo "'$searchValue' was not found in the array.";
}

?>

```

OUTPUT:

```

The index of 'mango' is: 2

```

47. Delete an element from the below array. And print the array elements in PHP. \$x = array (1, 2, 3, 4, 5);

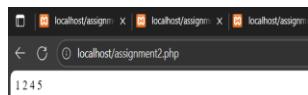
```
<?php
```

```

$x = array(1, 2, 3, 4, 5);
unset($x[2]);
$x = array_values($x);
foreach ($x as $value) {
    echo $value . " ";
}
?>

```

OUTPUT:



48. Record number handling in PHP with suitable examples.

```

<?php
echo "<h2> Record Number Handling in PHP</h2>";
$students = array("Anand", "Priya", "Karthik", "Divya", "Lakshmi");
echo "<h3>Example 1: Record Numbers with Array</h3>";
$recordNumber = 1;
foreach ($students as $name) {
    echo "Record #{$recordNumber}: $name<br>";
    $recordNumber++;
}
$studentMarks = array(
    array("name" => "Anand", "marks" => 85),
    array("name" => "Priya", "marks" => 92),
    array("name" => "Karthik", "marks" => 78),
    array("name" => "Divya", "marks" => 88)
);
echo "<h3>Example 2: Record Numbers with Multidimensional Array (Simulated
DB)</h3>";
$recordNumber = 1;
foreach ($studentMarks as $student) {

```

```

echo "Record #$recordNumber: " . $student['name'] . " - Marks: " . $student['marks'] .
"<br>";
$recordNumber++;
}

$items = array("Pen", "Pencil", "Eraser");
echo "<h3>Example 3: Using array_keys() to Get Record Numbers</h3>";
foreach (array_keys($items) as $index) {
    echo "Record #".($index + 1).": ". $items[$index]. "<br>";
}
?>

```

OUTPUT:

The screenshot shows a browser window with three tabs. The active tab is titled 'Record Number Handling in PHP'. It contains three examples:

- Example 1: Record Numbers with Array**

```
Record #1: Dhanuja - Marks: 95
Record #2: Lakshmi - Marks: 82
Record #3: Janani - Marks: 78
Record #4: Amirtha - Marks: 74
Record #5: Gopika - Marks: 58
Record #6: Jes - Marks: 60
```
- Example 2: Record Numbers with Multidimensional Array (Simulated DB)**

```
Record #1: Dhanuja - Marks: 95
Record #2: Lakshmi - Marks: 82
Record #3: Janani - Marks: 78
Record #4: Amirtha - Marks: 74
Record #5: Gopika - Marks: 58
Record #6: Jes - Marks: 60
```
- Example 3: Using array_keys() to Get Record Numbers**

```
Record #1: Dhanuja - Marks: 95
Record #2: Lakshmi - Marks: 82
Record #3: Janani - Marks: 78
Record #4: Amirtha - Marks: 74
Record #5: Gopika - Marks: 58
Record #6: Jes - Marks: 60
```

49. A sports team wants to evaluate player performance based on various statistical metrics. Design a system that utilizes numerical types and mathematical operators to calculate performance indices, averages, and rankings. How would you handle large datasets and perform complex calculations efficiently?

```

<?php
echo "<h2>Sports Team Player Performance Evaluation</h2>";
$players = [
    ["name"=>"Dhanuja","matches"=>15,"runs"=>820,"wickets"=>5],
    ["name"=>"Lakshmi","matches"=>18,"runs"=>1040,"wickets"=>8],
    ["name"=>"Janani","matches"=>20,"runs"=>970,"wickets"=>15],
    ["name"=>"Amirtha","matches"=>12,"runs"=>600,"wickets"=>2],
    ["name"=>"Gopika","matches"=>22,"runs"=>1200,"wickets"=>18],
    ["name"=>"Jes","matches"=>16,"runs"=>750,"wickets"=>6]
];
foreach($players as &$p) $p['PI']=round($p['runs']/($p['matches']+($p['wickets']*5)),2);
usort($players,fn($a,$b)>$b['PI']<=>$a['PI']);

```

