Fibonacci

```
<?php
function fibonacci($n) {
    if ($n <= 0) {
        return 0;
    } elseif ($n == 1) {
        return 1;
    } else {
        return fibonacci($n - 1) + fibonacci($n - 2);
    }
}

// Get input from the user
$n = readline("Enter the number of terms: ");

// Calculate and display the Fibonacci series
for ($i = 0; $i < $n; $i++) {
    echo fibonacci($i) . " ";
}
?>
```

Prime no.

```
<?php
function isPrime($num) {
    if ($num <= 1) {
        return false;
    }
    if ($num <= 3) {
        return true;
    }
    if ($num % 2 == 0 || $num % 3 == 0) {
        return false;
    }
    $i = 5;
    while ($i * $i <= $num) {
        if ($num % $i == 0 || $num % ($i + 2) == 0) {
            return false;
        }
        $i += 6;
    }
    return true;
}

// Get input from the user
$num = readline("Enter a number: ");

// Check if the number is prime
if (isPrime($num)) {
    echo "$num is a prime number.";
} else {
    echo "$num is not a prime number.";
}
?>
```

HTML FILENAME: Greet.html

```
<!DOCTYPE html>
<html>
<head>
    <title>Greeting Form</title>
</head>
<body>
    <h2>Enter your name</h2>
    <form action="gre.php" method="POST">
```

```
        <label for="name">Name:</label>
        <input type="text" id="name" name="name" required>
        <input type="submit" value="Submit">
    </form>
</body>
</html>
```

PHP FILENAME: Gre.php

```
<?php
  if($_POST){
  $name=$_POST['name'];
  echo "Hello, $name";
}
?>
```

STRING/Number PALINDROME

AIM:

To write a program for find the given string is palindrome or not.

PROGRAM:

```
<?php
 if(isset($_POST['str'])){
            $name=$_POST['str'];
            $rev=strrev($name);
            if($name==$rev){
            echo "$name is palindrome";
            }
            else{
            echo "$name is not palindrome";
            }
 }
  ?>
 <form method="post">
 <input type="text" name="str">
  <input type="submit" name="submit">
 </form>
```

For armstrong

```
<?php
if(isset($_POST['name'])){
	$num=$_POST['name'];
	$c=$num;
	$a=0;
	while($c!=0){
	$r=$c%10;
	$a=$a+$r*$r*$r;
	$c=$c/10;
	}
	if($num==$a){
	echo "It is an armstrong";
	}else{
	echo "It is not an armstrong";
	}
}
?>
<form method="post">
 <input type="text" name="name">
  <input type="submit" name="submit">
 </form>
```

NESTING OF MEMBER FUNCTION

AIM:

To write a program for find the large value using nesting of member function.

```
<?php

function findLargest($a, $b) {
    return max($a, $b);
}
```

```
$a = readline("Enter the first number: ");
$b = readline("Enter the second number: ");


$largest = findLargest($a, $b);
echo "The largest number is: $largest\n";
```

MATHEMATICAL CALCULATOR

```
<?php
function calculate($num1, $num2, $operator) {
    switch ($operator) {
        case '+':
            return $num1 + $num2;
        case '-':
            return $num1 - $num2;
        case '*':
            return $num1 * $num2;
        case '/':
            if ($num2 == 0) {
                return "Cannot divide by zero";
            } else {
                return $num1 / $num2;
            }
        default:
            return "Invalid operator";
    }
}
$num1 = readline("Enter the first number: ");
$num2 = readline("Enter the second number: ");
$operator = readline("Enter the operator (+, -, *, /): ");

$result = calculate($num1, $num2, $operator);
echo "Result: $result\n";
?>
```

AGE CALCULATOR

```
<?php

function calculateAge($birthdate) {
    $birthDate = new DateTime($birthdate);
    $currentDate = new DateTime();
    $age = $currentDate->diff($birthDate)->y;
    return $age;
}
$birthdate = readline("Enter your birthdate (YYYY-MM-DD): ");
$age = calculateAge($birthdate);
echo "Your age is: $age years old";
?>
```

To write a php program to manipulate arrays.

To write a program for display our personal information using various tags.

```
// filename index.html
<!DOCTYPE html>
<html>
<head>
    <title>Personal Information</title>
</head>
<body>
    <h1>Personal Information</h1>
    <?php include 'personal_info.php'; ?>
</body>
</html>

// filename personal_info.php
<?php
$name = "John Doe";
$age = 30;
$occupation = "Software Engineer";
```

```
$hobbies = "Reading, Hiking, Playing Guitar";
$email = "johndoe@example.com";
$phone = "123-456-7890";
$address = "123 Main Street, City, State, ZIP";

echo "<p><strong>Name:</strong> $name</p>";
echo "<p><strong>Age:</strong> $age</p>";
echo "<p><strong>Occupation:</strong> $occupation</p>";
echo "<p><strong>Hobbies:</strong> $hobbies</p>";
echo "<p><strong>Contact Information:</strong></p>";
echo "<ul>";
echo "<li><strong>Email:</strong> $email</li>";
echo "<li><strong>Phone:</strong> $phone</li>";
echo "<li><strong>Address:</strong> $address</li>";
echo "</ul>";
?>
```

