WEB DESIGN AND DEVELOPMENT

ASSIGNMENT 2

NAME: HARISMITHA.S

REG NO: 23BCS025

CLASS: III BSc.CS -A

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

```
<?php
$fruits = ["Apple", "Banana", "Mango", "Orange"];
$removedElement = array_shift($fruits);
echo "Removed Element: " . $removedElement . "\n";
print_r($fruits);
?>
```

OUTPUT:

```
Removed Element: Apple
Array
(
[0] => Banana
[1] => Mango
[2] => Orange
)
```

32. Infer the result of the following PHP code?

a.

```
<?php
$mail = "admin@example.com";
$mail = str_replace("a","@",$mail);
echo "Contact me at $mail.";
?>
```

OUTPUT:

Contact me at @dmin@ex@mple.com.

b. Determine the result of the following PHP code?

```
<?php
$names = array("alex", "jean", "emily", "jane");
$name = preg_grep("/^e/", $names);
print_r($name);
?>
```

OUTPUT:

```
Array
(
    [2] => emily
)
```

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

OUTPUT:

The value in the second row and third column is: 6

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

```
<?php

$text = "PHP is powerful. I love PHP because PHP is easy to learn.";

$result = preg_replace("/PHP/", "Python", $text);

echo $result;

?>
```

OUTPUT:

Python is powerful. I love Python because Python is easy to learn.

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

```
<?php

$sentence = "I am learning PHP programming.";

$words = array("PHP", "Java", "Python");

foreach ($words as $word) {

    if (strpos($sentence, $word) !== false) {

        echo "The string contains '$word'.<br>";

    } else {

        echo "The string does not contain '$word'.<br>";

    }

}

?>
```

OUTPUT:

The string contains 'PHP'.

The string does not contain 'Java'.

The string does not contain 'Python'.

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

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

OUTPUT:

The third fruit is: Mango

37. Explain Push and Pop in array functions.

array_push() (Push)

Purpose: Adds one or more elements to the **end** of an array.

Syntax: array_push($array, $value1, $value2, ...);

Example:

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

Result:

```
Array
(
    [0] => Apple
    [1] => Banana
    [2] => Mango
    [3] => Orange
)
```

array_pop() (Pop)

Purpose: Removes the **last element** of an array.
Syntax: array_pop($array);

Example:

```
<?php
$fruits = array("Apple", "Banana", "Mango");
$removed = array_pop($fruits);
echo "Removed: $removed\n";
print_r($fruits);
?>
```

Result:

```
Removed: Mango
Array
(
    [0] => Apple
    [1] => Banana
)
```

38. Interpret the steps to iterate over a PHP array using a while loop with an example

Steps to Iterate Over an Array with while

1. **Create an array** with elements.
2. **Find the array length** using count().
3. **Initialize an index** variable (start at 0).
4. **Use a while loop** that runs as long as the index is less than the length.
5. **Access each element** using $array[$index].
6. **Increment the index** in each loop iteration.

Example:

```
<?php

$fruits = array("Apple", "Banana", "Mango", "Orange");

$length = count($fruits);

$i = 0;

while ($i < $length) {

echo "Fruit at index $i: " . $fruits[$i] . "<br>";
```

```
$i++;  }  ?>
```

OUTPUT:

Fruit at index 0: Apple

Fruit at index 1: Banana

Fruit at index 2: Mango

Fruit at index 3: Orange

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

$students = [

"Alice" => 85,

"Bob" => 72,

"Charlie" => 90,

"David" => 60,

"Eva" => 95

];

function calculateGrade($score) {

if ($score >= 90) return "A";

elseif ($score >= 80) return "B";

elseif ($score >= 70) return "C";

elseif ($score >= 60) return "D";

else return "F";

}
```

```
echo "<h3>Student Grades:</h3>";

foreach ($students as $name => $score) {

$grade = calculateGrade($score);

echo "$name - Score: $score, Grade: $grade <br>";

}

$average = array_sum($students) / count($students);

$highestScore = max($students);

$lowestScore = min($students);

$topper = array_search($highestScore, $students);

$weakest = array_search($lowestScore, $students);

echo "<h3>Summary Report:</h3>";

echo "Class Average: " . round($average, 2) . "<br>";

echo "Top Performer: $topper (Score: $highestScore)<br>";

echo "Lowest Performer: $weakest (Score: $lowestScore)<br>";

