**LAB 8 TASK**

**K20 1052**

**BSE-6B**

1-

<?php

$integer = 42;

echo "Integer: " . $integer . "\n";

$float = 3.14;

echo "Float: " . $float . "\n";

$string = "Hello, world!";

echo "String: " . $string . "\n";

$boolean = true;

echo "Boolean: " . $boolean . "\n";

$array = array("apple", "banana", "orange");

echo "Array: ";

print\_r($array);

class Person {

  public $name;

  public $age;

  function \_\_construct($name, $age) {

    $this->name = $name;

    $this->age = $age;

  }

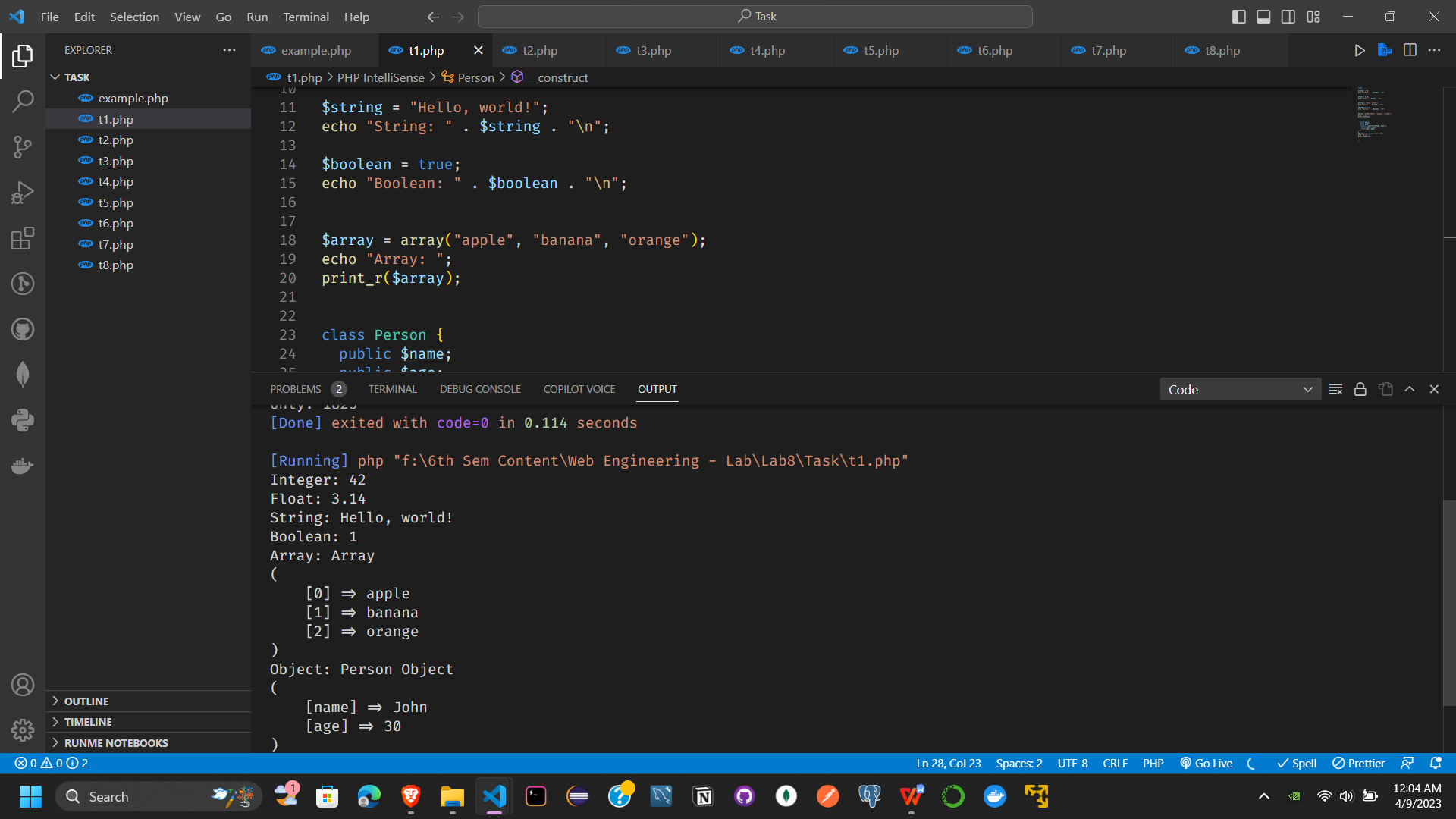
}

$person = new Person("John", 30);

echo "Object: ";

print\_r($person);

?>



2-

<?php

function operators\_demo() {

  $a = 5;

  $b = "10";

  // Arithmetic operators

  echo "Arithmetic operators:\n";

  echo "Addition: " . ($a + $b) . "\n";

  echo "Subtraction: " . ($a - $b) . "\n";

  echo "Multiplication: " . ($a \* $b) . "\n";

  echo "Division: " . ($a / $b) . "\n";

  echo "Modulus: " . ($a % $b) . "\n\n";

