Below is the written test for Full-Stack Developer position. Please read through the entire test before starting to write it.

1) Write the following function's body. A nested array is passed as parameter. You need to print all keys with their depth.

Sample Input:

```
$a = array (
        "key1" => 1,
        "key2" => array (
                "key3" => 1,
                "key4" => array (
                        "key5" => 4
                ),
        ),
);
```

Sample Output:

```
key1 1
key2 1
key3 2
key4 2
key5 3
```

```
function printDepth($data) {
        <write function body>
}
```

You may write additional functions. If you choose to use PHP built-in functions. Please write out the pseudo code for how they process the data.

2) Write a new function with same functionality from Question 1, but it should be able to handle a PHP object in addition to an array from Question 1.

Sample Input: class Person {

```
public function __construct($first_name, $last_name, $father) {
        $this->first_name = $first_name;
        $this->last_name = $last_name;
        $this->father = $father;
}
$person_a = new Person("User", "1", NULL)
$person_b = new Person("User", "2", $person_a)
$a = array (
        "key1" => 1,
        "key2" => array ( "key3" => 1,
        ),
);
```

Sample Output: key1 1
key2 1
key3 2

key4 2
key5 3
user: 3 first_name: 4 last_name: 4 father: 4
first_name: 5 last_name: 5 father: 5
function printDepth($data) { <write function body>

}

"key4" => array ("key5" => 4,
“User” => $person_b,),

You may write additional functions or update the initial object definition. You may write additional functions. If you choose to use PHP built-in functions. Please write out the pseudo code for how they process the data.

3) Write following functions body. 2 Nodes are passed as parameter. You need to find Least Common Ancestor and print its value. Node structure are as following:

class Node { $value;

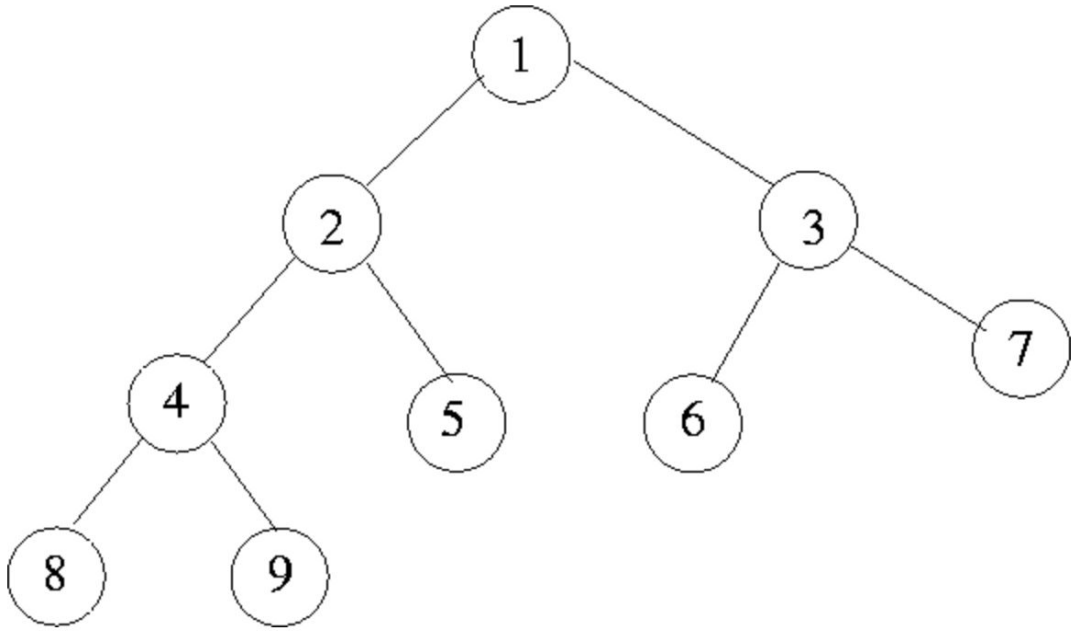
$parent; }

Ancestor Definition:

1. 1) Any node falls under parent chain till root node.
2. 2) A node is an ancestor of itself.

For example: if we consider Node 7 it’s ancestors will be 1, 3, and 7. All nodes values are unique for this tree.

You function needs to find least common ancestor (closest common ancestor). For an example for the tree image, if 6 and 7 passed to lca it should return 3, if 3 and 7 passed to lca it should return 3



function lca($node1, $node2) { <implement function

}
You may write additional functions.
Explain the Runtime and Memory requirements for your solution.