

PSG COLLEGE OF TECHNOLOGY, COIMBATORE – 641 004
Department of Applied Mathematics and Computational Sciences
MSc SOFTWARE SYSTEMS – Semester IV
20XW48 – Web Designing Lab
PROBLEM SHEET – PHP & MySQL

1. Write a PHP script to get the PHP version and configuration information.

```
<?php
phpinfo();
?>
```

```
phpinfo() PHP Version => 7.1.1-1 System => Linux checker 4.9.0-14-amd64 #1 SMP Debian 4.9.246-2
(2020-12-17) x86_64 Build Date => Jan 25 2017 14:31:59 Server API => Command Line Interface Virtual
Directory Support => disabled Configuration File (php.ini) Path => /etc/php/7.1/cli Loaded Configuration File =>
/etc/php/7.1/cli/php.ini Scan this dir for additional .ini files => /etc/php/7.1/cli/conf.d Additional .ini files parsed
=> /etc/php/7.1/cli/conf.d/10-opcache.ini, /etc/php/7.1/cli/conf.d/10-pdo.ini, /etc/php/7.1/cli/conf.d/20-calendar.ini,
/etc/php/7.1/cli/conf.d/20-ctype.ini, /etc/php/7.1/cli/conf.d/20-curl.ini, /etc/php/7.1/cli/conf.d/20-exif.ini, /etc/php
/7.1/cli/conf.d/20-fileinfo.ini, /etc/php/7.1/cli/conf.d/20-ftp.ini, /etc/php/7.1/cli/conf.d/20-gettext.ini, /etc/php
/7.1/cli/conf.d/20-gmp.ini, /etc/php/7.1/cli/conf.d/20-iconv.ini, /etc/php/7.1/cli/conf.d/20-json.ini, /etc/php/7.1/cli
/conf.d/20-pdo_sqlite.ini, /etc/php/7.1/cli/conf.d/20-phar.ini, /etc/php/7.1/cli/conf.d/20-posix.ini, /etc/php/7.1/cli
/conf.d/20-readline.ini, /etc/php/7.1/cli/conf.d/20-shmop.ini, /etc/php/7.1/cli/conf.d/20-sockets.ini, /etc/php/7.1/cli
/conf.d/20-sqlite3.ini, /etc/php/7.1/cli/conf.d/20-sysvmsg.ini, /etc/php/7.1/cli/conf.d/20-sysvsem.ini, /etc/php
/7.1/cli/conf.d/20-sysvshm.ini, /etc/php/7.1/cli/conf.d/20-tidy.ini, /etc/php/7.1/cli/conf.d/20-tokenizer.ini PHP API
=> 20160303 PHP Extension => 20160303 Zend Extension => 320160303 Zend Extension Build =>
API320160303,NTS PHP Extension Build => API20160303,NTS Debug Build => no Thread Safety => disabled
Zend Signal Handling => enabled Zend Memory Manager => enabled Zend Multibyte Support => disabled IPv6
Support => enabled DTrace Support => enabled Registered PHP Streams => https, ftps, compress.zlib, php, file,
glob, data, http, ftp, phar Registered Stream Socket Transports => tcp, udp, unix, udg, ssl, tls, tlsv1.0, tlsv1.1,
tlsv1.2 Registered Stream Filters => zlib.*, string.rot13, string.toupper, string.tolower, string.strip_tags, convert.*,
consumed, dechunk, convert.iconv.* This program makes use of the Zend Scripting Language Engine: Zend
Engine v3.1.0, Copyright (c) 1998-2017 Zend Technologies with Zend OPcache v7.1.1-1, Copyright (c)
1999-2017, by Zend Technologies
```

```
Configuration calendar
Calendar support => enabled Core PHP Version => 7.1.1-1 Directive => Local Value => Master Value
allow_url_fopen => On => On allow_url_include => Off => Off arg_separator.input => & => &
arg_separator.output => & => & auto_append_file => no value => no value auto_globals_jit => On => On
auto_prepend_file => no value => no value browscap => no value => no value default_charset => UTF-8 =>
UTF-8 default_mimetype => text/html => text/html disable_classes => no value => no value disable_functions =>
no value => no value display_errors => Off => Off display_startup_errors => Off => Off doc_root => no value =>
no value docref_ext => no value => no value docref_root => no value => no value enable_dl => Off => Off
enable_post_data_reading => On => On error_append_string => no value => no value error_log => no value =>
no value error_prepend_string => no value => no value error_reporting => 22527 => 22527 expose_php => On
=> On extension_dir => /usr/lib/php/20160303 => /usr/lib/php/20160303 file_uploads => On => On hard_timeout
=> 2 => 2 highlight.comment => #FF8000 => #FF8000 highlight.default => #0000BB => #0000BB
```