  // Comparison operators

  echo "Comparison operators:\n";

  echo "Equal: " . ($a == $b) . "\n";

  echo "Identical: " . ($a === $b) . "\n";

  echo "Not equal: " . ($a != $b) . "\n";

  echo "Not identical: " . ($a !== $b) . "\n";

  echo "Greater than: " . ($a > $b) . "\n";

  echo "Less than: " . ($a < $b) . "\n";

  echo "Greater than or equal to: " . ($a >= $b) . "\n";

  echo "Less than or equal to: " . ($a <= $b) . "\n\n";

  // Logical operators

  echo "Logical operators:\n";

  echo "And: " . ($a && $b) . "\n";

  echo "Or: " . ($a || $b) . "\n";

  echo "Not: " . (!$a) . "\n\n";

  // Bitwise operators

  $c = 10;

  echo "Bitwise operators:\n";

  echo "And: " . ($a & $c) . "\n";

  echo "Or: " . ($a | $c) . "\n";

  echo "Xor: " . ($a ^ $c) . "\n";

  echo "Not: " . (~$a) . "\n";

  echo "Left shift: " . ($a << 2) . "\n";

  echo "Right shift: " . ($a >> 1) . "\n\n";

  // Assignment operators

  echo "Assignment operators:\n";

  $a += $b;

  echo "Add and assign: " . $a . "\n";

  $a -= $b;

  echo "Subtract and assign: " . $a . "\n";

  $a \*= $b;

  echo "Multiply and assign: " . $a . "\n";

  $a /= $b;

  echo "Divide and assign: " . $a . "\n";

  $a %= $b;

  echo "Modulus and assign: " . $a . "\n";

  $a &= $c;

  echo "Bitwise AND and assign: " . $a . "\n";

  $a |= $c;

  echo "Bitwise OR and assign: " . $a . "\n";

  $a ^= $c;

  echo "Bitwise XOR and assign: " . $a . "\n";

  $a <<= 2;

  echo "Left shift and assign: " . $a . "\n";

  $a >>= 1;

  echo "Right shift and assign: " . $a . "\n\n";

  // Increment and decrement operators

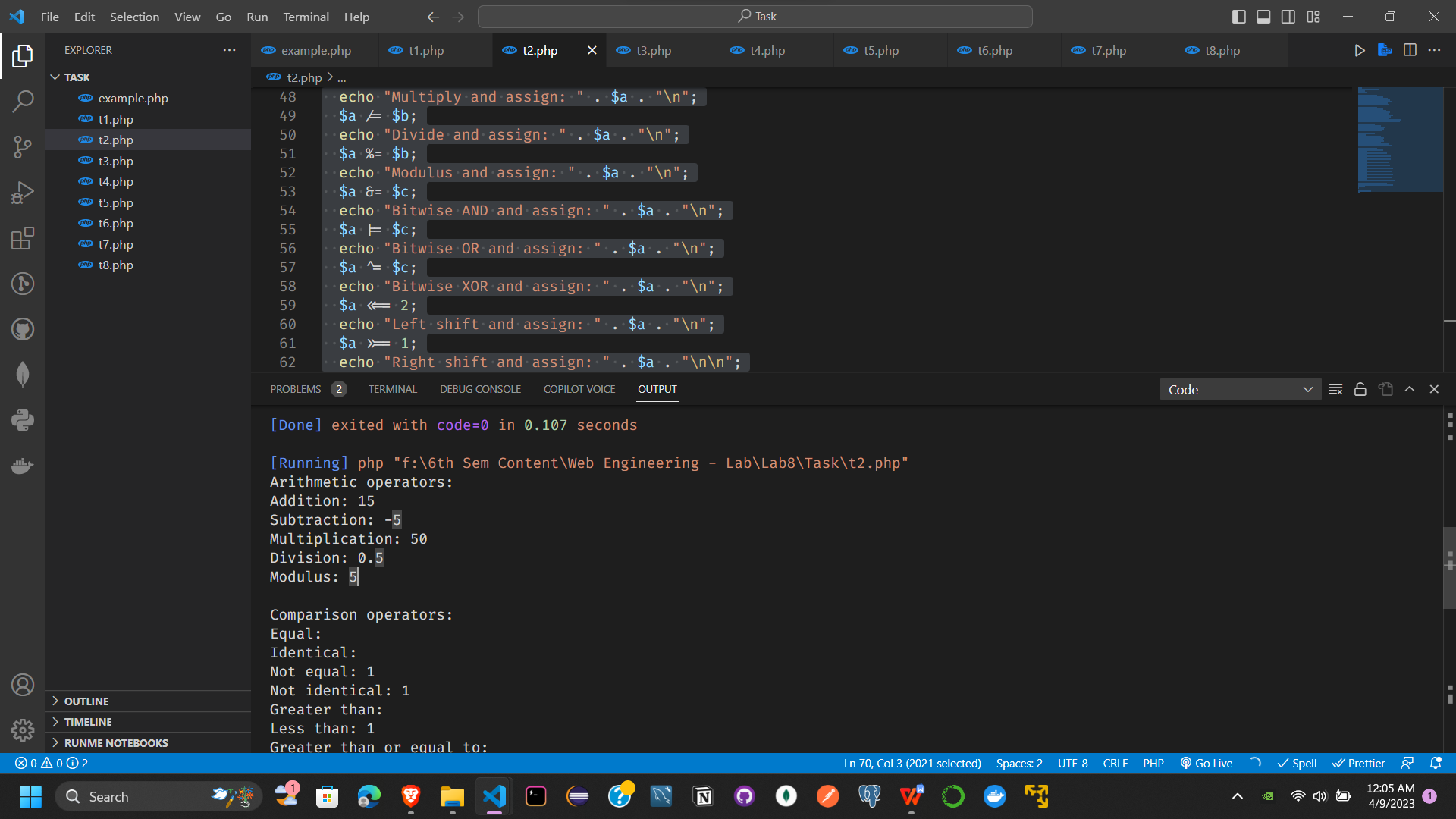
  echo "Increment and decrement operators:\n";

