

## Web Technology Practical Assignment

### SET 1.

1) Write a menu driven program to perform the following operations on an associative array:

- a) Display the elements of an array along with the keys.
- b) Display the size of an array

Ans :

```
<?php
$array = array(
    "one" => 1,
    "two" => 2,
    "three" => 3,
    "four" => 4,
    "five" => 5
);

echo "Menu:\n";
echo "1. Display the elements of an array along with the keys.\n";
echo "2. Display the size of an array.\n";
echo "Enter your choice: ";
$choice = (int)readline();

switch ($choice) {
    case 1:
        foreach ($array as $key => $value) {
            echo "$key: $value\n";
        }
        break;
    case 2:
        $size = count($array);
        echo "Size of the array: $size\n";
        break;
    default:
        echo "Invalid choice.\n";
}
?>
```

2) Write a menu driven program the following operation on an associative array

- a) Reverse the order of each element's key-value pair. [Hint: array\_flip()]
- b) Traverse the element in an array in random order. [Hint: shuffle()]

Ans :

```

<?php
$array = array(
    "one" => 1,
    "two" => 2,
    "three" => 3,
    "four" => 4,
    "five" => 5
);

echo "Menu:\n";
echo "1. Reverse the order of each element's key-value pair.\n";
echo "2. Traverse the element in an array in random order.\n";
echo "Enter your choice: ";
$choice = (int)readline();

switch ($choice) {
    case 1:
        $reversed_array = array_flip($array);
        print_r($reversed_array);
        break;
    case 2:
        $keys = array_keys($array);
        shuffle($keys);
        foreach ($keys as $key) {
            echo "$key: $array[$key]\n";
        }
        break;
    default:
        echo "Invalid choice.\n";
}
?>

```

3) Declare array. Reverse the order of elements, making the first element last and last element first and similarly rearranging other array elements.[Hint : array\_reverse()]

Ans :

```

<?php
$array = array(1, 2, 3, 4, 5);
$reversed_array = array_reverse($array);
print_r($reversed_array);
?>

```

SET B,

1. Declare a Multidimensional Array. Display specific element from a Multidimensional array. Also delete given element from the Multidimensional array.(After each operation display array content).

Ans:

```
<?php
$array = array(1, 2, 3, 4, 5);

echo "Menu:\n";
echo "1. Display the elements of the array.\n";
echo "2. Display the size of the array.\n";
echo "3. Reverse the order of the elements in the array.\n";
echo "Enter your choice: ";
$choice = (int)readline();

switch ($choice) {
    case 1:
        foreach ($array as $value) {
            echo "$value\n"; }
        break;
    case 2:
        $size = count($array);
        echo "Size of the array: $size\n";
        break;
    case 3:
        $reversed_array = array_reverse($array);
        print_r($reversed_array);
        break;
    default:
        echo "Invalid choice.\n";
}
?>
```

2. Write a menu driven program to perform the following stack related operations.

a) Insert an element in stack.

b) Delete an element from stack.[Hint: array\_push(), array\_pop()]

Ans :

```
<?php
$stack = array();
```

```

echo "Menu:\n";
echo "1. Insert an element in stack.\n";
echo "2. Delete an element from stack.\n";
echo "Enter your choice: ";
$choice = (int)readline();

switch ($choice) {
    case 1:
        echo "Enter the element to insert: ";
        $element = (int)readline();
        array_push($stack, $element);
        echo "Element inserted.\n";
        break;
    case 2:
        if (count($stack) > 0) {
            $element = array_pop($stack);
            echo "Deleted element: $element\n";
        } else {
            echo "Stack is empty.\n";
        }
        break;
    default:
        echo "Invalid choice.\n";
}
?>

```

3. Write a menu driven program to perform the following queue related operations

- a) Insert an element in queue
- b) Delete an element from queue
- c) Display the contents of queue

Ans :

```

<?php
$queue = array();

echo "Menu:\n";
echo "1. Insert an element in queue.\n";
echo "2. Delete an element from queue.\n";
echo "3. Display the contents of queue.\n";
echo "Enter your choice: ";
$choice = (int)readline();

switch ($choice) {
    case 1:
        echo "Enter the element to insert: ";

```

```

        $element = (int)readline();
        array_push($queue, $element);
        echo "Element inserted.\n";
        break;
    case 2:
        if (count($queue) > 0) {
            $element = array_shift($queue);
            echo "Deleted element: $element\n";
        } else {
            echo "Queue is empty.\n";
        }
        break;
    case 3:
        echo "Contents of queue:\n";
        foreach ($queue as $value) {
            echo "$value\n";
        }
        break;
    default:
        echo "Invalid choice.\n";
}
?>

```

SET C ,

1. Write a menu driven program to perform the following operations on associative arrays:

- a) Merge the given arrays.
- b) Find the intersection of two arrays.
- c) Find the union of two arrays.
- d) Find set difference of two arrays.

Ans :

```

<?php
$array1 = array(
    "one" => 1,
    "two" => 2,
    "three" => 3
);
$array2 = array(
    "three" => 3,
    "four" => 4,

```

```

    "five" => 5
);

echo "Menu:\n";
echo "1. Merge the given arrays.\n";
echo "2. Find the intersection of two arrays.\n";
echo "3. Find the union of two arrays.\n";
echo "4. Find set difference of two arrays.\n";
echo "Enter your choice: ";
$choice = (int)readline();
switch ($choice) {
    case 1:
        $merged_array = array_merge($array1, $array2);
        print_r($merged_array);
        break;
    case 2:
        $intersection = array_intersect($array1, $array2);
        print_r($intersection);
        break;
    case 3:
        $union = $array1 + $array2;
        print_r($union);
        break;
    case 4:
        $difference = array_diff($array1, $array2);
        print_r($difference);
        break;
    default:
        echo "Invalid choice.\n";
}
?>

```

2. Write a menu driven program to perform the following operations on associative arrays:

- a) Sort the array by values (changing the keys) in ascending, descending order.
- b) Also sort the array by values without changing the keys.
- c) Filter the odd elements from an array.

Ans :

```

<?php
$array = array(
    "one" => 1,
    "two" => 2,

```

```

    "three" => 3,
    "four" => 4,
    "five" => 5
);

echo "Menu:\n";
echo "1. Sort the array by values (changing the keys) in ascending order.\n";
echo "2. Sort the array by values (changing the keys) in descending order.\n";
echo "3. Sort the array by values without changing the keys.\n";
echo "4. Filter the odd elements from an array.\n";
echo "Enter your choice: ";
$choice = (int)readline();
switch ($choice) {
    case 1:
        asort($array);
        print_r($array);
        break;
    case 2:
        arsort($array);
        print_r($array);
        break;
    case 3:
        $sorted_array = $array;
        sort($sorted_array);
        print_r($sorted_array);
        break;
    case 4:
        $filtered_array = array_filter($array, function ($value) {
            return $value % 2 == 1;
        });
        print_r($filtered_array);
        break;
    default:
        echo "Invalid choice.\n";
}
?>

```

3. Sort the different arrays at a glance using single function.

Ans :

```

<?php
function sort_arrays(&$arrays) {
    foreach ($arrays as &$array) {
        sort($array);
    }
}

```

```
    }  
}  
  
$arrays = array(  
    array(3, 2, 1),  
    array(6, 5, 4),  
    array(9, 8, 7)  
);  
  
sort_arrays($arrays);  
  
foreach ($arrays as $array) {  
    print_r($array);  
}  
?>
```