

Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação.

Disciplina: Programacao II

AD1 - 1º semestre de 2012 - GABARITO

**Exercício 1:**

**a)**

```
<?php
```

```
function calcula_valor_pago($renda) {  
    $tabela = array(  
        array(0, 1566.61, 0, 0),  
        array(1566.62, 2347.85, 0.075, 117.49),  
        array(2347.86, 3130.51, 0.15, 293.58),  
        array(3130.52, 3911.63, 0.225, 528.37),  
        array(3911.63, 0, 0.275, 723.95)  
    );  
    $i = 0;  
    while($i < count($tabela) - 1) {  
        if($tabela[$i][1] >= $renda)  
            break;  
        $i++;  
    }  
    return $tabela[$i][2] * $renda - $tabela[$i][3];  
}  
?>
```

**b)**

```
function retorna_valor_pago_formatado($renda) {  
    setlocale(LC_ALL, 'pt_BR');  
    return sprintf("R$ %.2f", $renda);  
}
```

**Exercício 2:**

- a) `preg_match('/^\d{1,2}\.\d{1,2}\.\d{4}$/', $string)`
- b) `preg_match('/^[A-Z][a-z ]+([ ][A-Z][a-z ]+)*$/ ', $string)`

- c) `preg_match('/^[^0-9][a-zA-Z0-9_]+([.][a-zA-Z0-9_]+)*[.][a-zA-Z0-9_]*[.][a-zA-Z]{2,4}$/', $string)` **OBSERVAÇÃO: Solução mais geral do que a pedida na questão.**
- d) `preg_match('/^([0-9]{3}\.){2}[0-9]{3}-[0-9]{2}$/', $string)`

### Exercício 3:

a)

```
class Produto {

    private $_nome;

    private $_data_fabricacao;

    private $_validade;

    function __construct($nome, $data_fabricacao, $validade) {

        $this->_nome = $nome;

        $this->_data_fabricacao = $data_fabricacao;

        $this->_validade = $validade;

    }

}
```

b)

```
[..]

function isValid() {

    return $this->_validade > time('Y-m-d');

}

[..]
```

c)

```
<?php

$produto = new Produto('arroz', strtotime('2012-01-01'), strtotime('2012-02-01'));

if($produto->isValid()) echo "O produto está dentro da validade. ";

else echo "O produto está fora da validade. ";

?>
```

### Exercício 4:

Para as questões a seguir, assuma \$sudoku como a matriz global 9x9 do jogo atual, e a seguinte função utilitária:

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

function valores_distintos($array) {
    return (count(array_unique($array)) == 9);
}
```

**a)**

```
function verifica_linha($num_coluna) {
    global $sudoku;
    for ($i = 0; $i < 9; $i++)
        $arr[$i] = $sudoku[$num_coluna][$i];
    return valores_distintos($arr);
}
```

**b)**

```
function verifica_coluna($num_linha) {
    global $sudoku;
    for ($i = 0; $i < 9; $i++)
        $arr[$i] = $sudoku[$i][$num_linha];
    return valores_distintos($arr);
}
```

**c)**

```
function verifica_regiao($num_regiao) {
    global $sudoku;
```

```
    $linha_inicio = ((int)((int) $num_regiao / (int) 3)) * 3; //divisao inteira no php via cast
```

```
$col_inicio = ($num_regiao % 3) * 3;
```

```
for ($i = $col_inicio; $i < $col_inicio + 3; $i++)
```

```
    for ($j = $linha_inicio; $j < $linha_inicio + 3; $j++)
```

```
        $arr[] = $sudoku[$i][$j]; //operador de concatenacao ao final do array
```

```
    return valores_distintos($arr);
```

```
}
```

**d)**

```
function resultado_sudoku() {
```

```
    for ($i = 0; $i < 9; $i++) {
```

```
        if (!verifica_linha($i) || !verifica_coluna($i) || !verifica_regiao($i))
```

```
            return false;
```

```
    }
```

```
    return true;
```

```
}
```