  $a = 5;

}

operators\_demo();

?>



3-

<!-- Task 03: Write a program which will find the factorial of a given number.

Exit the program if the input number is negative.

Example of Factorial: Input number = 5

Factorial is=5\*4\*3\*2\*1

Note: Justify your choice of loop answering two important points:

  why your choice is optimal? why other looping structure would not be suitable? -->

 <?php

  $number = fun(5);

  echo $number;

  function fun($n){

    if($n == 0){

      return 1;

    }

    else{

      return $n \* fun($n-1);

    }

  }

  ?>

  <?php

  $number =5;

  $factorial = 1;

  for ($x=$number; $x>=1; $x--)

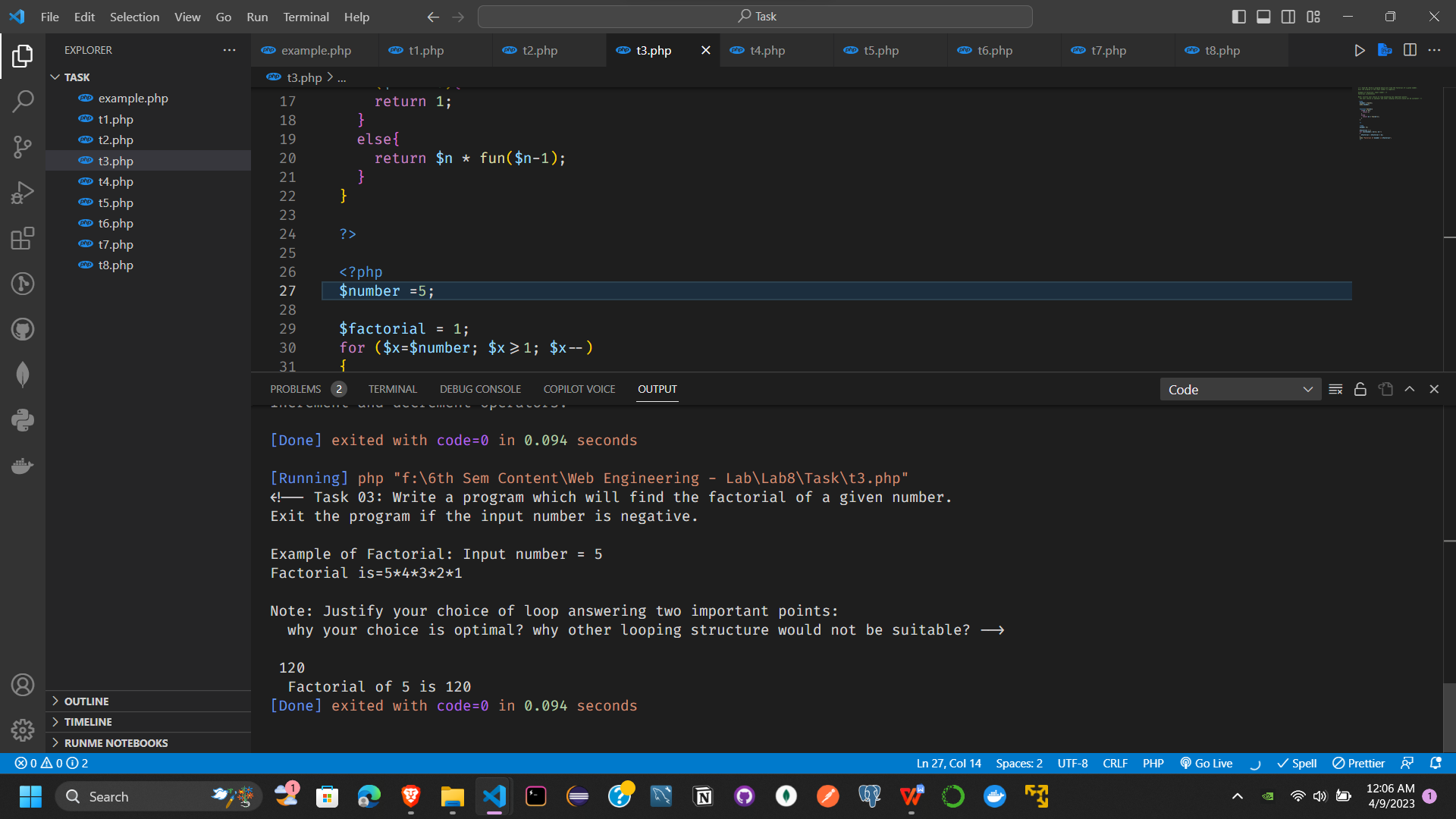
  {

    $factorial = $factorial \* $x;

  }

  echo "Factorial of $number is $factorial";

  ?>



4-

<!-- Task 04: Write a program which will generate the Fibonacci series up to 1000.

