***PHP Test***

**Q1** : Implement a groupByOwners function that :

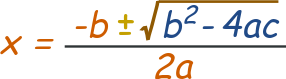
* Accepts an associative array containing the file owner name for each file name .
* Returns an associative array containing an array of file names for each owner name , in an order .

For example, for associative array [“Input.txt” => “Randy”,” Code.py”=> “Stan”,”Output.txt” =>”Randy”] the groupByOwners function should return [“Randy” => [“Input.txt”, “Output.txt”], “Stan”=> [“Code.py”]] .

**Q2**: Quadratic Equation Implement the function findRoots to find the roots of the quadratic equation:

ax2 + bx + c = 0. The function should return an array containing both roots in any order. If the equation has only one solution, the function should return that solution as both elements of the array. The equation will always have at least one solution.

The roots of the quadratic equation can be found with the following formula :



For example , find Roots (2, 10, 8) should return [-1,-4] or [-4,-1] as the roots of the equation 2X2 + 10X + 8 = 0 are -1 and -4 .

Answer 1 :

|  |
| --- |
| <?php |
|  |

|  |
| --- |
| class FileOwners |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| public static function groupByOwners($files) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| $file\_return = []; |
|  |

|  |
| --- |
| $files\_copy = $files; |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| print\_r($files); |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| foreach($files as $x => $x\_value) { |
|  |

|  |
| --- |
| $found = false; |
|  |

|  |
| --- |
| if (array\_key\_exists($x\_value,$file\_return)) |
|  |

|  |
| --- |
| array\_push($file\_return[$x\_value], $x); |
|  |

|  |
| --- |
| else { |
|  |

|  |
| --- |
| $file\_return[$x\_value] = array(); |
|  |

|  |
| --- |
| array\_push($file\_return[$x\_value], $x); |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| print\_r($file\_return); |
|  |

|  |
| --- |
| return NULL; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

|  |
| --- |
| $files = array |
|  |

|  |
| --- |
| ( |
|  |

|  |
| --- |
| "Input.txt" => "Randy", |
|  |

|  |
| --- |
| "Code.py" => "Stan", |
|  |

|  |
| --- |
| "Output.txt" => "Randy" |
|  |

|  |
| --- |
| ); |
|  |

var\_dump(FileOwners::groupByOwners($files));

function groupByOwners(array $files) : array

{

$result = [];

foreach($files as $file => $owner) {

$result[$owner][] = $file;

}

return $result;

}

**Answer 2 :**

|  |
| --- |
|  |
| <?php | |
|  | |

|  |
| --- |
| /\*\* |
|  |

|  |
| --- |
| \* @return array An array of two elements containing roots in any order |
|  |

|  |
| --- |
| \*/ |
|  |

|  |
| --- |
| function findRoots($a, $b, $c) |
|  |

|  |
| --- |
| { |
|  |

|  |
| --- |
| $delta = ($b \* $b) - 4 \* ($a \* $c); |
|  |

|  |
| --- |
| $x = (- $b - sqrt($delta)) / (2 \* $a); |
|  |

|  |
| --- |
| $y = (- $b + sqrt($delta)) / (2 \* $a); |
|  |

|  |
| --- |
| return [$x, $y]; |
|  |

|  |
| --- |
| } |
|  |

|  |
| --- |
|  |
|  |

print\_r(findRoots(2, 10, 8));