?>
```

OUTPUT:

Student Grades:
Alice - Score: 85, Grade: B
Bob - Score: 72, Grade: C
Charlie - Score: 90, Grade: A
David - Score: 60, Grade: D
Eva - Score: 95, Grade: A

Summary Report:
Class Average: 80.4
Top Performer: Eva (Score: 95)
Lowest Performer: David (Score: 60)

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@ World! 123 #PHP ^2025*";
$chars = str_split($text);
$cleanArray = array_filter($chars, function($ch) {
   return ctype_alnum($ch) || $ch === " ";
});
$cleanText = implode("", $cleanArray);
echo "Original: $text <br>";
echo "Cleaned: $cleanText";
?>
```

OUTPUT:

Original: Hello@ World! 123 #PHP ^2025*
Cleaned: Hello World 123 PHP 2025

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

```
<?php
$text = "Please contact us at support@example.com, sales@shop.org or admin123@test.co.in.";
$pattern = "/[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-z]{2,}/";
preg_match_all($pattern, $text, $matches);
echo "<h3>Extracted Email Addresses:</h3>";
print_r($matches[0]);
?>
```

OUTPUT:

Extracted Email Addresses:
Array
(
   [0] => support@example.com
   [1] => sales@shop.org
   [2] => admin123@test.co.in
)

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);
$all_marks = array_merge($marks1, $marks2, $marks3);
$maxMark = max($all_marks);
$minMark = min($all_marks);
echo "Maximum Marks: $maxMark <br>";
echo "Minimum Marks: $minMark <br>";
?>
```

OUTPUT:
Maximum Marks: 635
Minimum Marks: 111

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 = ["Pass123!", "weakpass", "Strong@2025", "Short1!"];
$pattern = "/^(?=.*[A-Z])(?=.*[a-z])(?=.*\d)(?=.*[@$!%*?&])[A-Za-z\d@$!%*?&]{8,}$/";
foreach ($passwords as $pwd) {
if (preg_match($pattern, $pwd)) {
echo "Valid Password: $pwd <br>";
} else {
echo "Invalid Password: $pwd <br>";
}
}
?>
```

OUTPUT:
Valid Password: Pass123!
Invalid Password: weakpass
Valid Password: Strong@2025
Invalid Password: Short1!

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
$songs = [
["title" => "Shape of You", "artist" => "Ed Sheeran", "duration" => 4.2],
["title" => "Blinding Lights", "artist" => "The Weeknd", "duration" => 3.5],
["title" => "Levitating", "artist" => "Dua Lipa", "duration" => 3.4],
["title" => "Perfect", "artist" => "Ed Sheeran", "duration" => 4.0]
];
$playlists = [];
function createPlaylist(&$playlists, $name) {
$playlists[$name] = [];
}
function addSong(&$playlists, $playlistName, $song) {
array_push($playlists[$playlistName], $song);
}
function removeSong(&$playlists, $playlistName, $title) {
$playlists[$playlistName] = array_filter(
$playlists[$playlistName],
fn($s) => $s['title'] !== $title
);
}
function displayPlaylist($playlists, $playlistName) {
echo "<h3>Playlist: $playlistName</h3>";
foreach ($playlists[$playlistName] as $song) {
echo "{$song['title']} by {$song['artist']} ({$song['duration']} mins)<br>";
}
echo "Total Songs: " . count($playlists[$playlistName]) . "<br><br>";
}
function sortPlaylistByTitle(&$playlists, $playlistName) {
usort($playlists[$playlistName], fn($a, $b) => strcmp($a['title'], $b['title']));
}

createPlaylist($playlists, "Chill Vibes");
addSong($playlists, "Chill Vibes", $songs[0]);
addSong($playlists, "Chill Vibes", $songs[1]);
addSong($playlists, "Chill Vibes", $songs[3]);

displayPlaylist($playlists, "Chill Vibes");
```

```
sortPlaylistByTitle($playlists, "Chill Vibes");

displayPlaylist($playlists, "Chill Vibes");

removeSong($playlists, "Chill Vibes", "Perfect");

displayPlaylist($playlists, "Chill Vibes");
?>
```

OUTPUT:

```
Playlist: Chill Vibes
Shape of You by Ed Sheeran (4.2 mins)
Blinding Lights by The Weeknd (3.5 mins)
Perfect by Ed Sheeran (4.0 mins)
Total Songs: 3

