**WDD Assignment 2**

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

<?php

$arr = array(10, 20, 30, 40);

array\_shift($arr); // removes the first element

print\_r($arr);

?>

**Output:**

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

**32a. Infer the result of the following PHP code:**

<?php

$mail = "admin@example.com";

$mail = str\_replace("a","@",$mail);

echo "Contact me at $mail.";

?>

**Answer:**  
The string replaces every occurrence of "a" with "@".  
**Output:**

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

**32b. Determine the result of the following PHP code:**

<?php

$names = array("alex", "jean", "emily", "jane");

$name = preg\_grep("/^e/", $names);

print\_r($name);

?>

**Answer:**  
preg\_grep selects elements starting with e.  
**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),

array(4,5,6),

array(7,8,9)

);

echo $matrix[1][2]; // second row, third column (index starts from 0)

?>

**Output:**

6

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

<?php

$text = "The sky is blue. The blue sea is deep.";

$result = preg\_replace("/blue/", "green", $text);

echo $result;

?>

**Output:**

The sky is green. The green sea is deep.

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

<?php

$haystack = "Hello, welcome to PHP programming";

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

foreach ($needles as $needle) {

if (strpos($haystack, $needle) !== false) {

echo "$needle found in string.<br>";

} else {

echo "$needle not found in string.<br>";

}

}

?>

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

<?php

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

echo $fruits[2]; // third element

?>

**Output:**

Cherry

**37. Explain Push and Pop in array functions.**

* **array\_push()** adds elements to the end of an array.
* **array\_pop()** removes the last element from an array.

<?php

$stack = array("a","b");

array\_push($stack, "c", "d"); // push elements

print\_r($stack);

array\_pop($stack); // pop last element

print\_r($stack);

?>

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

<?php

$colors = array("Red", "Green", "Blue");

$index = 0;

while ($index < count($colors)) {

echo $colors[$index] . "<br>";

$index++;

}

?>

**39. Automate calculation of student grades using arrays and array functions.**

* Use arrays to store student names and marks.
* Use loops to calculate grades.

<?php

$students = array(

"Alice" => 85,

"Bob" => 72,

"Charlie" => 90

);

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

if ($score >= 85) $grade = "A";

elseif ($score >= 70) $grade = "B";

else $grade = "C";

echo "$name scored $score and got grade $grade.<br>";

}

?>

**40. Remove all characters from a string except a-z A-Z 0-9 or space using array or regex.**

<?php

$str = "Hello@123!! World###";

$result = preg\_replace("/[^a-zA-Z0-9 ]/", "", $str);

echo $result;

?>

**Output:**

Hello123 World

**41. Use regex to extract all email addresses from a string into an array.**

<?php

$str = "Contact us at test@example.com or sales@domain.org";

preg\_match\_all("/[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}/", $str, $matches);

print\_r($matches[0]);

?>

**42. Find maximum and minimum marks from arrays.**

<?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);

$all = array\_merge($marks1,$marks2,$marks3);

echo "Max: " . max($all) . "<br>";

echo "Min: " . min($all);

?>

**43. Regular expression to validate password.**

* Minimum 8 chars, at least one uppercase, one lowercase, one digit, one special char:

$pattern = "/^(?=.\*[A-Z])(?=.\*[a-z])(?=.\*\d)(?=.\*[@$!%\*?&])[A-Za-z\d@$!%\*?&]{8,}$/";

**44. Music playlist management using arrays.**

<?php

$playlist = array("Song1", "Song2");

array\_push($playlist, "Song3"); // add song

unset($playlist[1]); // remove song

print\_r($playlist);

?>

**45. Compare two multidimensional arrays and return difference.**

<?php

function array\_diff\_assoc\_recursive($a, $b) {

$diff = array();

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

if (is\_array($value)) {

if (!isset($b[$key]) || !is\_array($b[$key])) {

$diff[$key] = $value;

} else {

$new\_diff = array\_diff\_assoc\_recursive($value, $b[$key]);

if (!empty($new\_diff)) $diff[$key] = $new\_diff;

}

} elseif (!isset($b[$key]) || $b[$key] !== $value) {

$diff[$key] = $value;

}

}

return $diff;

}

?>

**46. Find index of specific value in array.**

<?php

$arr = array("apple","banana","cherry");

$index = array\_search("banana", $arr);

echo $index;

?>

**47. Delete an element and print array.**

<?php

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

unset($x[2]); // delete element at index 2

print\_r($x);

?>

**48. Record number handling in PHP with example.**

* Use arrays with numeric indices to store sequential records.