Also find the sum of the generated Fibonacci numbers divisible by 3, 5 and 7 only.

Example of Fibonacci series is: 1 1 2 3 5 8 13 25..........

Note: Do this task by using for loop and while loop. Also identify which one is more efficient? -->

<?php

$max = 1000;

// initialize the first two Fibonacci numbers

$first = 0;

$second = 1;

$sum = 0;

// loop to generate the Fibonacci series up to 1000

while ($first + $second <= $max) {

  $third = $first + $second;

  $first = $second;

  $second = $third;

  // check if the current Fibonacci number is divisible by 3, 5 and 7

  if ($third % 3 == 0 || $third % 5 == 0 || $third % 7 == 0) {

    $sum += $third;

    // echo $sum . " ";

  }

}

$first = 0;

$second = 1;

echo "Fibonacci series up to 1000: 0 1 ";

for ($i = 1; $i <= $max; $i = $first + $second) {

  $third = $first + $second;

  echo $third . " ";

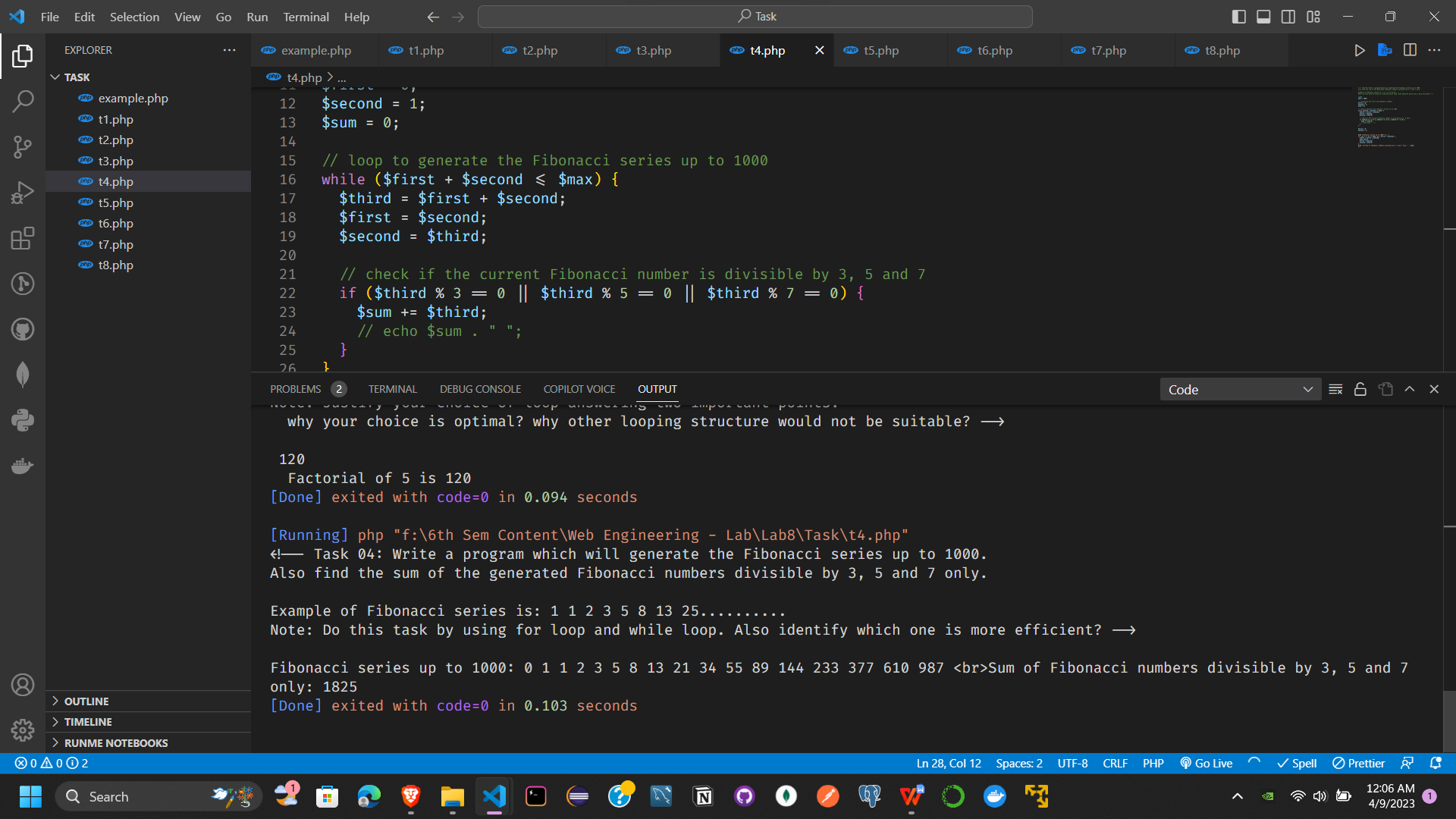
  $first = $second;