Playlist: Chill Vibes (sorted)
Blinding Lights by The Weeknd (3.5 mins)
Perfect by Ed Sheeran (4.0 mins)
Shape of You by Ed Sheeran (4.2 mins)
Total Songs: 3

Playlist: Chill Vibes (after removal)
Blinding Lights by The Weeknd (3.5 mins)
Shape of You by Ed Sheeran (4.2 mins)
Total Songs: 2
```

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

```
<?php
function array_diff_recursive($array1, $array2) {
   $result = [];

   foreach ($array1 as $key => $value) {
      if (is_array($value)) {
         if (!isset($array2[$key]) || !is_array($array2[$key])) {
            $result[$key] = $value;
         } else {
            $diff = array_diff_recursive($value, $array2[$key]);
            if (!empty($diff)) {
               $result[$key] = $diff;
```

```
            }
        }
        } else {
                if (!isset($array2[$key]) || $array2[$key] !== $value) {
            $result[$key] = $value;
          }
        }
    }

    return $result;
}

$array1 = [
    "name" => "Alice",
    "age" => 25,
    "skills" => ["PHP", "MySQL", "JavaScript"],
    "city" => "New York"
];
$array2 = [
    "name" => "Alice",
    "age" => 30,
    "skills" => ["PHP", "MySQL"],
    "country" => "USA"
];
$diff = array_diff_recursive($array1, $array2);
echo "<pre>";
print_r($diff);
echo "</pre>";
?>
```

OUTPUT:

```
Array
(
    [age] => 25
    [skills] => Array
        (
            [2] => JavaScript
        )

    [city] => New York
)
```

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

```
<?php
$fruits = array("Apple", "Banana", "Mango", "Orange", "Grapes");
$search = "Mango";
$index = array_search($search, $fruits);
if ($index !== false) {
   echo "The value '$search' is found at index: $index";
} else {
   echo "The value '$search' was not found in the array.";
}
?>
```

OUTPUT:

The value 'Mango' is found at index: 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:

1 2 4 5

48. Record number handling in PHP with suitable examples.

Basic Record Number Handling with Arrays

```
<?php
$students = [
   ["id" => 1, "name" => "Alice", "marks" => 85],
   ["id" => 2, "name" => "Bob", "marks" => 72],
   ["id" => 3, "name" => "Charlie", "marks" => 90],
   ["id" => 4, "name" => "David", "marks" => 65]
```

```
];
$recordNumber = 2;
$record = $students[$recordNumber];
echo "Record $recordNumber → ID: {$record['id']}, Name: {$record['name']}, Marks: {$record['marks']}";
?>
```

OUTPUT:

Record 2 → ID: 3, Name: Charlie, Marks: 90

Looping with Record Numbers

```
<?php
foreach ($students as $index => $student) {
   echo "Record $index: {$student['id']} - {$student['name']} - {$student['marks']} <br>";
}
?>
```

OUTPUT:

Record 0: 1 - Alice - 85
Record 1: 2 - Bob - 72
Record 2: 3 - Charlie - 90
Record 3: 4 - David – 65

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
$players = [
["name" => "Alice", "matches" => 10, "goals" => 15, "assists" => 5],
["name" => "Bob", "matches" => 12, "goals" => 10, "assists" => 7],
["name" => "Charlie", "matches" => 8, "goals" => 12, "assists" => 6],
["name" => "David", "matches" => 15, "goals" => 20, "assists" => 10]
];
foreach ($players as &$player) {
// Example formula: (Goals * 4 + Assists * 2) / Matches
$player['performance_index'] = round(
(($player['goals'] * 4) + ($player['assists'] * 2)) / $player['matches'], 2
);
}
```

```
usort($players, function($a, $b) {
return $b['performance_index'] <=> $a['performance_index'];
});
$totalGoals = array_sum(array_column($players, 'goals'));
$totalAssists = array_sum(array_column($players, 'assists'));
$totalMatches = array_sum(array_column($players, 'matches'));
$avgGoals = round($totalGoals / count($players), 2);
$avgAssists = round($totalAssists / count($players), 2);
echo "<h3>Player Rankings:</h3>";
foreach ($players as $rank => $player) {
echo ($rank+1) . ". {$player['name']} - Performance Index: {$player['performance_index']}<br>";
}
echo "<h3>Team Averages:</h3>";
echo "Average Goals per Player: $avgGoals<br>";
echo "Average Assists per Player: $avgAssists<br>";
?>
```