```

echo "<table border=1
cellpadding=5><tr><th>Rank</th><th>Name</th><th>Matches</th><th>Runs</th><th>Wi
ckets</th><th>PI</th></tr>";
foreach($players as $i=>$p)

    echo
"<tr><td>".($i+1)."</td><td>{$p['name']}</td><td>{$p['matches']}

```

OUTPUT:

Sports Team Player Performance Evaluation					
Rank	Name	Matches	Runs	Wickets	Performance Index
1	Gopika	22	1200	18	123.5
2	Janani	20	970	15	123.5
3	Lakshmi	18	1040	8	97.78
4	Dhanuja	15	820	5	79.67
5	Jes	16	750	6	76.88
6	Amirtha	12	600	2	60

Team Statistics

Average Runs per Player: 896.67
Average Wickets per Player: 9
Total Matches Played by Team: 103

50. Construct a PHP script to lower-case and upper-case, all elements in an array.

```

<?php

$names = array("Dhanuja", "Lakshmi", "Janani", "Amirtha", "Gopika", "Jes");

echo "<h2> Original Array</h2>";

print_r($names);

$lowercaseArray = array_map('strtolower', $names);

$uppercaseArray = array_map('strtoupper', $names);

echo "<h2> Lowercase Array</h2>";

print_r($lowercaseArray);

echo "<h2> Uppercase Array</h2>";

print_r($uppercaseArray);

?>

```

OUTPUT:

The screenshot shows a browser window with three tabs labeled "Original Array", "Lowercase Array", and "Uppercase Array". Each tab displays an array structure with elements: "Dhanuja [1] => Lakshmi [2] => Janani [3] => Amirtha [4] => Gopika [5] => Jes".

51. Differentiate between array_shift() and array_unshift() in PHP.

```
<?php
$fruits = array("Apple", "Banana", "Cherry");
echo "<h3>Original Array:</h3>";
print_r($fruits);
$removed = array_shift($fruits);
echo "<br><strong>Removed (array_shift):</strong> $removed<br>";
print_r($fruits);
array_unshift($fruits, "Mango", "Pineapple");
echo "<br><strong>After array_unshift:</strong><br>";
print_r($fruits);
?>
```

OUTPUT:

The screenshot shows a browser window with the title "localhost/assignment2.php". It displays the original array, the removed element ("Apple"), and the array after unshift ("Mango", "Pineapple", "Banana", "Cherry").

52. Compare stack and queue operations using PHP with appropriate examples.

```
<?php
echo "<h2> Stack (LIFO)</h2>";
$stack = array();
array_push($stack, "Apple");
array_push($stack, "Banana");
array_push($stack, "Cherry");
echo "Stack after pushes: ";
print_r($stack);
$popped = array_pop($stack);
echo "<br>Element popped: $popped<br>";
echo "Stack after pop: ";
```

```

print_r($stack);
echo "<hr>";
echo "<h2> Queue (FIFO)</h2>";
$queue = array();
array_push($queue, "Red");
array_push($queue, "Green");
array_push($queue, "Blue");
echo "Queue after enqueues: ";
print_r($queue);
$dequeued = array_shift($queue);
echo "<br>Element dequeued: $dequeued<br>";
echo "Queue after dequeue: ";
print_r($queue);
?>

```

OUTPUT:

53. Demonstrate the difference in behaviour of `array_pop()` and `array_shift()` using a numeric array.

