Q . 1

1. <?php

function Palindrome($string){

    if (strrev($string) == $string){

        return 1;

    }

    else{

        return 0;

    }

}

$original = "DAD";

if(Palindrome($original)){

    echo "Palindrome";

}

else {

echo "Not a Palindrome";

}

?>

2. <?php

$num = 0;

$n1 = 0;

$n2 = 1;

echo "<h3>Fibonacci series for first 12 numbers: </h3>";

echo "\n";

echo $n1.' '.$n2.' ';

while ($num < 10 )

{

    $n3 = $n2 + $n1;

    echo $n3.' ';

    $n1 = $n2;

    $n2 = $n3;

    $num = $num + 1;

?>

3. <?php

$arr = array(100,40,1,5,17,50,4);

$max = $arr[0];

foreach($arr as $key => $val){

if($max < $val){

$max = $val;

}

}

print $max;

?>

4. <?php

$a=234;

$b=345;

$a=$a+$b;

$b=$a-$b;

$a=$a-$b;

echo "Value of a: $a</br>";

echo "Value of b: $b</br>";

?>

Q .2

1. 121

2. Error

Q .3

1. option b - bool(true)

2. option a – 2

3. option d – 5

Q . 4

A session creates a file in a temporary directory on the server where registered session variables and their session id are stored. This data will be available to all pages on the site amid that visit. Sessions generally store temporary data to allow multiple PHP pages to offer a complete functional transaction for the same user.

A cookie is a small file with the maximum size of 4KB that the web server stores on the client computer. Once a cookie has been set, all page requests that follow return the cookie name and value. A cookie can only be read from the domain that it has been issued from. A cookie is often used to identify a user. Each time the same computer requests a page with a browser, it will send the cookie too. With PHP, you can both create and retrieve cookie values.

Difference –

* Cookies are client-side files that contain user information, whereas Sessions are server-side files that contain user information.
* Cookie is not dependent on session, but Session is dependent on Cookie.
* Cookie expires depending on the lifetime you set for it, while a Session ends when a user closes his/her browser.
* The maximum cookie size is 4KB whereas in session, you can store as much data as you like.
* Cookie does not have a function named unsetcookie() while in Session you can use Session\_destroy(); which is used to destroy all registered data or to unset some

Q . 5

TRUNCATE

TRUNCATE removes all rows from a table. The operation cannot be rolled back and no triggers will be fired. As such, TRUNCATE is faster and doesn't use as much undo space as a DELETE. Table level lock will be added when Truncating.

DELETE

The DELETE command is used to remove rows from a table. A WHERE clause can be used to only remove some rows. If no WHERE condition is specified, all rows will be removed. After performing a DELETE operation you need to COMMIT or ROLLBACK the transaction to make the change permanent or to undo it. Note that this operation will cause all DELETE triggers on the table to fire. Row level lock will be added when deleting.

Q . 6

## mysql\_fetch\_assoc

The function returns an associative array of strings that corresponds to the fetched row, or FALSE if there are no more rows. The associativity array tells us about the key value pair, whereas the key tells about any column name and the value tells about the row value.

Here we can map the column name as key and value as row.

## mysql\_fetch\_array

This function name suggests that it returns an array. It fetches a result row as an associative array, a numeric array, or both. It has both numeric values as well as string values for a key.

Q . 7

There is no record

Q . 8

Steps to secure database are -

## Separate the Database and Web Servers

## Encrypt Stored Files

## Encrypt Your Backups Too

## Minimize Use of 3rd Party Apps

## Don't Use a Shared Server

## Enable Security Controls