PHP arrays exercises

**1.**$color = array('white', 'green', 'red', 'blue', 'black');  
Write a script which will display the following string -    
"The memory of that scene for me is like a frame of film forever frozen at that moment: the red carpet, the green lawn, the white house, the leaden sky. The new president and his first lady. - Richard M. Nixon"  
and the words 'red', 'green' and 'white' will come from $color.

<?php

$color = array('white', 'green', 'red', 'blue', 'black');

echo "The memory of that scene for me is like a frame of film forever frozen at that moment: the $color[2] carpet, the $color[1] lawn, the $color[0] house, the leaden sky. The new president and his first lady. - Richard M. Nixon"."\n";

?>

**2.**$color = array('white', 'green', 'red'')   
Write a PHP script which will display the colors in the following way :   
*Output :*   
white, green, red,

* green
* red
* white

<?php

$color = array('white', 'green', 'red');

foreach ($color as $c)

{

echo "$c, ";

}

sort($color);

echo "<ul>";

foreach ($color as $y)

{

echo "<li>$y</li>";

}

echo "</ul>";

?>

**3.**$ceu = array( "Italy"=>"Rome", "Luxembourg"=>"Luxembourg", "Belgium"=> "Brussels", "Denmark"=>"Copenhagen", "Finland"=>"Helsinki", "France" => "Paris", "Slovakia"=>"Bratislava", "Slovenia"=>"Ljubljana", "Germany" => "Berlin", "Greece" => "Athens", "Ireland"=>"Dublin", "Netherlands"=>"Amsterdam", "Portugal"=>"Lisbon", "Spain"=>"Madrid", "Sweden"=>"Stockholm", "United Kingdom"=>"London", "Cyprus"=>"Nicosia", "Lithuania"=>"Vilnius", "Czech Republic"=>"Prague", "Estonia"=>"Tallin", "Hungary"=>"Budapest", "Latvia"=>"Riga", "Malta"=>"Valetta", "Austria" => "Vienna", "Poland"=>"Warsaw") ;

Create a PHP script which displays the capital and country name from the above array $ceu. Sort the list by the capital of the country.

*Sample Output :*  
The capital of Netherlands is Amsterdam   
The capital of Greece is Athens   
The capital of Germany is Berlin   
- - - - - - - - - - - - - - - - - - - - - - - - -   
- - - - - - - - - - - - - - - - - - - - - - - - -

<?php

$ceu = array( "Italy"=>"Rome", "Luxembourg"=>"Luxembourg",

"Belgium"=> "Brussels", "Denmark"=>"Copenhagen",

"Finland"=>"Helsinki", "France" => "Paris",

"Slovakia"=>"Bratislava", "Slovenia"=>"Ljubljana",

"Germany" => "Berlin", "Greece" => "Athens",

"Ireland"=>"Dublin", "Netherlands"=>"Amsterdam",

"Portugal"=>"Lisbon", "Spain"=>"Madrid",

"Sweden"=>"Stockholm", "United Kingdom"=>"London",

"Cyprus"=>"Nicosia", "Lithuania"=>"Vilnius",

"Czech Republic"=>"Prague", "Estonia"=>"Tallin",

"Hungary"=>"Budapest", "Latvia"=>"Riga","Malta"=>"Valetta",

"Austria" => "Vienna", "Poland"=>"Warsaw") ;

asort($ceu) ;

foreach($ceu as $country => $capital)

{

echo "The capital of $country is $capital"."\n" ;

}

?>

**4.**$x = array(1, 2, 3, 4, 5);  
Delete an element from the above PHP array. After deleting the element, integer keys must be normalized.   
*Sample Output :*   
array(5) { [0]=> int(1) [1]=> int(2) [2]=> int(3) [3]=> int(4) [4]=> int(5) }   
array(4) { [0]=> int(1) [1]=> int(2) [2]=> int(3) [3]=> int(5) }

<?php

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

var\_dump($x);

unset($x[3]);

$x = array\_values($x);

echo '';

var\_dump($x);

?>

**5.** $color = array(4 => 'white', 6 => 'green', 11=> 'red');  
Write a PHP script to get the first element of the above array.   
*Expected result :* white

<?php

$color = array(4 => 'white', 6 => 'green', 11=> 'red');

echo reset($color)."\n";

?>

**6.** Write a PHP script which decodes the following JSON string.   
*Sample JSON code :*  
{"Title": "The Cuckoos Calling",  
"Author": "Robert Galbraith",  
"Detail": {  
"Publisher": "Little Brown"  
}}  
*Expected Output :*  
Title : The Cuckoos Calling  
Author : Robert Galbraith  
Publisher : Little Brown

<?php

function w3rfunction($value,$key)

{

echo "$key : $value"."\n";

}

$a = '{"Title": "The Cuckoos Calling",

"Author": "Robert Galbraith",

"Detail":

{

"Publisher": "Little Brown"

}

}';

$j1 = json\_decode($a,true);

array\_walk\_recursive($j1,"w3rfunction");