OUTPUT:

Player Rankings:
1. David - Performance Index: 6.67
2. Charlie - Performance Index: 6.5
3. Alice - Performance Index: 6
4. Bob - Performance Index: 4.17

Team Averages:
Average Goals per Player: 14.25
Average Assists per Player: 7

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

```
<?php
$fruits = array("Apple", "Banana", "MANGO", "Orange");

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

$uppercaseArray = array_map('strtoupper', $fruits);
echo "Original Array: ";
print_r($fruits);

echo "<br>Lowercase Array: ";
```

```
print_r($lowercaseArray);
echo "<br>Uppercase Array: ";
print_r($uppercaseArray);
?>
```

OUTPUT:

Original Array: Array ([0] => Apple [1] => Banana [2] => MANGO [3] => Orange)
Lowercase Array: Array ([0] => apple [1] => banana [2] => mango [3] => orange)
Uppercase Array: Array ([0] => APPLE [1] => BANANA [2] => MANGO [3] => ORANGE)

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

1. array_shift()

Definition: **Removes the** first element **of an array.**

Effect on array: Remaining elements are re-indexed starting from 0.

Return Value: The removed element.

```
<?php
$numbers = array(10, 20, 30, 40);
$removed = array_shift($numbers);
echo "Removed Element: $removed <br>";
print_r($numbers);
?>
```

OUTPUT:

Removed Element: 10

Array ([0] => 20 [1] => 30 [2] => 40)

2. array_unshift()

Definition: Adds one or more elements **to the beginning** of an array.

Effect on array: Existing elements are shifted to higher indexes.

Return Value: The new number of elements in the array.

```
<?php
$numbers = array(10, 20, 30, 40);
$newCount = array_unshift($numbers, 5, 1);
echo "New Count: $newCount <br>";
print_r($numbers);
?>
```

OUTPUT:

New Count: 6

Array ([0] => 5 [1] => 1 [2] => 10 [3] => 20 [4] => 30 [5] => 40)

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

Stack (LIFO – Last In, First Out)

```
<?php
$stack = array();
array_push($stack, "A");
array_push($stack, "B");
array_push($stack, "C");
echo "Stack after pushes: ";
print_r($stack);
$popped = array_pop($stack);
echo "<br>Popped Element: $popped";
echo "<br>Stack after pop: ";
```

```
print_r($stack);

?>
```

OUTPUT:

Stack after pushes: Array ([0] => A [1] => B [2] => C)

Popped Element: C

Stack after pop: Array ([0] => A [1] => B)

Queue (FIFO – First In, First Out)

```
<?php

$queue = array();

array_push($queue, "A");

array_push($queue, "B");

array_push($queue, "C");

echo "Queue after enqueues: ";

print_r($queue);

$dequeued = array_shift($queue);

echo "<br>Dequeued Element: $dequeued";

echo "<br>Queue after dequeue: ";

print_r($queue);

?>
```

OUTPUT:

Queue after enqueues: Array ([0] => A [1] => B [2] => C)

Dequeued Element: A

Queue after dequeue: Array ([0] => B [1] => C)

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

1. array_pop()

```
<?php

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

echo "Original Array: ";

print_r($numbers);

$removed = array_pop($numbers);

echo "<br>Removed (using array_pop): $removed";

echo "<br>Array after array_pop: ";

print_r($numbers);

?>
```

OUTPUT:

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

Removed (using array_pop): 50

Array after array_pop: Array ([0] => 10 [1] => 20 [2] => 30 [3] => 40)

2. array_shift()

```
<?php

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

echo "Original Array: ";

print_r($numbers);

$removed = array_shift($numbers);

echo "<br>Removed (using array_shift): $removed";

echo "<br>Array after array_shift: ";
```

```
print_r($numbers);

?>
```

OUTPUT:

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

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 "Ticket booked for: $name <br>";

}

function serveCustomer(&$queue) {

   if (!empty($queue)) {

      $served = array_shift($queue);

      echo " Ticket issued to: $served <br>";

   } else {

      echo " No customers in the queue.<br>";