2. Create a PHP script that displays 1-2-3-4-5-6-7-8-9-10 on one line. There will be no hyphen(-) at starting and ending position.

```
<?php
for ($i=1;$i<=10;$i++)
{
```

```

if($i==10)
    echo $i;
else
    echo $i."-";
}
?>

```

1-2-3-4-5-6-7-8-9-10

3. Write a PHP program to keep track of the number of visitors visiting the web page and display the count of visitors with proper headings.

```

<?php
session_start();
if(isset($_SESSION['views']))
    $_SESSION['views'] = $_SESSION['views']+1;
else
    $_SESSION['views']=1;
echo"views = ".$_SESSION['views'];
?>

```

views = 1

4. Write a PHP program to display a digital clock which displays the current time of the server.

```

<?php
date_default_timezone_set("Asia/Kolkata");
echo date('h:i:s A');
?>

```

02:03:02 PM

5. Write a PHP script to calculate and display average temperature, five lowest and highest temperatures using functions.

```

<?php
$temperature = array(97,75,96,89,88,93,100,79,98,88);

function listvalues($value)
{
    for($i = 0; $i < count($value); $i++)
    {
        echo "$value[$i] ";
    }
    echo "<br>";
}

```

```
function printAverage($array)
{
    $total = 0;
    foreach($array as $element)
    {
        $total += $element;
    }
    echo "Average:";
    echo number_format($total / count($array), 1);
    echo "<br>";
}
```

```
sort($temperature);
listvalues($temperature);
printAverage($temperature);
echo "List of five lowest temperatures : ";
    for($i = 0; $i < 5; $i++)
    {
        echo "$temperature[$i], ";
    }
    echo "<br>";
```

```
echo "List of five highest temperatures : ";
    for($i = count($temperature) - 5; $i <= count($temperature) - 1; $i++)
    {
        echo "$temperature[$i], ";
    }

?>
```

75 79 88 88 89 93 96 97 98 100

Average:90.3

List of five lowest temperatures : 75, 79, 88, 88, 89,

List of five highest temperatures : 93, 96, 97, 98, 100,

6. Write a PHP script that removes the whitespaces from a string.

Sample String: 'The quick " " brown fox'

Expected Output: Thequick""brownfox

```
<?php
```

```
$str = "Hello World, it's PHP!";
```

```
echo str_replace(" ", "", $str)
```

```
?>
```

```
HelloWorld,it'sPHP.
```

7. Write a PHP function that checks whether a passed string is palindrome or not? A palindrome is word, phrase, or sequence that reads the same backward as forward, e.g., madam or nurses run.

```
<?php
$str = "MADAM";
if ($str==strrev($str))
    echo "Palindrome";
else
    echo "NO.";
?>
```

Palindrome

8. Write a PHP script to sort the following associative array using functions:
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
$person = array("Sophia"=>"31","Jacob"=>"41","William"=>"39","Ramesh"=>"40");
function sortByValue($p){
    asort($p);
    echo "Ascending Sort by value<br>";
    foreach ($p as $x=>$xV)
        {echo "Key=" . $x . ", Value=" . $xV;
        echo "<br>";}
    echo "<br>";

    arsort($p);
    echo "Descending Sort by value<br>";
    foreach ($p as $x=>$xV)
        {echo "Key=" . $x . ", Value=" . $xV;
        echo "<br>";}
    echo "<br>";
}

function sortByKey($p){
```

Ascending Sort by value
Key=Sophia, Value=31
Key=William, Value=39
Key=Ramesh, Value=40
Key=Jacob, Value=41

Descending Sort by value
Key=Jacob, Value=41
Key=Ramesh, Value=40
Key=William, Value=39
Key=Sophia, Value=31

Ascending Sort by key
Key=Jacob, Value=41
Key=Ramesh, Value=40
Key=Sophia, Value=31
Key=William, Value=39

Descending Sort by key
Key=William, Value=39
Key=Sophia, Value=31
Key=Ramesh, Value=40
Key=Jacob, Value=41