To write a program for display the date and time.

```
<?php
$currentTime = date("Y-m-d H:i:s");
echo "Current date and time: " . $currentTime;
?>
```

To write a program for displaying our curriculum vitae.

```
<!DOCTYPE html>
<html>
<head>
    <title>Curriculum Vitae</title>
</head>
<body>
    <h1>Curriculum Vitae</h1>

    <?php
    $name = "Your Name";
    $email = "your_email@example.com";
    $phone = "123-456-7890";
    $linkedin = "https://www.linkedin.com/in/your-linkedin-profile";

    $education = [
        "degree" => "Bachelor of Science in Computer Science",
        "university" => "University Name",
        "graduationYear" => "2024"
    ];

    $skills = [
        "programmingLanguages" => "Python, Java, C++",
        "frameworks" => "Django, Flask, React",
        "databases" => "MySQL, PostgreSQL",
        "tools" => "Git, GitHub, Visual Studio Code"
    ];

    $experience = [
        "position" => "Software Developer",
        "company" => "Company Name",
        "dates" => "2023 - Present",
        "responsibilities" => [
            "Developed web applications using Python and Django",
            "Worked with MySQL databases",
            "Collaborated with a team of developers"
        ]
    ];

    $projects = [
        "projectName" => "Project Name",
        "technologies" => "Python, Django, MySQL",
        "link" => "https://github.com/your-username/project-repo"
    ];
    ?>

    <p><strong>Name:</strong> <?php echo $name; ?></p>
    <p><strong>Email:</strong> <?php echo $email; ?></p>
    <p><strong>Phone:</strong> <?php echo $phone; ?></p>
    <p><strong>LinkedIn:</strong> <?php echo $linkedin; ?></p>

    <h3>Education</h3>
    <p><?php echo $education["degree"] . " from " . $education["university"] . " in " .
$education["graduationYear"]; ?></p>
```

```
  <h3>Skills</h3>
  <p><?php echo implode(", ", $skills["programmingLanguages"]); ?></p>
  <p><?php echo implode(", ", $skills["frameworks"]); ?></p>
  <p><?php echo implode(", ", $skills["databases"]); ?></p>
  <p><?php echo implode(", ", $skills["tools"]); ?></p>

  <h3>Experience</h3>
  <p><strong>Position:</strong> <?php echo $experience["position"]; ?></p>
  <p><strong>Company:</strong> <?php echo $experience["company"]; ?></p>
  <p><strong>Dates:</strong> <?php echo $experience["dates"]; ?></p>
  <p><strong>Responsibilities:</strong></p>
  <ul>
    <?php foreach ($experience["responsibilities"] as $responsibility) { ?>
      <li><?php echo $responsibility; ?></li>
    <?php } ?>
  </ul>

  <h3>Projects</h3>
  <p><strong>Project Name:</strong> <?php echo $projects["projectName"]; ?></p>
  <p><strong>Technologies:</strong> <?php echo $projects["technologies"]; ?></p>
  <p><strong>Link:</strong> <?php echo $projects["link"]; ?></p>
</body>
</html>
```

UNit-4 (Revision)

1. **Write a PHP script to limit the maximum number of concurrent sessions for a user to 3.**

```
<?php
session_start();
$maxSession=3;
if(!isset($_SESSION['session_count'])){
	$_SESSION['session_count']=1;
}
else{
	$_SESSION['session_count']++;
	if($_SESSION['session_count']>$maxSession){
	session_unset();
	session_destroy();
	echo " Session is not exist";
	exit;
	}
}
echo "Session_count . Session_alert ".$_SESSION['session_count'];
?>
```

2. Write a PHP script to display the last time the session was accessed by the user.

```
<?php
session_start();
if(isset($_SESSION['last_at'])){
$last=$_SESSION['last_at'];
echo "Last access time ".date('Y-m-d H:i:s',$last);
$_SESSION['last_at']=time();
}
else{
$_SESSION['last_at']=time();
echo"Session started first access.";
}
?>
```

3. Write a PHP script to set a cookie and a session variable with the same name. Display their values to compare.

```
<?php
        $cookieName="myCookie";
        $value="Cookie Value";
        setcookie($cookieName,$value,time() + 3600, "/");
        session_start();
        $_SESSION[$cookieName]=$value;
        echo "Cookie value: " .$_COOKIE[$cookieName];
        echo "<br>Session value: ".$_SESSION[$cookieName];
?>
```