  $second = $third;

}

echo "<br>Sum of Fibonacci numbers divisible by 3, 5 and 7 only: " . $sum;

?>



5-

<?php

$array = array(5, 4, 3, 2, 1, 6, 10, 9, 7, 8);

$element = 7;

$index = 0;

$found = false;

function search($array, $element, $index, $found)

{

  if ($index < count($array)) {

    if ($array[$index] == $element) {

      $found = true;

      echo "Element found at index: " . $index;

    } else {

      $index++;

      search($array, $element, $index, $found);

    }

  } else {

    if ($found == false) {

      echo "Element not found";

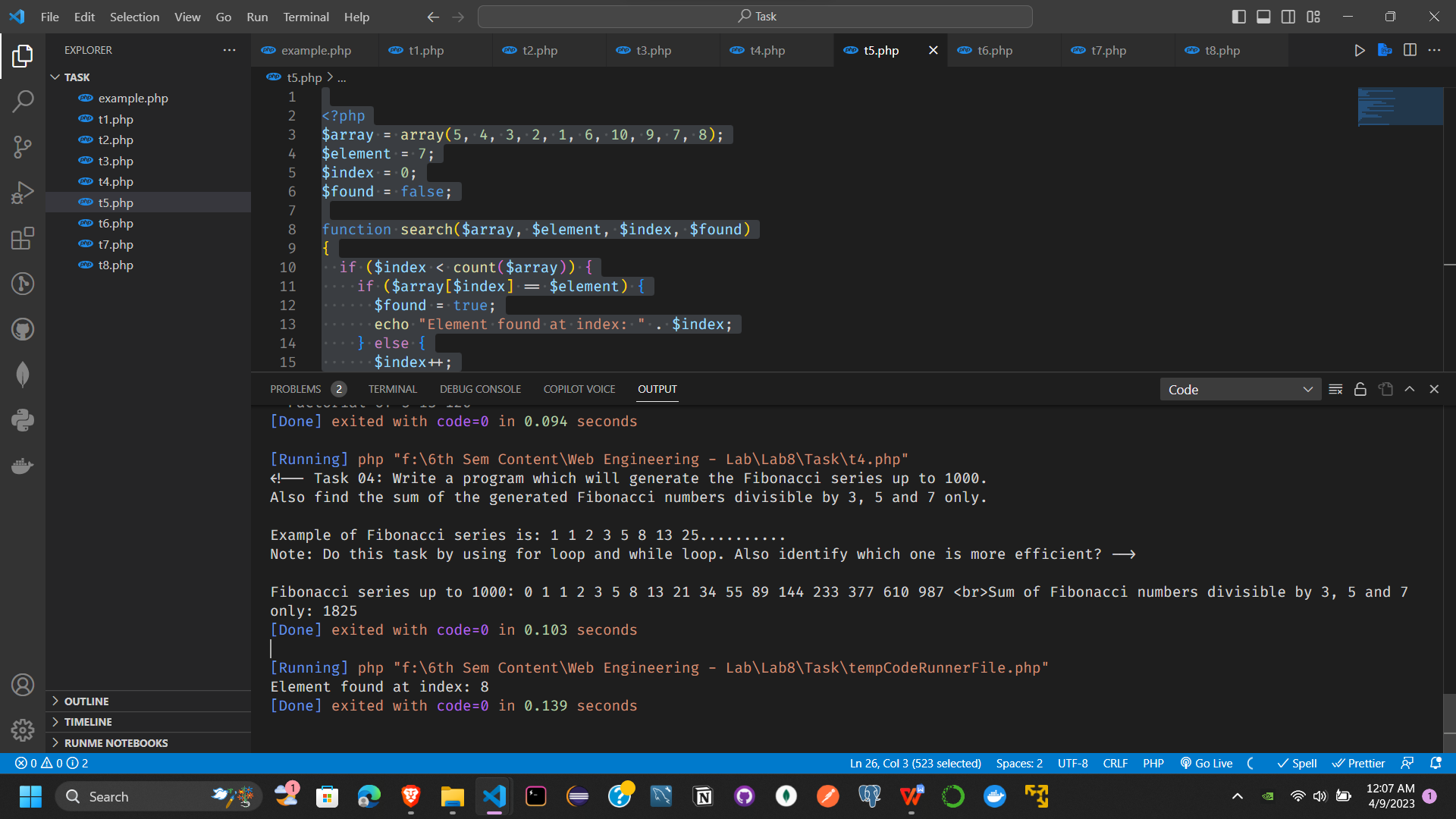
    }

  }

}

search($array, $element, $index, $found);

?>



6-

<?php

$a = 8;

$b = 9;

$c = 3;

// calculate the value of b2 - 4ac

$discriminant = $b \* $b - 4 \* $a \* $c;

// find the roots of the quadratic equation

switch ($discriminant) {

  case $discriminant > 0:

    $root1 = (-$b + sqrt($discriminant)) / (2 \* $a);

    $root2 = (-$b - sqrt($discriminant)) / (2 \* $a);

    echo "The roots are $root1 and $root2";

    break;

  case $discriminant == 0:

    $root1 = $root2 = -$b / (2 \* $a);

    echo "The roots are $root1 and $root2";

    break;

  case $discriminant < 0:

    $realPart = -$b / (2 \* $a);

    $imaginaryPart = sqrt(-$discriminant) / (2 \* $a);

    echo "The roots are $realPart + $imaginaryPart i and $realPart - $imaginaryPart i";

    break;

}

echo "<br>";

// with if else

if ($discriminant > 0) {

  $root1 = (-$b + sqrt($discriminant)) / (2 \* $a);

  $root2 = (-$b - sqrt($discriminant)) / (2 \* $a);

  echo "The roots are $root1 and $root2";

} else if ($discriminant == 0) {

  $root1 = $root2 = -$b / (2 \* $a);

  echo "The roots are $root1 and $root2";