   }

}

function displayQueue($queue) {

   if (!empty($queue)) {
```

```
        echo " Current Queue: " . implode(", ", $queue) . "<br><br>";
    } else {
        echo " Queue is empty.<br><br>";
    }
}
bookTicket($ticketQueue, "Alice");
bookTicket($ticketQueue, "Bob");
bookTicket($ticketQueue, "Charlie");
displayQueue($ticketQueue);
serveCustomer($ticketQueue);
serveCustomer($ticketQueue);
displayQueue($ticketQueue);
bookTicket($ticketQueue, "David");
displayQueue($ticketQueue);
serveCustomer($ticketQueue);
serveCustomer($ticketQueue);
serveCustomer($ticketQueue);
?>
```

OUTPUT:

Ticket booked for: Alice

Ticket booked for: Bob

Ticket booked for: Charlie

Current Queue: Alice, Bob, Charlie

Ticket issued to: Alice

Ticket issued to: Bob

Current Queue: Charlie

Ticket booked for: David

Current Queue: Charlie, David

Ticket issued to: Charlie

Ticket issued to: David

No customers in the queue.

55. Develop a PHP script that uses stack functions to reverse a string.

```
<?php
function reverseString($str) {
    $stack = array();   // empty stack
    for ($i = 0; $i < strlen($str); $i++) {
        array_push($stack, $str[$i]);
}
$reversed = "";
while (!empty($stack)) {
        $reversed .= array_pop($stack);
    }
return $reversed;
}
$original = "Hello PHP";
echo "Original String: $original <br>";
```

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

?>
```

OUTPUT:

Original String: Hello PHP

Reversed String: PHP olleH

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

```
<?php

$fruits = array("Banana", "Apple", "Mango", "Orange");

echo "Original Array: ";

print_r($fruits);

echo "<br><br>";

sort($fruits);

echo "After sort() (Ascending, values, keys reindexed): ";

print_r($fruits);

echo "<br><br>";

rsort($fruits);

echo "After rsort() (Descending, values, keys reindexed): ";

print_r($fruits);

echo "<br><br>";

$ages = array("John"=>25, "Alice"=>30, "Bob"=>20);

asort($ages);

echo "After asort() (Ascending by value, keys preserved): ";

print_r($ages);
```

```
echo "<br><br>";

arsort($ages);

echo "After arsort() (Descending by value, keys preserved): ";

print_r($ages);

echo "<br><br>";

ksort($ages);

echo "After ksort() (Ascending by key): ";

print_r($ages);

echo "<br><br>";

krsort($ages);

echo "After krsort() (Descending by key): ";

print_r($ages);

?>
```

OUTPUT:

Original Array: Array ([0] => Banana [1] => Apple [2] => Mango [3] => Orange)

After sort() (Ascending, values, keys reindexed): Array ([0] => Apple [1] => Banana [2] => Mango [3] => Orange)

After rsort() (Descending, values, keys reindexed): Array ([0] => Orange [1] => Mango [2] => Banana [3] => Apple)

After asort() (Ascending by value, keys preserved): Array ([Bob] => 20 [John] => 25 [Alice] => 30)

After arsort() (Descending by value, keys preserved): Array ([Alice] => 30 [John] => 25 [Bob] => 20)

After ksort() (Ascending by key): Array ([Alice] => 30 [Bob] => 20 [John] => 25)

After krsort() (Descending by key): Array ([John] => 25 [Bob] => 20 [Alice] => 30)

57. Outline the Regular Expression with appropriate examples

1. Matching Digits

```
<?php
$str = "My roll number is 12345";
if (preg_match("/\d+/", $str, $match)) {
echo "Found number: " . $match[0];
}
?>
```

OUTPUT:

Found number: 12345

2. Matching Email Address

```
<?php
$email = "test@example.com";
if (preg_match("/^[\w\.-]+@[\w\.-]+\.\w+$/", $email)) {
echo "Valid email";
} else {
echo "Invalid email";
}
?>
```

OUTPUT:

Valid email

3. Extract All Words Starting with a Capital Letter

```
<?php
$text = "Alice and Bob are Learning PHP";
preg_match_all("/\b[A-Z][a-z]*\b/", $text, $matches);
print_r($matches[0]);
?>
```

OUTPUT:

Array ([0] => Alice [1] => Bob [2] => Learning [3] => PHP)

4. Replacing Digits

```
<?php
$str = "Order number: 98765";
$newStr = preg_replace("/\d+/", "#####", $str);
echo $newStr;
?>
```

OUTPUT:

Order number: #####

5. Password Validation

```
<?php
$password = "MyPass@123";
if (preg_match("/^(?=.*[A-Z])(?=.*[a-z])(?=.*\d)(?=.*[@$!%*?&]).{8,}$/", $password)) {
echo "Strong Password";
} else {
echo "Weak Password";
```

```
}

?>
```