```

<?php
$numbers = array(10, 20, 30, 40, 50);
echo "<h2>Original Array:</h2>";
print_r($numbers);
echo "<hr><h3>Using array_pop() (removes from END)</h3>";
$popValue = array_pop($numbers);
echo "Value removed using array_pop(): $popValue<br>";
echo "Array after array_pop(): ";
print_r($numbers);
echo "<hr><h3>Reset Array for array_shift() Example</h3>";
$numbers = array(10, 20, 30, 40, 50);

```

```

echo "Original Array Again: ";
print_r($numbers);

echo "<hr><h3>Using array_shift() (removes from BEGINNING)</h3>";
$shiftValue = array_shift($numbers);

echo "Value removed using array_shift(): $shiftValue<br>";
echo "Array after array_shift(): ";
print_r($numbers);

?>

```

OUTPUT:

```

Original Array:
Array ( [0] => 10 [1] => 20 [2] => 30 [3] => 40 [4] => 50 )

Using array_pop() (removes from END)
Value removed using array_pop(): 50
Array after array_pop(): Array ( [0] => 10 [1] => 20 [2] => 30 [3] => 40 )

Reset Array for array_shift() Example
Original Array Again: Array ( [0] => 10 [1] => 20 [2] => 30 [3] => 40 [4] => 50 )

Using array_shift() (removes from BEGINNING)
Value removed using array_shift(): 10
Array after array_shift(): Array ( [0] => 20 [1] => 30 [2] => 40 [3] => 50 )

```

54. Design a PHP program that simulates a ticket booking queue using built-in array functions.

```

<?php

$ticketQueue = array();

function bookTicket(&$queue, $name) {
    array_push($queue, $name);
    echo "$name has been added to the booking queue.<br>";
}

function serveCustomer(&$queue) {
    if (!empty($queue)) {
        $customer = array_shift($queue);
        echo "$customer's ticket has been booked!<br>";
    } else {
        echo "No customers in the queue.<br>";
    }
}

bookTicket($ticketQueue, "Dhanuja");
bookTicket($ticketQueue, "Lakshmi");
bookTicket($ticketQueue, "Jannai");

```

```

bookTicket($ticketQueue, "Amirtha");
bookTicket($ticketQueue, "Gopika");
bookTicket($ticketQueue, "Jes");
echo "<hr><b>Current Queue:</b><br>";
print_r($ticketQueue);
echo "<hr><b>Serving Customers:</b><br>";
serveCustomer($ticketQueue);
serveCustomer($ticketQueue);
echo "<hr><b>Queue after serving two customers:</b><br>";
print_r($ticketQueue);
?>

```

OUTPUT:

```

Dianja has been added to the booking queue.
Lakshmi has been added to the booking queue.
Janani has been added to the booking queue.
Amirtha has been added to the booking queue.
Gopika has been added to the booking queue.
Jes has been added to the booking queue.

Current Queue:
Amitha [0] ==> Dianja [1] ==> Lakshmi [2] ==> Janani [3] ==> Amirtha [4] ==> Gopika [5] ==> Jes

Serving Customers:
Dianja's ticket has been booked!
Lakshmi's ticket has been booked!

Queue after serving two customers:
Janani [0] ==> Lakshmi [1] ==> Amirtha [2] ==> Gopika [3] ==> Jes

```

56. What are all the Functions available to sort a PHP array?

```

<?php

function reverseString($input) {

    $stack = array();

    for ($i = 0; $i < strlen($input); $i++) {
        array_push($stack, $input[$i]);
    }

    $reversed = "";

    while (!empty($stack)) {
        $reversed .= array_pop($stack);
    }

    return $reversed;
}

$original = "Hello World";
echo "Original String: " . $original . "<br>";

```

```
echo "Reversed String: " . reverseString($original);
?>
```



OUTPUT:

57. Outline the Regular Expression with appropriate examples.

```
<?php

echo "<h2>Regular Expression Examples in PHP</h2>";

$email = "test@example.com";

if (preg_match("/^[\w.-]+@[\\w.-]+\.[a-z]{2,}$/i", $email)) {
    echo " Valid Email<br>";
} else {
    echo " Invalid Email<br>";
}

$text = "There are 3 cats, 14 dogs, and 27 birds./";

preg_match_all("/\\d+/", $text, $matches);

echo "Numbers Found: ";

print_r($matches[0]);

echo "<br>";

$str = "Hello World PHP";

