



Q1	2,0	
Q2	2,0	
Q3	3,0	
Q4	3,0	

Fundação CECIERJ – Vice Presidência de Educação Superior à Distância

Curso de Tecnologia em Sistemas de Computação

Disciplina: Programação de Aplicações Web

Professores: Diego Passos e Uéverton dos Santos Souza

Gabarito da AP1 – 1º Semestre de 2019

Nome: _____

Questão 1: Suponha que você esteja implementando uma calculadora em PHP. Sua calculadora recebe os valores numéricos sobre os quais as operações serão realizadas na forma *strings* (cadeias de caracteres textuais). Antes de fazer a conversão da *string* para um número, você deseja fazer uma verificação de se o formato do número está correto. Suponha que sua calculadora aceite números com as seguintes características:

- O número pode ou não ser precedido dos sinais de menos ou mais (apenas um deles, não os dois).
- Após o sinal (se existir), o número pode ter qualquer quantidade de algarismos decimais, desde que haja ao menos um.
- Estes algarismos podem ou não ser seguidos de uma vírgula.
- A vírgula pode ser seguida de uma quantidade qualquer de algarismos decimais (pode, inclusive, ser seguida de zero algarismos).

Escreva uma expressão regular em PHP que permita verificar se uma string representa um número nesse formato.

Resposta: uma possível solução para a questão é a seguinte expressão regular:

`^[+-]?[0-9]+,[0-9]*$`

Esta expressão é composta pelos seguintes elementos:

- “^”: indica início da string.
- “[+]?”: indica que, após o início da string, pode haver um sinal (“+” ou “-”). O caractere “?” indica que os caracteres do conjunto “[+]” são esperados 0 ou 1 vezes, o que permite que o sinal seja suprimido do número.
- “[0-9]+”: indica que, após o sinal (se existir), esperamos uma sequência qualquer de algarismos decimais. O caractere “+” indica que é necessário haver ao menos um algarismo, mas pode haver mais.
- “,”?: indica que, após a parte inteira do número, espera-se possivelmente uma vírgula. O caractere “?” indica que a vírgula é opcional: pode aparecer 0 ou 1 vez.
- “[0-9]*”: similar à parte inteira, indica que a parte fracionária (i.e., depois da vírgula) pode ser composta de 0 ou mais algarismos decimais (o que é definido pelo caractere “*”).

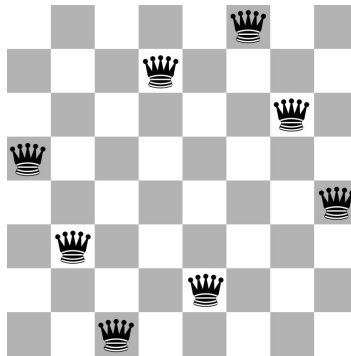
Questão 2: Escreva uma função em PHP que receba como argumento um vetor de números (não necessariamente inteiros e potencialmente repetidos), e imprima os mesmos valores, mas **sem repetição**. Por exemplo, para um vetor com os valores 1.65, 7, 3, 1.65, 3, 5, sua função deve imprimir os valores 1.65, 3, 5, 7 (a ordem não é importante).

Resposta: uma possível solução é:

```
function supprimeDuplicatas($a) {  
  
    // Vetor $semDuplicatas armazenará os elementos como  
    // chaves.  
    $semDuplicatas = array();  
    // Iteramos por todos os elementos do vetor original.  
    foreach ($a as $val) {  
        // Usamos cada um como chave do vetor $semDuplicatas,  
        // escrevendo algum valor na posição correspondente.  
        $semDuplicatas[strval($val)] = 1;  
    }  
    // Finalmente, iteramos por todas as entradas existentes em $semDuplicatas,  
    // imprimindo as chaves.  
    foreach ($semDuplicatas as $i => $val) {  
  
        echo "$i<br>";  
    }  
}
```

Questão 3: Um tabuleiro de Xadrez é composto por 8x8 casas de cores alternadas. Segundo as regras do jogo, a dama é a peça com maior mobilidade, podendo se mover quantas casas quiser na diagonal, na vertical ou na horizontal.

Em 1884, o Alemão Max Bezzel criou um problema conhecido como *Problema das 8 Rainhas*. O problema consiste em posicionar 8 rainhas em um tabuleiro de Xadrez de forma que nenhuma esteja no caminho de um possível movimento de outra. Em outras palavras, não deve haver duas rainhas na mesma diagonal, na mesma linha ou na mesma coluna. A Figura abaixo ilustra uma possível solução:



Suponha que você deseje escrever um programa em PHP para resolver o Problema das 8 Rainhas. Para representar o tabuleiro, você utiliza uma matriz de 8x8 números inteiros: se uma casa possui uma rainha, a posição correspondente da matriz tem valor 1; caso contrário, a posição correspondente tem valor 0.

Nessas condições, escreva uma função em PHP que receba como argumento um tabuleiro com as 8 rainhas já distribuídas e **verifique se o posicionamento dessas rainhas constitui uma solução** para o Problema das 8 Rainhas.