OUTPUT:

Strong Password

58. Construct a PHP program to extract the mail addresses in the given

string using regular expression.

```
<?php

$text = "You can contact us at support@example.com, sales@shop.org, or admin123@test.co.in";

$pattern = "/[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-z]{2,}/";

preg_match_all($pattern, $text, $matches);

echo "Extracted Email Addresses: <br>";

foreach ($matches[0] as $email) {

echo $email . "<br>";

}

?>
```

OUTPUT:

Extracted Email Addresses:

support@example.com

sales@shop.org

admin123@test.co.in

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;  // Avoid division by zero

   }

   $sum = array_sum($numbers);

   $count = count($numbers);

   return $sum / $count;

}

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

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

echo "Average Value: " . calculateAverage($data);

?>
```

OUTPUT:

Numbers: 10, 20, 30, 40, 50

Average Value: 30

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

```
<?php

function searchValue($array, $search) {

$key = array_search($search, $array);  // Search for value

if ($key !== false) {
```

```
return "Value '$search' found at key: '$key'";
} else {
return "Value '$search' not found in array.";
}
}
$students = array(
"101" => "Alice",
"102" => "Bob",
"103" => "Charlie",
"104" => "David"
);
echo searchValue($students, "Charlie") . "<br>";
echo searchValue($students, "Eve");
?>
```

OUTPUT:

Value 'Charlie' found at key: '103'

Value 'Eve' not found in array.

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

1.Using unset()

```
<?php
$arr = array(1, 2, 3, 4, 5);
unset($arr[2]); // removes element at index 2 (value = 3)
print_r($arr);
```

```
?>
```

OUTPUT:

```
Array ( [0] => 1 [1] => 2 [3] => 4 [4] => 5 )
```

2.Using array_splice()

```
<?php

$arr = array(1, 2, 3, 4, 5);

array_splice($arr, 2, 1); // remove 1 element at index 2

print_r($arr);

?>
```

OUTPUT:

```
Array ( [0] => 1 [1] => 2 [2] => 4 [3] => 5 )
```

3.Using array_diff() (by value)

```
<?php

$arr = array(1, 2, 3, 4, 5);

$arr = array_diff($arr, [3]); // remove value 3

print_r($arr);

?>
```

OUTPUT:

```
Array ( [0] => 1 [1] => 2 [3] => 4 [4] => 5 )
```

62. Demonstrate a PHP script which rounds the following values with 1 decimal digit precision.

Sample values : 1.65 1.65 -1.54

```
<?php
```

```
$values = array(1.65, 1.65, -1.54);

echo "Original Values: " . implode(", ", $values) . "<br><br>";

foreach ($values as $val) {

   $rounded = round($val, 1);

   echo "Original: $val → Rounded (1 decimal): $rounded <br>";

}

?>
```

OUTPUT:

Original Values: 1.65, 1.65, -1.54

Original: 1.65 → Rounded (1 decimal): 1.6

Original: 1.65 → Rounded (1 decimal): 1.6

Original: -1.54 → Rounded (1 decimal): -1.5

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

function sumEvenNumbers($arr) {

   $sum = 0;

   foreach ($arr as $num) {

      if ($num % 2 == 0) {

         $sum += $num;

      }

   }

   return $sum;

}
```

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

echo "Array: " . implode(", ", $numbers) . "<br>";

echo "Sum of even numbers = " . sumEvenNumbers($numbers);

?>
```

OUTPUT:

Array: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10

Sum of even numbers = 30

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

$sales = array(

    2019 => 120,

    2020 => 135,

    2021 => 150,

    2022 => 180,

    2023 => 210

);

echo "<h3> Historical Sales Data</h3>";

foreach ($sales as $year => $value) {

    echo "Year $year: $value <br>";

}$growthRates = array();

$previous = null;

foreach ($sales as $year => $value) {
```

```
    if ($previous !== null) {
$growth = (($value - $previous) / $previous) * 100;
        $growthRates[$year] = round($growth, 2); // % growth
    } $previous = $value;
}
echo "<h3> Yearly Growth Rates</h3>";
foreach ($growthRates as $year => $rate) {
    echo "Growth in $year: $rate% <br>";
}
$averageGrowth = array_sum($growthRates) / count($growthRates);
echo "<h3> Average Growth Rate: " . round($averageGrowth, 2) . "%</h3>";
$lastYear = array_key_last($sales);
$lastSales = $sales[$lastYear];
$forecast = array();
for ($i = 1; $i <= 3; $i++) {
    $futureYear = $lastYear + $i;
    $lastSales = $lastSales * (1 + $averageGrowth / 100);
    $forecast[$futureYear] = round($lastSales, 2);
}
echo "<h3> Sales Forecast (Next 3 Years)</h3>";
foreach ($forecast as $year => $value) {
    echo "Forecast for $year: $value <br>";
}
```