$result = preg_replace("/\\s+/", " ", $str);

echo "After Space Cleanup: $result<br>";

$str2 = "apple,banana;cherry orange";

$parts = preg_split("/[^a-z]+/i", $str2);

echo "Split Words: ";

print_r($parts);

echo "<br>";

$sentence = "Dhanuja and Lakshmi went to Chennai with Gopika./";

preg_match_all("/\\b[A-Z][a-z]*\\b/", $sentence, $names);

echo "Names Found: ";

print_r($names[0]);
```

```
?>
```



```
Regular Expression Examples in PHP
Valid Email
Numbers Found: Array ( [0] => 3 [1] => 14 [2] => 27 )
After Space CleanUp: Hello World PHP
Split Words: Array ( [0] => apple [1] => banana [2] => cherry [3] => orange )
Names Found: Array ( [0] => Dhanuja [1] => Lakshmi [2] => Chennai [3] => Gopika )
```

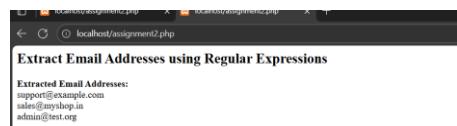
OUTPUT:

58. Construct a PHP program to extract the mail addresses in the given string using regular expression.

```
<?php
```

```
echo "<h2>Extract Email Addresses using Regular Expressions</h2>";  
  
$text = "Contact us at support@example.com, sales@myshop.in or admin@test.org for more info.;"  
  
$pattern = "/[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,6}/";  
  
if (preg_match_all($pattern, $text, $matches)) {  
  
    echo "<strong>Extracted Email Addresses:</strong><br>";  
  
    foreach ($matches[0] as $email) {  
  
        echo $email . "<br>";  
  
    }  
  
} else {  
  
    echo "No email addresses found.";  
  
}  
  
?>
```

OUTPUT:



```
Extract Email Addresses using Regular Expressions  
Extracted Email Addresses:  
support@example.com  
sales@myshop.in  
admin@test.org
```

59. Create a function that takes an array of numbers as input and returns the average value.

```
<?php  
  
function calculateAverage($numbers) {  
  
    if (count($numbers) === 0) {  
  
        return 0;  
  
    }  
  
    $sum = array_sum($numbers);
```

```

$average = $sum / count($numbers);

return $average;

}

$nums = array(10, 20, 30, 40, 50);

$avg = calculateAverage($nums);

echo "Numbers: " . implode(", ", $nums) . "<br>";

echo "Average Value: " . $avg;

?>

```



OUTPUT:

60. Write a PHP function to search a specified value within the values of an associative array.

```

<?php

function searchValueInAssocArray($array, $value) {

    $keys = array_keys($array, $value, true);

    return $keys;

}

$students = array(
    "dhanuja" => 95,
    "lakshmi" => 82,
    "janani" => 68,
    "amirtha" => 74,
    "gopika" => 55,
    "jes" => 82
);

$searchValue = 82;

$resultKeys = searchValueInAssocArray($students, $searchValue);

if (!empty($resultKeys)) {

    echo "Value $searchValue found in: " . implode(", ", $resultKeys);

} else {

    echo "Value $searchValue not found in the array.";
}

```

```
}
```

```
?>
```



OUTPUT:

61. Recall the steps to delete an element from an array?

```
<?php
```

```
echo "<h1>Delete an Element from an Array in PHP</h1>";  
  
$students = array("dhanuja", "lakshmi", "janani", "amirtha", "gopika", "jes");  
  
echo "<h3>Original Array:</h3>";  
  
print_r($students);  
  
$elementToDelete = "janani";  
  
$index = array_search($elementToDelete, $students);  
  
if ($index !== false) {  
  
    unset($students[$index]);  
  
}  
  
$students = array_values($students);  
  
echo "<h3>Array After Deleting '$elementToDelete':</h3>";  
  
print_r($students);  
  
?>
```



OUTPUT:

62. Demonstrate a PHP script that rounds the following values with 1 decimal digit precision. Sample values : 1.65 1.65 -1.54