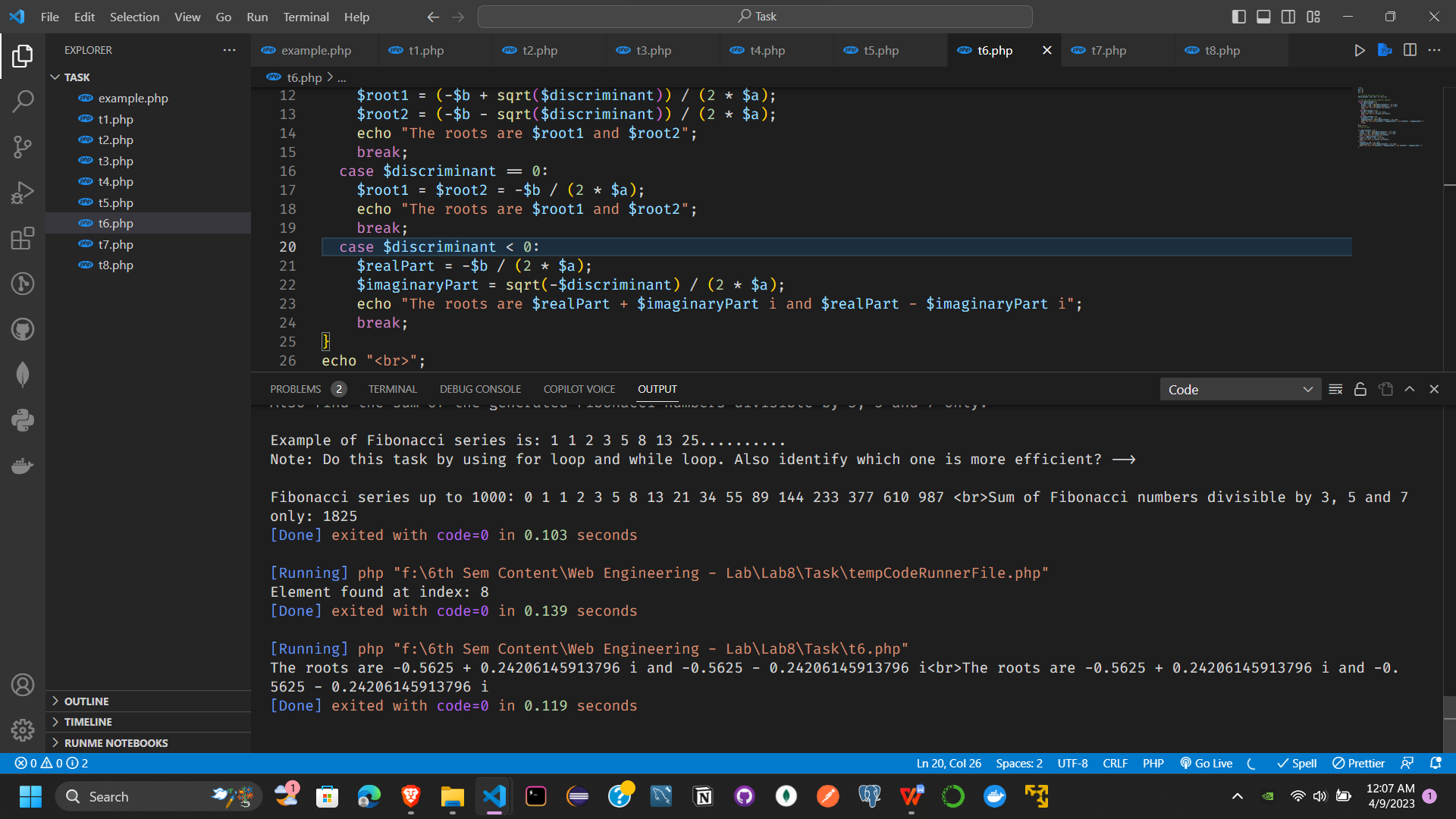
} else {

  $realPart = -$b / (2 \* $a);

  $imaginaryPart = sqrt(-$discriminant) / (2 \* $a);

  echo "The roots are $realPart + $imaginaryPart i and $realPart - $imaginaryPart i";

}



7-

<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <title>Document</title>

</head>

<body>

  <h1>Signup Form</h1>

  <form action="" method="get">

    <label for="name">Name</label>

    <input type="text" name="name" id="name">

