



Q1	2,0	
Q2	3,0	
Q3	2,0	
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 AP3 – 1º Semestre de 2019

Nome: _____

Questão 1: Considere que você esteja trabalhando no desenvolvimento de um editor de texto. A funcionalidade básica do editor já está implementada, mas você gostaria de adicionar capacidades de verificação ortográfica e sintática. Em particular, em um dado momento, você gostaria de escrever uma função que verifique a estrutura básica de uma frase:

- Começar sempre com uma letra maiúscula.
- Conter qualquer combinação de letras (maiúsculas e minúsculas), algarismos, espaços, hífen e vírgulas.
- Terminar com algum tipo de pontuação (ponto-final, interrogação ou exclamação).

Pede-se:

- a) Escreva uma **expressão regular** que verifique se uma string atende a essas características.
- b) Escreva uma **função (sem usar expressões regulares)** que faça a mesma verificação.

Solução:

- a) Uma possível solução é a seguinte expressão regular:

`^[A-Z][a-zA-Z0-9 \-,]*[\.!\?]\$`

Essa expressão regular é composta pelos seguintes elementos:

- “^[A-Z]”: determina que a frase deve começar, necessariamente, por uma letra maiúscula.
- “[a-zA-Z0-9 \-,*]”*: determina que, em seguida, a frase deve conter zero ou mais ocorrências de caracteres representando letras minúsculas (“a-z”), letras maiúsculas (“A-Z”), algarismos (“0-9”), espaços (“ ”), hífen (“-”) ou vírgulas (“,”).
- “[\.\!\?]*\$”: determina que, por fim, a frase deve terminar por um ponto-final, uma exclamação ou uma interrogação.

b) Uma possível solução é dada pela seguinte função:

```
function verificaFrase($frase) {

    // Verifica se primeiro caractere é uma letra maiúscula
    if ($frase[0] < 'A' or $frase[0] > 'Z') {

        // Não: frase já está errada.
        echo "Frase incorreta!<br>";
        return ;
    }

    // Primeiro caractere está ok. Itera pelos demais (exceto último).
    for ($i = 1; $i < strlen($frase) - 1; $i++) {

        // Para cada caractere intermediário da frase, verifica se é
        // maiúscula, minúscula, algarismo, espaço, hífen ou vírgula.
        // Se alguma dessas condições for verdadeira, passamos para
        // a próxima iteração.
        if ($frase[$i] >= 'A' and $frase[$i] <= 'Z') continue ;
        if ($frase[$i] >= 'a' and $frase[$i] <= 'z') continue ;
        if ($frase[$i] >= '0' and $frase[$i] <= '9') continue ;
        if ($frase[$i] == ' ') continue ;
        if ($frase[$i] == '-') continue ;
        if ($frase[$i] == ',') continue ;

        // Se chegamos aqui, i-ésimo caractere está fora da convenção.
        // Frase está errada.
        echo "Frase incorreta!<br>";
        return ;
    }

    // Neste ponto, único potencial erro da frase é o último caractere.
    // Comparamos este último caractere a ponto-final, exclamação e interrogação.
    $ultimoChar = $frase[strlen($frase) - 1];
    if ($ultimoChar != "." and $ultimoChar != "!" and $ultimoChar != "?") {
```

```

        // Não é nenhum dos caracteres esperados. Erro.
        echo "Frase incorreta!<br>";
    }
    else {

        // Último caractere também passou no teste. Frase está correta.
        echo "Frase correta!<br>";
    }
}

```

Questão 2: Suponha que você trabalhe em uma empresa que está conduzindo uma pesquisa sobre nomes populares no Brasil. Sua empresa obteve dados sobre os nomes de cada indivíduo brasileiro e sua tarefa é processar estes dados de forma a extrair estatísticas relevantes à pesquisa. Em particular, você recebe a tarefa de determinar os 5 nomes mais comuns dentre todos os brasileiros. Para isso, você resolve escrever um programa em PHP que realize este processamento.