<?php

$records = array(101,102,103);

foreach($records as $record) {

echo "Record No: $record<br>";

}

?>

**49. Sports team performance system using arrays and math.**

<?php

$players = array("Player1"=>50, "Player2"=>70, "Player3"=>90);

$avg = array\_sum($players)/count($players);

echo "Average Score: $avg<br>";

arsort($players); // rank players

print\_r($players);

?>

**50. Lower-case and upper-case all elements in array.**

<?php

$arr = array("One","Two","Three");

print\_r(array\_map("strtolower", $arr));

print\_r(array\_map("strtoupper", $arr));

?>

**51. Difference between array\_shift() and array\_unshift().**

* array\_shift() removes first element.
* array\_unshift() adds elements at the beginning.

<?php

$arr = array(2,3);

array\_unshift($arr,1); // [1,2,3]

array\_shift($arr); // removes 1

print\_r($arr);

?>

**52. Compare stack vs queue operations using PHP.**

* **Stack:** LIFO using array\_push() and array\_pop()
* **Queue:** FIFO using array\_push() and array\_shift()

<?php

// stack

$stack=array();

array\_push($stack,1,2,3);

array\_pop($stack);

// queue

$queue=array();

array\_push($queue,1,2,3);

array\_shift($queue);

?>

**53. Difference between array\_pop() and array\_shift().**

<?php

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

array\_pop($arr); // removes last element

array\_shift($arr); // removes first element

?>

**54. Ticket booking queue simulation.**

<?php

$queue = array("User1","User2");

array\_push($queue,"User3");

echo array\_shift($queue)." booked ticket<br>";

?>

**55. Reverse string using stack functions.**

<?php

$str = "PHP";

$arr = str\_split($str);

while(!empty($arr)) echo array\_pop($arr);

?>

**56. Functions available to sort arrays.**

* sort(), rsort(), asort(), arsort(), ksort(), krsort(), natsort(), natcasesort(), usort(), uksort(), uasort().

**57. Outline regex with examples.**

* /abc/ matches "abc"
* /^[0-9]+$/ matches digits only
* /^[A-Z]/ matches starting with uppercase letter.

**58. Extract mail addresses using regex.**

<?php

$str = "Email: admin@example.com, test@mail.org";

preg\_match\_all("/[a-zA-Z0-9.\_%+-]+@[a-zA-Z0-9.-]+\.[a-z]{2,}/", $str, $matches);

print\_r($matches[0]);

?>

**59. Function returning average of array numbers.**

<?php

function average($arr){ return array\_sum($arr)/count($arr); }

echo average(array(10,20,30));

?>

**60. Search specified value within associative array.**

<?php

$arr = array("a"=>"red","b"=>"green");

$key = array\_search("green",$arr);

echo $key;

?>

**61. Steps to delete an element from array.**

* Use unset($arr[index]) or array\_splice().

<?php

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

unset($arr[1]);

print\_r($arr);

?>

**62. Round numbers with 1 decimal precision.**

<?php

echo round(1.65,1)."<br>";

echo round(-1.54,1)."<br>";

?>

**63. Function returning sum of even numbers.**

<?php

function sumEven($arr){

return array\_sum(array\_filter($arr,function($n){return $n%2==0;}));

}

echo sumEven(array(1,2,3,4,5));

?>

**64. Retail company sales forecasting system using arrays and math.**

<?php

$sales=array(1000,1200,1300,1500);

$growth=($sales[count($sales)-1]-$sales[0])/$sales[0]\*100;

echo "Growth: $growth%";

?>

**65. PHP script checks if string contains another string.**

(Same as #35 simplified.)

<?php

if(strpos("Hello World","World")!==false) echo "Found";

else echo "Not Found";

?>

**66. Difference between count() and sizeof().**

* Both return number of elements in array. sizeof() is an alias of count().

<?php

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

echo count($arr); //3

echo sizeof($arr); //3

?>

**67. Tokenize sentence and count occurrences.**

<?php

$str="Hello world hello";

$words=preg\_split("/\s+/",$str);

$count=array\_count\_values(array\_map("strtolower",$words));

print\_r($count);

?>

**68. Catch division by zero error using try-catch.**

<?php

try {

$a=10; $b=0;

if($b==0) throw new Exception("Division by zero");

echo $a/$b;

} catch(Exception $e){ echo $e->getMessage(); }

?>

**69. Change array values to upper or lower case.**

<?php

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

print\_r(array\_map("strtolower",$Color));

print\_r(array\_map("strtoupper",$Color));

?>

**70. Take user input sequence of numbers and store in array.**

<?php

$input="10 20 30";

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

print\_r($arr);

?>