    <br>

    <label for="email">Email</label>

    <input type="email" name="email" id="email">

    <br>

    <label for="phone">Phone</label>

    <input type="number" name="phone" id="phone">

    <br>

    <input type="submit" value="Submit">

  </form>

  <?php

  $name = $\_GET['name'];

  $email = $\_GET['email'];

  $phone = $\_GET['phone'];

  echo "<br>Name: " . $name . "<br>";

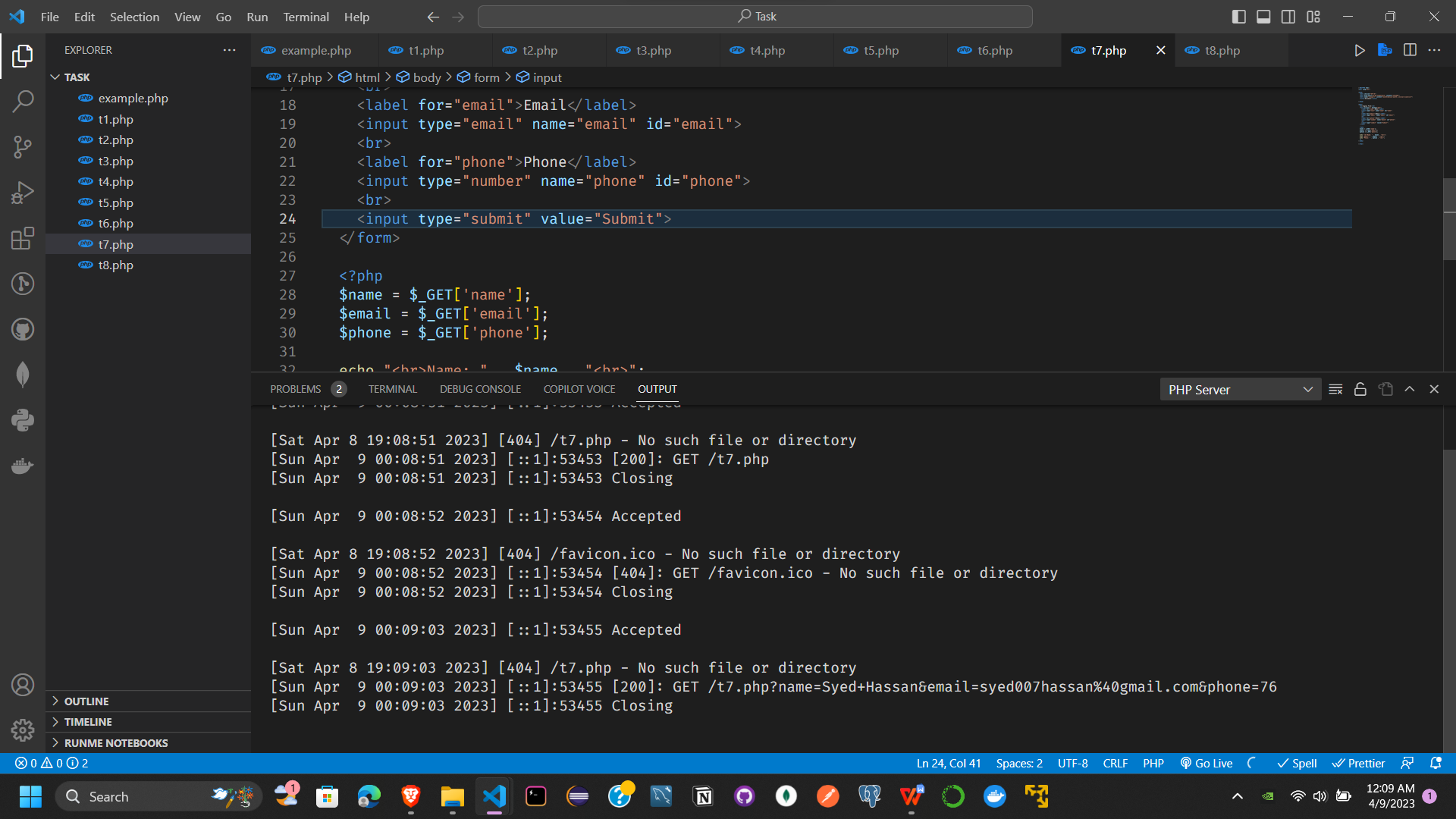
  echo "Email: " . $email . "<br>";