Escreva uma função em PHP que receba como parâmetro um **vetor** em que cada posição (numeradas sequencialmente a partir da posição 0) contém o primeiro nome de cada brasileiro. Sua função deverá **determinar e imprimir** os 5 nomes que mais ocorrem.

Solução: uma possível solução é dada pela função a seguir:

```

function cincoMaisComuns($nomes) {

    // Contar ocorrências dos nomes
    $freq = array();
    for ($i = 0; $i < count($nomes); $i++) {

        // Usamos um vetor associativo chamado $freq para
        // contar o número de ocorrências de cada nome.
        if (!array_key_exists($nomes[$i], $freq)) {

            // Primeira ocorrência do nome: inicializar.
            $freq[$nomes[$i]] = 1;
        }
        else {

            // Esse nome já ocorreu antes: atualizar contador.
            $freq[$nomes[$i]]++;
        }
    }

    // Achar os 5 nomes mais comuns. A ideia é começar pelo mais comum

```

```

// (elemento que mais ocorreu), imprimi-lo, zerar seu número de
// ocorrências, e repetir o processo para o próximo.
for ($i = 0; $i < 5; $i++) {

    // Achar o i-ésimo nome mais comum.
    $maisComum = "";
    $maisComumContagem = 0;
    foreach ($freq as $nome => $contagem) {

        if ($contagem > $maisComumContagem) {

            $maisComum = $nome;
            $maisComumContagem = $contagem;
        }
    }

    // Imprimir nome encontrado.
    echo ($i + 1) . "o nome mais comum: " . $maisComum . "<br>";

    // Zerar sua frequência para encontrarmos o próximo na próxima iteração.
    $freq[$maisComum] = 0;
}
}

```

Questão 3: Suponha que uma tabela denominada *funcionario* contém duas colunas, a saber, *nome* e *salario*. Como você escreveria um script PHP para imprimir o nome e salário do quinto funcionário mais bem pago?

Solução:

```

...
$query = "SELECT nome, salario FROM funcionario ORDER BY salario desc";

if (mysql_num_rows(mysql_query($query)) >= 5) {

    mysql_fetch_seek(mysql_query($query), 5);

    $row = mysql_fetch_array();

    echo "O funcionário $row['nome'] é quinto mais bem pago com um salário de
    $row['salario']";
}
...

```

Questão 4: Uma companhia mantém uma lista de preços de produtos armazenada em uma tabela chamada PRODUTO dentro de um banco de dados chamado BANCO e que contém as seguintes colunas: CODIGO (um número inteiro), NOME (uma string) e PRECO (um número decimal). Escreva em PHP uma mini-aplicação consistindo de dois módulos:

- a) O programa consulta.php mostra um formulário com apenas um campo, correspondente ao código de um produto. Uma vez que o usuário preenche o formulário e o submete, o programa carrega o programa *resposta.php*.

Solução mínima para consulta.php:

```
<form action="resposta.php" method="POST"> <!-- ou GET -->

    <input name="codigo" />

    <input type="submit" />

</form>
```

- b) O programa resposta.php consulta o banco de dados pelo produto correspondente ao código e, caso o encontre, exibe seu nome e preço. Caso o produto não se encontre no banco, exibe a mensagem "não encontrado".

Possível solução para resposta.php:

```
<?php

$conn = mysql_connect("localhost","root"," ");
mysql_select_db("banco", $conn);

$sql = 'SELECT * FROM PRODUTO WHERE CODIGO = '.$_REQUEST['codigo'];

$result = mysql_query($sql);

if(mysql_num_rows($result) == 0) echo "não encontrado";

//Deverá fazer apenas uma iteração, mas não há problema
while($row = mysql_fetch_array($result)){
    echo "Nome do produto: " . $row[0] . "<br/>";
    echo "Preço do produto: " . $row[1] . "<br/>";
}

?>
```