```

ksort($p);
echo "Ascending Sort by value<br>";
foreach ($p as $x=>$xV)
{echo "Key=" . $x . ", Value=" . $xV;
echo "<br>";}
echo "<br>";

krsort($p);
echo "Descending Sort by value<br>";
foreach ($p as $x=>$xV)
{echo "Key=" . $x . ", Value=" . $xV;
echo "<br>";}
echo "<br>";
}
sortByValue($person);
sortByKey($person);
?>

```

9. Write a PHP script to store and retrieve persistent data across a client session.

```

<?php
session_start();
$_SESSION['luser_name'] = 'Mac';
echo $_SESSION['luser_name'];
?>

```



10. Write a PHP script to simulate the Shopping Cart, allow users to select items from a catalog and save them for later access using session.

```

<?php
session_start();
$products = [
    1 => ["name" => "Apple", "price" => 1.23],
    2 => ["name" => "Mango", "price" => 2.34],
    3 => ["name" => "Orange", "price" => 3.45],
    4 => ["name" => "Pear", "price" => 4.32]
];
foreach ($products as $pid=>$p){ ?>
    <div class="pCell">

```

```

    Apple
    $1.23
    Add To Cart
  }
  Mango
  $2.34
  Add To Cart
  }
  Orange
  $3.45
  Add To Cart
  }
  Pear
  $4.32

```

```

<div class="pTxt">
<div class="pName"><?=$p["name"]?></div>
<div class="pPrice">$<?=$p["price"]?></div>
</div>
<button class="pAdd" onclick="cart.add(<?=$pid?>)">
Add To Cart
</button>
</div>}
<?php } ?>

```

```

<?php
if (!isset($_SESSION["cart"])) { $_SESSION["cart"] = []; }
function respond ($status=1, $msg="") {
    exit(json_encode(["status"=>$status, "msg"=>$msg]));
}
if (isset($_POST["req"])) { switch ($_POST["req"]) {
    default: respond(0, "Invalid Request");
    case "add":
        $qty = &$_SESSION["cart"][$_POST["pid"]];
        if (isset($qty)) { $qty++; } else { $qty = 1; }
        if ($qty > 99) { $qty = 99; }
        respond();
    case "set":
        $qty = &$_SESSION["cart"][$_POST["pid"]];
        $qty = $_POST["qty"];
        if ($qty > 99) { $qty = 99; }
        if ($qty <= 0) { unset($_SESSION["cart"][$_POST["pid"]]); }
        respond();}}
?>

```

11. Write a PHP script to store and retrieve cookies in a web page.

```

<?php
$cookie_name = "user";
$cookie_value = "Mac";
setcookie($cookie_name, $cookie_value, time() + (86400 * 30), "/"); // 86400 = 1 day
?>

<html>
<body>

```

```

<?php
if(!isset($_COOKIE[$cookie_name])) {
    echo "Cookie named '" . $cookie_name . "' is not set!";
} else {
    echo "Cookie '" . $cookie_name . "' is set!<br>";
    echo "Value is: " . $_COOKIE[$cookie_name];
}
?>

```

Cookie named 'user' is not set!

12. Write a PHP script named `states.php` that creates a variable `$states` with the value *"Mississippi Texas Massachusetts Kansas"*. The script should perform the following tasks:

- a) Search for a word in `$states` that ends in *xas*. Store this word in element 0 of an array named `$statesArray`.
- b) Search for a word in `$states` that begins with *k* and ends in *s*. Perform a case-insensitive comparison. Store this word in element 1 of `$statesArray`.
- c) Search for a word in `$states` that begins with *M* and ends in *s*. Store this element in element 2 of the array.
- d) Search for a word in `$states` that ends in *a*. Store this word in element 3 of the array.
- e) Search for a word in `$states` at the beginning of the string that starts with *M*. Store this word in element 4 of the array.
- f) Output the array `$statesArray` to the screen.