  echo "Phone: " . $phone . "<br>";

  ?>

</body>

</html>



8-

<?php

$txt = "Hello world!";

$x = 5;

$y = 10.5;

echo $txt;

echo "<br>";

echo $x;

echo "<br>";

echo $y;

// operators in php

// arithmetic operators

// + - \* / % \*\*

$x = 10;

$y = 6;

echo $x + $y;

echo "<br>";

echo $x - $y;

echo "<br>";

echo $x \* $y;

echo "<br>";

echo $x / $y;

echo "<br>";

echo $x % $y;

echo "<br>";

echo $x \*\* $y;

echo "<br>";

// assignment operators

// = += -= \*= /= %= \*\*=

$x = 10;

$x += 5;

echo $x;

echo "<br>";

$x = 10;

$x -= 5;

echo $x;

echo "<br>";

$x = 10;

$x \*= 5;

echo $x;

echo "<br>";

$x = 10;

$x /= 5;

echo $x;

echo "<br>";

$x = 10;

$x %= 5;

echo $x;

echo "<br>";

$x = 10;

$x \*\*= 5;

echo $x;

echo "<br>";

// comparison operators

// == != > < >= <= === !==

$x = 10;

$y = 10;

var\_dump($x == $y);

echo "<br>";

var\_dump($x != $y);

echo "<br>";

var\_dump($x > $y);

echo "<br>";

var\_dump($x < $y);

echo "<br>";

var\_dump($x >= $y);

echo "<br>";

var\_dump($x <= $y);

echo "<br>";

var\_dump($x === $y);

echo "<br>";

var\_dump($x !== $y);

echo "<br>";

// increment and decrement operators

// ++ --

$x = 10;

echo ++$x;

echo "<br>";

echo $x++;

echo "<br>";

echo $x;

echo "<br>";

echo --$x;

echo "<br>";

echo $x--;

echo "<br>";

echo $x;

echo "<br>";

// logical operators

// and && or || xor !

$x = 10;

$y = 20;

var\_dump($x < 20 and $y > 10);

echo "<br>";

var\_dump($x < 20 && $y > 10);

echo "<br>";

var\_dump($x < 20 or $y > 10);

echo "<br>";

var\_dump($x < 20 || $y > 10);

echo "<br>";

var\_dump($x < 20 xor $y > 10);

echo "<br>";

var\_dump(!($x < 20 and $y > 10));

echo "<br>";

// string operators

// . .=

$x = "Hello";

$y = "World";

echo $x . $y;

echo "<br>";

echo $x .= $y;

echo "<br>";

// array operators

// + == === != !==

$x = array("a" => "red", "b" => "green");

$y = array("c" => "blue", "d" => "yellow");

var\_dump($x + $y);

echo "<br>";

var\_dump($x == $y);

echo "<br>";

var\_dump($x === $y);

echo "<br>";

var\_dump($x != $y);

echo "<br>";

var\_dump($x !== $y);

echo "<br>";

// conditional statements

// if else elseif

$t = date("H");

if ($t < "20") {

  echo "Have a good day!";

}

// switch

$favcolor = "red";

switch ($favcolor) {

  case "red":

    echo "Your favorite color is red!";

    break;

  case "blue":

    echo "Your favorite color is blue!";

    break;

  case "green":

    echo "Your favorite color is green!";

    break;

  default:

    echo "Your favorite color is neither red, blue, nor green!";

}

// loops

// while

$x = 1;

while ($x <= 5) {

  echo "The number is: $x <br>";

  $x++;

}

// do while

$x = 1;

do {

  echo "The number is: $x <br>";

  $x++;

} while ($x <= 5);

// for

for ($x = 0; $x <= 10; $x++) {

  echo "The number is: $x <br>";

}

// foreach

$colors = array("red", "green", "blue", "yellow");

foreach ($colors as $value) {

  echo "$value <br>";

}