```
<?php
```

```
echo "<h1>Rounding Values in PHP</h1>";  
  
$values = array(1.65, 1.65, -1.54);  
  
echo "<h3>Original Values:</h3>";  
  
print_r($values);  
  
$roundedValues = array_map(function($val) {  
  
    return round($val, 1);
```

```

}, $values);

echo "<h3>Rounded Values (1 decimal place):</h3>";

print_r($roundedValues);

?>

```

```

Original Values:
Array ( [0] => 1.65 [1] => 1.65 [2] => -1.54 )

Rounded Values (1 decimal place):
Array ( [0] => 1.7 [1] => 1.7 [2] => -1.5 )

```

OUTPUT:

63. Discover a function that takes an array of numbers as input and returns the sum of all the even numbers in the array.

```

<?php

echo "<h1>Sum of Even Numbers in an Array</h1>";

function sumEvenNumbers($numbers) {

    $sum = 0;

    foreach ($numbers as $num) {

        if ($num % 2 == 0) {

            $sum += $num;

        }
    }

    return $sum;
}

$array = array(1, 2, 3, 4, 5, 6, 7, 8, 9, 10);

echo "<h3>Array:</h3>";

print_r($array);

$sumEven = sumEvenNumbers($array);

echo "<h3>Sum of Even Numbers:</h3>";

echo $sumEven;

?>

```

```

Array:
Array ( [0] => 1 [1] => 2 [2] => 3 [3] => 4 [4] => 5 [5] => 6 [6] => 7 [7] => 8 [8] => 9 [9] => 10 )

Sum of Even Numbers:
30

```

OUTPUT:

64. A retail company wants to forecast future sales based on historical data. Develop a system that utilizes numerical types, mathematical operators to analyze sales trends, calculate growth rates, and generate sales forecasts using arrays in php.

```
<?php  
echo "<h1>Sales Forecast System</h1>";  
  
$sales =  
["January"=>120,"February"=>135,"March"=>150,"April"=>160,"May"=>180,"June"=>200  
];  
  
echo "<h3>Historical Sales Data:</h3><ul>";  
  
foreach($sales as $m=>$v) echo "<li>$m : $v</li>"; echo "</ul>";  
  
$g=[]; $prev=null;  
  
foreach($sales as $m=>$v){ if($prev) $g[$m]=round(((v-$prev)/$prev)*100,2); $prev=$v; }  
  
echo "<h3>Monthly Growth Rates (%):</h3><ul>";  
  
foreach($g as $m=>$r) echo "<li>$m : $r%</li>"; echo "</ul>";  
  
$avg=array_sum($g)/count($g);  
  
echo "<h3>Average Monthly Growth Rate: ".round($avg,2)."%</h3>";  
  
$last=end($sales); $f=[];  
  
foreach(["July","August","September"] as $m){ $last=round($last*(1+$avg/100),2);  
$f[$m]=$last; }  
  
echo "<h3>Forecasted Sales:</h3><ul>";  
  
foreach($f as $m=>$v) echo "<li>$m : $v</li>"; echo "</ul>";  
?  

```

OUTPUT:

The screenshot shows a web page titled "Sales Forecast System". It contains three main sections: "Historical Sales Data (in thousands)", "Monthly Growth Rates (%)", and "Forecasted Sales (in thousands)".

- Historical Sales Data (in thousands):**
 - January : 120
 - February : 135
 - March : 150
 - April : 160
 - May : 180
 - June : 200
- Monthly Growth Rates (%):**
 - February : 12.5%
 - March : 8.33%
 - April : 6.67%
 - May : 11.11%
 - June : 11.11%
- Average Monthly Growth Rate: 10.78%**
- Forecasted Sales (in thousands):**
 - July : 221.56
 - August : 234.44
 - September : 271.9

65. Demonstrate PHP script that checks if a string contains another string and displays the result.