```

<?php
$states='Mississippi Texas Massachusetts Kansas';
$states=explode(' ', $states);

```

```
$statesArray=array();
```

Result

Array ([4] => Massachusetts [0] => Texas [2] => Massachusetts)

```

foreach ($states as $state){
    $length=strlen($state);
    if(substr($state,$length-3,3)=='xas')$statesArray[0]=$state;
    if((substr($state,0,1)=='k') and (substr($state,$length-1,1)=='s'))$statesArray[1]=$state;
    if((substr($state,0,1)=='M') and (substr($state,$length-1,1)=='s'))$statesArray[2]=$state;
    if((substr($state,$length-1,1)=='a'))$statesArray[3]=$state;
    if((substr($state,0,1)=='M'))$statesArray[4]=$state;
}

```

```

echo "Result <br>";
print_r($statesArray);
?>

```

13. Write a PHP script that tests whether an e-mail address is input correctly. Verify that the input begins with a series of characters, followed by the @ character, another series of characters, a period (.) and a final series of characters. Test your program, using both valid and invalid e-mail addresses.

```

<?php
$email1= "gigi@outlook.com";
$email2 = "btsnottoday.kr";
if (filter_var($email2, FILTER_VALIDATE_EMAIL))
    echo "' . $email2 . "' = Valid!";
else
    echo "' . $email2 . "' = Invalid!";
echo "<br>";
if (filter_var($email1, FILTER_VALIDATE_EMAIL))
    echo "' . $email1 . "' = Valid!";
else
    echo "' . $email1 . "' = Invalid!";
?>

```

"btsnottoday.kr" = Invalid
"gigi@outlook.com" = Valid

14. Write a PHP program to sort the student records which are stored in the database using selection sort.

```

<?php
$servername = "10.1.67.167";
$username = "msc20pw19";
$password = "msc20pw";
$db = msc20pw19;

// Create connection
$conn = mysqli_connect($servername, $username, $password,$db);

// Check connection
if (!$conn) {
    die("Connection failed: " . mysqli_connect_error());
}

```



```
//TABLE CREATION
```

```
$sql = "CREATE TABLE student(id INT(6) PRIMARY KEY,name VARCHAR(30) NOT NULL)";
```

```
if(mysqli_query($conn,$sql)){  
    echo 'table created successfully';  
}
```

```
//TABLE INSERTION
```

```
$quer1 = "INSERT INTO student values(1,'abc');";  
$quer1 .= "INSERT INTO student values(2,'xyz');";  
$quer1 .= "INSERT INTO student values(3,'pqr');";  
$quer1 .= "INSERT INTO student values(4,'rst');";  
$quer1 .= "INSERT INTO student values(5,'uvw');";  
$quer1 .= "INSERT INTO student values(6,'xyz');";
```

```
if(mysqli_multi_query($conn,$quer1)){  
    echo 'inserted successfully';  
}
```

```
$sql="SELECT name FROM student;";  
$arr = mysqli_query($conn,$sql);  
if(mysqli_num_rows($arr)>0){  
    $names = array();  
    while($row = mysqli_fetch_assoc($arr)){  
        array_push($names,$row['name']);  
    }  
    echo 'Student Records fetched from database:<br>';  
    echo implode(' ', $names);  
  
    echo "<br>Students record after selection sort:<br>";  
    echo implode(' ', selection_sort($names));  
}else{  
    echo 'no data found';
```

```

}
function selection_sort($data)
{
for($i=0; $i<count($data)-1; $i++) {
    $min = $i;
    for($j=$i+1; $j<count($data); $j++) {
        if(strcmp($data[$j], $data[$min]) < 0){
            $min = $j;
        }
    }
    $data = swap_positions($data, $i, $min);
}
return $data;
}
function swap_positions($data1, $left, $right) {
    $backup_old_data_right_value = $data1[$right];
    $data1[$right] = $data1[$left];
    $data1[$left] = $backup_old_data_right_value;
    return $data1;
}
?>

```

15. Write a PHP script that obtains a URL and its description from a user and stores the information into a database using MySQL. Create and run a SQL script with a database named *URL* and a table named *URLTable*. The first field of the table should contain an actual *URL*, and the second, which is named Description, should contain a description of the *URL*. Use *www.deitel.com* as the first *URL*, and input *'Cool site!'* as its description. The second *URL* should be *www.php.net*, and the description should be *'The official PHP'* site. After each new *URL* is submitted, print the contents of the database in a table.