?>

**7.** Write a PHP script that inserts a new item in an array in any position.   
*Expected Output :*  
Original array :   
1 2 3 4 5   
After inserting '$' the array is :  
1 2 3 $ 4 5

<?php

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

echo 'Original array : '."\n";

foreach ($original as $x)

{echo "$x ";}

$inserted = '$';

array\_splice( $original, 3, 0, $inserted );

echo " \n After inserting '$' the array is : "."\n";

foreach ($original as $x)

{echo "$x ";}

echo "\n"

?>

**8.** Write a PHP script to sort the following associative array :   
array("Sophia"=>"31","Jacob"=>"41","William"=>"39","Ramesh"=>"40") in   
a) ascending order sort by value  
b) ascending order sort by Key  
c) descending order sorting by Value  
d) descending order sorting by Key

<?php

echo "Associative array : Ascending order sort by value";

$array2=array("Sophia"=>"31","Jacob"=>"41","William"=>"39","Ramesh"=>"40"); asort($array2);

foreach($array2 as $y=>$y\_value)

{

echo "Age of ".$y." is : ".$y\_value."

";

}

echo "Associative array : Ascending order sort by Key";

$array3=array("Sophia"=>"31","Jacob"=>"41","William"=>"39","Ramesh"=>"40"); ksort($array3);

foreach($array3 as $y=>$y\_value)

{

echo "Age of ".$y." is : ".$y\_value."

";

}

echo "Associative array : Descending order sorting by Value";

$age=array("Sophia"=>"31","Jacob"=>"41","William"=>"39","Ramesh"=>"40");

arsort($age);

foreach($age as $y=>$y\_value)

{

echo "Age of ".$y." is : ".$y\_value."

";

}

echo "Associative array : Descending order sorting by Key";

$array4=array("Sophia"=>"31","Jacob"=>"41","William"=>"39","Ramesh"=>"40"); krsort($array4);

foreach($array4 as $y=>$y\_value)

{

echo "Age of ".$y." is : ".$y\_value."

";

}

?>

**9.** Write a PHP script to calculate and display average temperature, five lowest and highest temperatures.   
Recorded temperatures : 78, 60, 62, 68, 71, 68, 73, 85, 66, 64, 76, 63, 75, 76, 73, 68, 62, 73, 72, 65, 74, 62, 62, 65, 64, 68, 73, 75, 79, 73  
*Expected Output :*  
Average Temperature is : 70.6   
List of seven lowest temperatures : 60, 62, 63, 63, 64,   
List of seven highest temperatures : 76, 78, 79, 81, 85,

<?php

$month\_temp = "78, 60, 62, 68, 71, 68, 73, 85, 66, 64, 76, 63, 81, 76, 73,

68, 72, 73, 75, 65, 74, 63, 67, 65, 64, 68, 73, 75, 79, 73";

$temp\_array = explode(',', $month\_temp);

$tot\_temp = 0;

$temp\_array\_length = count($temp\_array);

foreach($temp\_array as $temp)

{

$tot\_temp += $temp;

}

$avg\_high\_temp = $tot\_temp/$temp\_array\_length;

echo "Average Temperature is : ".$avg\_high\_temp."";

sort($temp\_array);

echo " List of seven lowest temperatures :";

for ($i=0; $i< 5; $i++)

{

echo $temp\_array[$i].", ";

}

echo "List of seven highest temperatures :";

for ($i=($temp\_array\_length-5); $i< ($temp\_array\_length); $i++)

{

echo $temp\_array[$i].", ";

}

?>

**10.** Write a PHP program to sort an array of positive integers using the Bead-Sort Algorithm.   
According to Wikipedia "Bead-sort is a natural sorting algorithm, developed by Joshua J. Arulanandham, Cristian S. Calude and Michael J. Dinneen in 2002. Both digital and analog hardware implementations of bead sort can achieve a sorting time of O(n); however, the implementation of this algorithm tends to be significantly slower in software and can only be used to sort lists of positive integers".  
  
*Input array* : Array ( [0] => 5 [1] => 3 [2] => 1 [3] => 3 [4] => 8 [5] => 7 [6] => 4 [7] => 1 [8] => 1 [9] => 3 )   
*Expected Result* : Array ( [0] => 8 [1] => 7 [2] => 5 [3] => 4 [4] => 3 [5] => 3 [6] => 3 [7] => 1 [8] => 1 [9] => 1 )

**11.** Write a PHP program to merge (by index) the following two arrays.   
*Sample arrays* :   
$array1 = array(array(77, 87), array(23, 45));  
$array2 = array("w3resource", "com");   
*Expected Output* :

Array

(

[0] => Array

(

[0] => w3resource

[1] => 77

[2] => 87

)

[1] => Array

(

[0] => com

[1] => 23

[2] => 45

)

)

**12.** Write 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 )

**13.** Write a PHP script which displays all the numbers between 200 and 250 that are divisible by 4.   
Note : Do not use any PHP control statement.   
*Expected Output* : 200,204,208,212,216,220,224,228,232,236,240,244,248

**14.** Write a PHP script to get the shortest/longest string length from an array.   
*Sample arrays* : ("abcd","abc","de","hjjj","g","wer")  
*Expected Output* : The shortest array length is 1. The longest array length is 4.