4. Write a PHP script to retrieve and display user preferences stored in the session variable.

```
<?php
session_start();
if (isset($_SESSION['preferences'])){
        $userPreferences = $_SESSION['preferences'];
        echo "User Preferences:</br>";
        foreach ($userPreferences as $key => $value) {
        echo $key . ": " . $value . "</br>";
        }
 }
else
{
        echo "No user preferences found.";
}
?>
```

5. Write a PHP script to set a session timeout after 30 minutes of inactivity.

```
<?php
session_start();
$timeout = 1800;
if (isset($_SESSION['LAST_ACTIVITY']) && (time() - $_SESSION['LAST_ACTIVITY'] >
$timeout)) {
        session_unset();
        session_destroy();
        echo "Session expired. Please log in again.";
} else {
        $_SESSION['LAST_ACTIVITY'] = time(); // Update last activity time
        echo "Session is active.";
}
?>
```

6. Write a PHP calculator class which will accept two values as arguments, then add them, subtract them, multiply them together, or divide them on request.

```
<?php
class MyCalculator {
        public $a, $b;
        public function __construct($a, $b) {
        $this->a = $a;
        $this->b = $b;
        }
        public function add() {
        return $this->a + $this->b;
        }

        public function subtract() {
        return $this->a - $this->b;
        }
        public function multiply() {
        return $this->a * $this->b;
        }
        public function divide() {
        return $this->b != 0 ? $this->a / $this->b : "Cannot divide by zero";
        }
}
$calc = new MyCalculator(12, 6);
echo $calc->add() . "\n";       // 18
echo $calc->subtract() . "\n";  // 6
echo $calc->multiply() . "\n";  // 72
echo $calc->divide() . "\n";      // 2
?>
```

7. Write a PHP class that sorts an ordered integer array with the help of sort() function.

Sample array : **array(11, -2, 4, 35, 0, 8, -9)**

```
<?php

class ArraySort {

        protected $array;

        public function __construct(array $array) {

        $this->array = $array;

        }

        public function sortAscending() {

        $sorted = $this->array;

        sort($sorted);

        return $sorted;

        }

}

$sortArray = new ArraySort([11, -2, 4, 35, 0, 8, -9]);

print_r($sortArray->sortAscending());

?>
```

8. Write a PHP class 'Rectangle' that has properties for length and width. Implement methods to calculate the rectangle's area and perimeter.

```
<?php
```

```
class Rectangle {
    private $length;
    private $width;

    public function __construct($length, $width) {
    $this->length = $length;
    $this->width = $width;
    }

    public function getArea() {
    return $this->length * $this->width;
    }

    public function getPerimeter() {
    return 2 * ($this->length + $this->width);
    }
}

$rectangle = new Rectangle(12, 9);
echo "Area: " . $rectangle->getArea() . "</br>";
echo "Perimeter: " . $rectangle->getPerimeter() . "</br>";
?>
```

9. Write a PHP class called 'Shape' with an abstract method 'calculateArea()'. Create two subclasses, 'Triangle' and 'Rectangle', that implement the 'calculateArea()' method.

```
<?php
abstract class Shape {
        abstract public function calculateArea();
}

class Triangle extends Shape {
        private $base, $height;

        public function __construct($base, $height) {
        $this->base = $base;
        $this->height = $height;
        }

        public function calculateArea() {
        return 0.5 * $this->base * $this->height;
        }
}

class Rectangle extends Shape {
        private $length, $width;
```

```
    public function __construct($length, $width) {
    $this->length = $length;
    $this->width = $width;
    }

    public function calculateArea() {
    return $this->length * $this->width;
    }
}

$triangle = new Triangle(5, 7);
echo "Triangle Area: " . $triangle->calculateArea() . "<br>";

$rectangle = new Rectangle(4, 6);
echo "Rectangle Area: " . $rectangle->calculateArea() . "<br>";
?>
```

10. Write a PHP interface called 'Resizable' with a method 'resize()'. Implement the 'Resizable' interface in a class called 'Square' and add functionality to resize the square.

```
<?php
// Resizable interface with resize method
interface Resizable {
    public function resize($newSize);
}


// Square class implementing the Resizable interface
class Square implements Resizable {
    private $side;


    // Constructor to initialize side length
    public function __construct($side) {
    $this->side = $side;
    }


    // Calculate area
    public function area() {
    return $this->side * $this->side;
    }


    // Resize the square
    public function resize($newSize) {
    $this->side = $newSize;
    }
```

```
        // Get side length
        public function getSide() {
        return $this->side;
        }
}

// Create a square with side 4
$square = new Square(4);
echo "Side: " . $square->getSide() . "<br>";
echo "Area: " . $square->area() . "<br>";

// Resize square to side 6
$square->resize(6);
echo "New Side: " . $square->getSide() . "<br>";
echo "New Area: " . $square->area() . "<br>";
?>
```