```
<?php  
$mainString = "The quick brown fox jumps over the lazy dog";  
  
$searchString = "fox";
```

```

if (strpos($mainString, $searchString) !== false) {
    echo "The string '<b>$searchString</b>' was found in the main string.";
} else {
    echo "The string '<b>$searchString</b>' was NOT found in the main string.";
}
?>

```

OUTPUT:



67. Construct a program that tokenizes a sentence into words using regular expressions. Then, count the number of occurrences of each word and display the results.

```

<?php
$sentence = "PHP is great. PHP is easy to learn. Learning PHP is fun!";
$sentence = strtolower($sentence);
preg_match_all("/\w+/", $sentence, $words);
$wordsArray = $words[0];
$wordCounts = array_count_values($wordsArray);
echo "<h2>Word Occurrences</h2>";
echo "<table border='1' cellpadding='5'>";
echo "<tr><th>Word</th><th>Count</th></tr>";
foreach ($wordCounts as $word => $count) {
    echo "<tr><td>" . htmlspecialchars($word) . "</td><td>" . $count . "</td></tr>";
}
echo "</table>";
?>

```

OUTPUT:

Word Occurrences	
Word	Count
php	3
is	3
great	1
easy	1
to	1
learn	1
learning	1
fun	1

68. Construct a PHP script that catches a division by zero error using try-catch.

```

<?php

function divide($numerator, $denominator) {
    if ($denominator == 0) {
        throw new Exception("Error: Division by zero is not allowed.");
    }
    return $numerator / $denominator;
}

$numerator = 10;
$denominator = 0;
try {
    $result = divide($numerator, $denominator);
    echo "Result: " . $result;
} catch (Exception $e) {
    echo $e->getMessage();
}
?>

```

OUTPUT:



69. Build a PHP function to change the following array's all values to upper or lower case.

Sample arrays :

```
$Color = array('A' => 'Blue', 'B' => 'Green', 'c' => 'Red');
```

Expected Output :

Values are in lower case.

Array ([A] => blue [B] => green [c] => red)

Values are in upper case.

Array ([A] => BLUE [B] => GREEN [c] => RED)

```
<?php
```

```
$Color = array('A' => 'Blue', 'B' => 'Green', 'c' => 'Red');
```

```
function arrayToLower($arr) {
```

```

        return array_map('strtolower', $arr);
    }

function arrayToUpper($arr) {
    return array_map('strtoupper', $arr);
}

$lowerCaseArray = arrayToLower($Color);
echo "Values are in lower case.<br>";
print_r($lowerCaseArray);
echo "<br><br>";

$upperCaseArray = arrayToUpper($Color);
echo "Values are in upper case.<br>";
print_r($upperCaseArray);
?>

```

OUTPUT:

```

Values are in lower case
Array ( [A] => green [B] => red )
Values are in upper case
Array ( [A] => GREEN [B] => BLUE [C] => RED )

```

70. Create a PHP program to take input, a sequence of numbers from the user and store it in a list or array.

```

<?php

$numbers = array();

if ($_SERVER["REQUEST_METHOD"] == "POST") {
    $input = $_POST['numbers_input'];
    $numbers = preg_split("/[\s,]+/", $input);
    $numbers = array_map('floatval', $numbers);
}

?>

<html>
<head>
    <title>Number Input to Array</title>
</head>
<body>

```

```

<h2>Enter a sequence of numbers (comma or space separated):</h2>
<form method="post">
    <input type="text" name="numbers_input" size="50" placeholder="e.g. 1, 2, 3, 4, 5"
required>
    <button type="submit">Submit</button>
</form>
<?php
if (!empty($numbers)) {
    echo "<h3>Numbers stored in array:</h3>";
    echo "<pre>";
    print_r($numbers);
    echo "</pre>";
}
?>
</body>
</html>

```

OUTPUT:



Enter a sequence of numbers (comma or space separated):



Enter a sequence of numbers (comma or space separated):

Numbers stored in array:

```
Array
(
    [0] => 50
    [1] => 60
    [2] => 70
    [3] => 80
    [4] => 90
)
```