**15.** Write a PHP script to generate unique random numbers within a range.   
*Sample Range*: (11, 20)   
*Sample Output* : 17 16 13 20 14 19 18 15 11 12

**16.** Write a PHP script to get the largest key in an array. 

**17.** Write a PHP function that returns the lowest integer that is not 0. 

**18.** Write a PHP function to floor decimal numbers with precision.   
Note: Accept three parameters number, precision, and $separator  
*Sample Data* :   
1.155, 2, "."  
100.25781, 4, "."  
-2.9636, 3, "."  
 *Expected Output* :   
1.15   
100.2578   
-2.964

**19.** Write a PHP script to print "second" and Red from the following array.   
*Sample Data* :   
$color = array ( "color" => array ( "a" => "Red", "b" => "Green", "c" => "White"),  
"numbers" => array ( 1, 2, 3, 4, 5, 6 ),  
"holes" => array ( "First", 5 => "Second", "Third"));

**20.** Write a PHP function to sort an array according to another array acting as a priority list. 

**21.** Write a PHP function to sort subnets. 

**22.** Write a PHP script to sort the following array by the day (page\_id) and username. 

**23.** Write a PHP program to sort a multi-dimensional array set by a specific key. 

**24.** Write a PHP script to sort an array using case-insensitive natural ordering. 

**25.** Write a PHP function to sort entity letters.

**26.** Write a PHP function to shuffle an associative array, preserving key, value pairs.

**27.** Write a PHP function to generate a random password (contains uppercase, lowercase, numeric and other) using shuffle() function.

**28.** Write a PHP script to sort an array in reverse order (highest to lowest).

**29.** Write a PHP program to generate an array with a range taken from a string.

**30.** Write a PHP program to create a letter range with arbitrary length.

**31.** Write a PHP program to get the index of the highest value in an associative array.

**32.** Write a PHP program to get the extension of a file.

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

**34.** Write a PHP program to sort an associative array (alphanumeric with case-sensitive data) by values.

**35.** Write a PHP script to trim all the elements in an array using array\_walk function.

**36.** Write a PHP script to lower-case and upper-case, all elements in an array.

**37.** Write a PHP script to count the total number of times a specific value appears in an array.

**38.** Write a PHP function to create a multidimensional unique array for any single key index.

**39.** Write a PHP program to remove duplicate values from an array which contains only strings or only integers.

**40.** Write a PHP program to get a sorted array without preserving the keys.

**41.** Write a PHP program to identify the email addresses which are not unique.

**42.** Write a PHP function to find unique values from multidimensional arrays and flatten them in zero depth.

**43.** Write a PHP script to merge two commas separated lists with unique value only.

**44.** Write a PHP a function to remove a specified, duplicate entry from an array.

**45.** Write a PHP script to do a multi-dimensional difference, i.e. returns values of the first array that are not in the second array.   
Note : Use array\_udiff() function. 

**46.** Write a PHP function to check whether all array values are strings or not.

**47.** Write a PHP function to get an array with the first key and value.

**48.** Write a PHP function to set union of two arrays.

**49.** Write a PHP script to get an array containing all the entries of an array which have the keys that are present in another array.  
*Test Data :* 1st array : ('c1' => 'Red', 'c2' => 'Green', 'c3' => 'White', c4 => 'Black')  
2nd array : ('c2', 'c4')  
*Output :*  
Array   
(   
[c2] => Green  
[c4] => Black  
)

**50.** Write a PHP script to get the last value of an array without affecting the pointer.

**51.** Write a PHP program to filter out some elements with certain key-names.  
*Test Data :*  
1st array : ('c1' => 'Red', 'c2' => 'Green', 'c3' => 'White', c4 => 'Black')  
2nd array : ('c2', 'c4')  
*Output :*  
Array  
(   
[c1] => Red   
[c3] => White   
)

**52.** Write a PHP function to filter a multi-dimensional array. The function will return those items that will match with the specified value.

**53.** Write a PHP script to delete a specific value from an array using array\_filter() function.

**54.** Write a PHP script to remove all white spaces in an array.

**55.** Write a PHP function to convert a string to an array (trimming every line and remove empty lines).

**56.** Write a PHP script to create a two-dimensional array (4x4), initialized to 10.

**57.** Write a PHP function to compares two multidimensional arrays and returns the difference.

**58.** Write a PHP script to combine (using one array for keys and another for its values) the following two arrays.  
('x', 'y', 'y'), (10, 20, 30)

**59.** Write a PHP program to create a range like the following array.  
Array   
(  
[20] => 2  
[21] => 3  
[22] => 4  
[23] => 5  
[24] => 6  
[25] => 7  
)