?>

OUTPUT:

Historical Sales Data

Year 2019: 120

Year 2020: 135

Year 2021: 150

Year 2022: 180

Year 2023: 210

Yearly Growth Rates

Growth in 2020: 12.5%

Growth in 2021: 11.11%

Growth in 2022: 20%

Growth in 2023: 16.67%

Average Growth Rate: 15.07%

Sales Forecast (Next 3 Years)

Forecast for 2024: 241.65

Forecast for 2025: 278.03

Forecast for 2026: 319.79

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

```
<?php

$mainString = "Welcome to PHP programming!";

$search = "PHP";

if (strpos($mainString, $search) !== false) {
```

```
    echo "The string '$search' was found in: '$mainString'";
} else {
    echo "The string '$search' was NOT found in: '$mainString'";
}
?>
```

OUTPUT:

The string 'PHP' was found in: 'Welcome to PHP programming!'

66. Difference between count() and sizeof() function in PHP.

```
<?php
$fruits = array("Apple", "Banana", "Mango");
$numbers = array(
    array(1, 2, 3),
    array(4, 5),
    array(6, 7, 8, 9)
);
echo "Using count()<br>";
echo "Fruits count: " . count($fruits) . "<br>";
echo "Numbers count (non-recursive): " . count($numbers) . "<br>";
echo "Numbers count (recursive): " . count($numbers, COUNT_RECURSIVE) . "<br><br>";
echo "Using sizeof() <br>";
echo "Fruits size: " . sizeof($fruits) . "<br>";
echo "Numbers size (non-recursive): " . sizeof($numbers) . "<br>";
echo "Numbers size (recursive): " . sizeof($numbers, COUNT_RECURSIVE) . "<br>";
```

```
?>
```

OUTPUT:

Using count()

Fruits count: 3

Numbers count (non-recursive): 3

Numbers count (recursive): 12

Using sizeof()

Fruits size: 3

Numbers size (non-recursive): 3

Numbers size (recursive): 12

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, and PHP is powerful!";

$words = preg_split("/[\s,!.?]+/", strtolower($sentence), -1, PREG_SPLIT_NO_EMPTY);

$wordCount = array_count_values($words);

echo "Original Sentence: $sentence <br><br>";

echo "Word Occurrences:<br>";


foreach ($wordCount as $word => $count) {

    echo $word . " => " . $count . "<br>";

}

?>
```

OUTPUT:

Original Sentence: PHP is great, and PHP is powerful!

Word Occurrences:

php => 2

is => 2

great => 1

and => 1

powerful => 1

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

```
<?php
$num1 = 10;
$num2 = 0;
try {
    if ($num2 == 0) {
        throw new DivisionByZeroError("Error: Division by zero not allowed.");
    }
    $result = $num1 / $num2;
    echo "Result: " . $result;
} catch (DivisionByZeroError $e) {
    echo "Caught Exception: " . $e->getMessage();
}
?>
```

OUTPUT:

Caught Exception: Error: Division by zero not allowed.

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
function changeCase($array) {
    $lowerArray = array_map('strtolower', $array);
    echo "Values are in lower case.<br>";
    print_r($lowerArray);
    echo "<br><br>";
    $upperArray = array_map('strtoupper', $array);
    echo "Values are in upper case.<br>";
    print_r($upperArray);
}
$Color = array('A' => 'Blue', 'B' => 'Green', 'c' => 'Red');
changeCase($Color);
```

```
?>
```

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)

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

```
<?php

$input = "10, 20, 30, 40, 50";

$numbers = explode(",", $input);

$numbers = array_map('trim', $numbers);

echo "Stored Numbers in Array:\n";

print_r($numbers);

?>
```

OUTPUT:

Stored Numbers in Array:

Array

([0] => 10

  [1] => 20

  [2] => 30

  [3] => 40

  [4] => 50

)