```

<?php
    $conn = mysqli_connect('localhost', 'admin', 'admin', 'test');
    if(!$conn){
        die('Connection failed: '.mysqli_connect_error());
    }

    $sql = 'create table URLTable(
        url varchar(400) primary key, description varchar(100) );';

```

```

        if(!mysqli_query($conn, $sql)){
            die('Error: ' .mysqli_error($conn));
        }
        echo 'Table created successfully';

        if(isset($_POST['submit'])){
            $url = $_POST['url'];
            $desc = $_POST['desc'];

            $sql = 'insert into urltable values("'.$url.'", "'.$desc.'");';
            if(!mysqli_query($conn, $sql)){
                die('Error: ' .mysqli_error($conn));
            }
        }
    }
?>

<html>
<head>
    <title>15</title>
</head>
<body>
    <form action="15.php" method="post">
        <label for="">URL</label>
        <input type="url" name="url" ><br>
        <label for="">Description</label>
        <input type="text" name="desc" id=""><br>
        <input type="submit" name="submit">
    </form>

    <table>
        <tr>
            <th>URL</th>
            <th>Description</th>
        </tr>

        <?php
            $result = mysqli_query($conn, 'select * from urltable;');

```

```

        if(mysqli_num_rows($result)>0){
            while($row = mysqli_fetch_assoc($result)){
                $p = sprintf("<tr><td><a href='%s'>%s</a></td><td>%s</td></tr>", $row['url'],
                $row['url'], $row['description']);
                echo $p;
            }
        }
    ?>
</table>
</body>
</html>

```

16. Develop a PHP program to authenticate users with a valid *User Id* and *Password* before granting access to a protected resource.

```

<?php
$msg = "";
$conn = mysqli_connect('localhost', 'admin', 'admin', 'test');
if(!$conn){
    die('Connection failed: ' .mysqli_connect_error());
}

// //TABLE CREATION
$query = 'create table users (username varchar(20) primary key, password varchar(20));';
if(!mysqli_query($conn, $query)){
    die('Error: ' .mysqli_error($conn));
}
echo 'Table created successfully';

//TABLE INSERTION
$query = 'insert into users values("abc", "123");';
$query .= 'insert into users values("xyz", "456");';
$query .= 'insert into users values("pqr", "789");';
$query .= 'insert into users values("rst", "101");';
$query .= 'insert into users values("uvw", "102");';
$query .= 'insert into users values("xyz", "103");';
if(!mysqli_multi_query($conn, $query)){

```

```

        die('Error: ' .mysqli_error($conn));
    }
    echo 'Inserted successfully';
    if(isset($_POST['login'])){
        $username = $_POST['username'];
        $password = $_POST['password'];
        $query = "select * from users where username = '$username' and password = '$password'";
        $result = mysqli_query($conn, $query);
        if(mysqli_num_rows($result)>0){
            $msg = 'Password Found';
        }else{
            $msg = "Password not Found";
        }
    }
}

```

```

?>
<html>
<head>
    <title>16th</title>
    <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css"
rel="stylesheet"
integrity="sha384-EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65Vohhpuu
COMLASjC" crossorigin="anonymous">
</head>
<body>
    <div class = "container">
    </div>
    <div class = "container" style="width: 40%; margin: 40px auto">
    <form class = "" role = "form" action = "
<?php echo htmlspecialchars($_SERVER['PHP_SELF']);?>" method = "post">
        <div class="form-outline mb-4">
            <label class="form-label" for="username">User name</label>
            <input type="text" id="username" class="form-control" name="username" />
        </div>

```

```
<div class="form-outline mb-4">
  <label class="form-label" for="password">Password</label>
  <input type="password" id="password" class="form-control"
name="password"/>
</div>
  <button type="submit" class="btn btn-primary btn-block mb-4"
name="login">Sign in</button>
</form>
<h4 class = "form-signin-heading" style="text-align: center;">
<?php echo $msg; ?></h4>
</div>
</body>
</html>
```