Resposta: uma possível solução é a seguinte função:

```
function verificaSolucao($tab) {
    $nRainhasNoTabuleiro = 0;
    // Encontrar a posição de cada rainha no tabuleiro. Para isso,
    // varrer todas as posições.
    for ($i = 0; $i < 8; $i = $i + 1) {
        for ($j = 0; $j < 8; $j = $j + 1) {
            if ($tab[$i][$j] == 1) {
                // Rainha encontrada.
                $nRainhasNoTabuleiro = $nRainhasNoTabuleiro + 1;
                // Verificar se há outra rainha na mesma linha.
                $rainhasNaLinha = 0;
                for ($k = 0; $k < 8; $k++) {
```

```

        if ($tab[$i][$k] == 1) $rainhasNaLinha = $rainhasNaLinha + 1;
    }
    // Verificar se há outra rainha na mesma coluna.
    $rainhasNaColuna = 0;
    for ($k = 0; $k < 8; $k++) {
        if ($tab[$k][$j] == 1) $rainhasNaColuna = $rainhasNaColuna + 1;
    }
    // Verificar se há outra rainha na mesma diagonal principal.
    $rainhasNaDiagP = 1;
    $k = 1;
    while ($k + $i < 8 and $k + $j < 8) {
        if ($tab[$k + $i][$k + $j] == 1) $rainhasNaDiagP = $rainhasNaDiagP+1;
        $k = $k + 1;
    }
    $k = 1;
    while ($i - $k >=0 and $j - $k >= 0) {
        if ($tab[$i - $k][$j - $k] == 1) $rainhasNaDiagP = $rainhasNaDiagP+1;
        $k = $k + 1;
    }
    // Verificar se há outra rainha na mesma diagonal secundaria.
    $rainhasNaDiagS = 1;
    $k = 1;
    while ($k + $i < 8 and $j - $k >= 0) {
        if ($tab[$k + $i][$j - $k] == 1) $rainhasNaDiagS = $rainhasNaDiagS+1;
        $k = $k + 1;
    }
    $k = 1;
    while ($i - $k >=0 and $j + $k < 8) {
        if ($tab[$i - $k][$j + $k] == 1) $rainhasNaDiagS = $rainhasNaDiagS+1;
        $k = $k + 1;
    }
    if ($rainhasNaLinha <> 1 or $rainhasNaColuna <> 1 or $rainhasNaDiagP <> 1
or $rainhasNaDiagS <> 1) {
        echo "Não é solução!";
        return ;
    }
}
}
}
}
if ($nRainhasNoTabuleiro <> 8) echo "Não é solução!";
else echo "É solução!";
}

```

Questão 4: Considere novamente o enunciado da questão anterior. Assuma que você decida agora usar o suporte da linguagem PHP a orientação a objetos para auxiliar na

escrita do seu programa. Em particular, você decide criar uma classe chamada *Solucao*, que representa e manipula soluções do Problema das 8 Rainhas. Sua classe deve armazenar:

- O tabuleiro como uma matriz de 8x8 números inteiros: se uma casa possui uma rainha, a posição correspondente da matriz tem valor 1; caso contrário, a posição correspondente tem valor 0.
- O número de rainhas atualmente colocadas no tabuleiro.

Além disso, sua classe deve possuir os seguintes métodos:

- Método que inicializa o tabuleiro para um estado vazio (isto é, sem nenhuma rainha).
- Método que adiciona uma rainha em uma casa do tabuleiro (dadas a linha e a coluna da casa).
- Método que remove a rainha localizada em uma casa do tabuleiro (dadas a linha e a coluna da casa).
- Método que imprime o estado atual do tabuleiro (isto é, para cada casa, se há ou não dama).

Escreva essa classe em PHP. Note que você só precisa implementar os quatro métodos pedidos: outros métodos **não são necessários** para efeito dessa questão.

Resposta: uma possível solução é:

```
class OitoRainhas {

    private $tab;
    private $nRainhasPosicionadas;

    public function InicializaTabuleiro() {

        $this->tab = array();
        for ($i = 0; $i < 8; $i = $i + 1) {

            $this->tab[$i] = array();
            for ($j = 0; $j < 8; $j = $j + 1) {

                $this->tab[$i][$j] = 0;

            }

        }

        $this->nRainhasPosicionadas = 0;

    }

}
```

```

public function InsereRainha($linha, $coluna) {

    $this->tab[$linha][$coluna] = 1;
    $this->nRainhasPosicionadas = $this->nRainhasPosicionadas + 1;
}

public function RemoveRainha($linha, $coluna) {

    $this->tab[$linha][$coluna] = 0;
    $this->nRainhasPosicionadas = $this->nRainhasPosicionadas - 1;
}

public function ImprimeTabuleiro() {

    for ($i = 0; $i < 8; $i = $i + 1) {
        for ($j = 0; $j < 8; $j = $j + 1) {

            if ($this->tab[$i][$j] == 0)
                echo "0 ";
            else
                echo "1 ";

        }
        echo "<br>";
    }
}

```

Note que o enunciado não pede que os métodos façam qualquer tipo de verificação (por exemplo, se a rainha a ser removida existe, de fato, no tabuleiro). Assim, tais verificações foram suprimidas dessa solução.