



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação
Disciplina Programação de Aplicações Web
Profs. Cristina Nader e Daniel de Oliveira
Gabarito da AD1 1º semestre de 2017.

Questão 1. [1,0 ponto] Responda sucintamente às seguintes questões (atenção: **use suas próprias palavras!** Cópias da internet não serão consideradas!):

- a. Por que PHP é considerada uma linguagem "fracamente tipada"?

R: Porque nela, os tipos são definidos em tempo de execução; conforme valores são atribuídos a variáveis, esse tipo interno varia, diferindo de linguagens fortemente tipadas, em que se exige que tipos sejam definidos na declaração das variáveis.

- b. Dentre as características apresentadas da linguagem PHP, quais você considera as mais importantes? Por que?

R: A resposta do aluno pode variar bastante nessa questão. Eu particularmente aprecio muito a flexibilidade, simplicidade e objetividade do uso do PHP. Não mais do que uma linha de código é necessária para enviar um e-mail, várias funções utilitárias dadas de forma "nativa" na configuração padrão da maioria dos servidores web com interpretador PHP.

Questão 2. [3,0 pontos] Considere um tabuleiro de um jogo chamado "ache a soma". Esse jogo consiste em encontrar sequências de números na horizontal ou vertical que somados sejam iguais a um valor pré-escolhido. Implemente na linguagem PHP uma função chamada **acheSequencia(\$tabuleiro,\$soma)** que permita encontrar as sequências de números que somados sejam iguais a \$soma. Nesta função, \$tabuleiro representa um array bidimensional de números. O resultado devem ser exibir na tela todas as sequências de números encontradas com a soma desejada, separando as verticais das horizontais. A seguir apresentamos um exemplo de tabuleiro e o resultado para valor de soma 9.

1	3	6	7	2	5
4	4	6	2	2	4
2	3	5	1	1	5
5	3	5	1	5	7
6	2	4	2	1	3
3	2	3	1	7	4

Exemplo de Saída:

Horizontal: 3,6 - 7,2 - 5,4 - 3,5,1 - 3,2,3,1

Vertical: 6,3 - 5,4 - 7,2 - 4,5

ERRATA: A saída “5,4” está em vermelho pois esta só seria possível se as linhas/colunas da tabela fossem consideradas contíguas. Como essa possibilidade resultaria em dezenas de possibilidades não cobertas no exemplo de saída da questão, por facilidade, o gabarito tratará cada linha/coluna isoladamente no cômputo da soma. Ainda assim, a saída mais apropriada de exemplo para a questão teria sido:

Horizontal: 3,6 - 7,2 - 3,5,1 - 3,5,1 - 2,4,2,1 - 3,2,3,1

Vertical: 6,3 - 5,4 - 7,2 - 2,1,5,1 - 5,4 - 4,5

R: Uma das possíveis implementações é:

```
<?php
```

```
function acheSequencia($tabuleiro,$soma) {
    echo '<p>Horizontal: ' . formataArray(varreArrayContando($tabuleiro,
    $soma), true) . '</p>';
    echo '<p>Vertical: ' . formataArray(varreArrayContando(
    transpor($tabuleiro), $soma)) . '</p>';
}
```

```
function formataArray($matriz) {
    $string = '';
    foreach($matriz as $arr) {
        foreach($arr as $a)
            $string .= $a . ',';
        $string = rtrim($string, ',') . ' - ';
    }
}
```

```

    }
    return rtrim($string, ' - ');
}

function varreArrayContando($matriz, $soma) {
    $arrays_validos = array();

    foreach($matriz as $linha) {
        $fila = array(); //uso de uma fila eh muito pertinente
        foreach($linha as $elemento) {

            //insiro o elemento na fila
            array_unshift($fila, $elemento);

            if(array_sum($fila) == $soma) {
                //encontrando soma = $soma, adiciono na saida
                $arrays_validos []= array_reverse($fila);
            } else if(array_sum($fila) > $soma) {

                //retiro elementos ate que ela tenha soma <= $soma
                while(array_sum($fila) > $soma) array_pop($fila);

                if(array_sum($fila) == $soma) {
                    $arrays_validos []= array_reverse($fila);
                }
            }
        }
    }
    return $arrays_validos;
}

//Magica: resultado da chamada de null na funcao
//array_map para todos os elementos
//eh a transposicao da matriz de entrada
function transpor($matriz) {
    array_unshift($matriz, null);
    return call_user_func_array('array_map', $matriz);
}

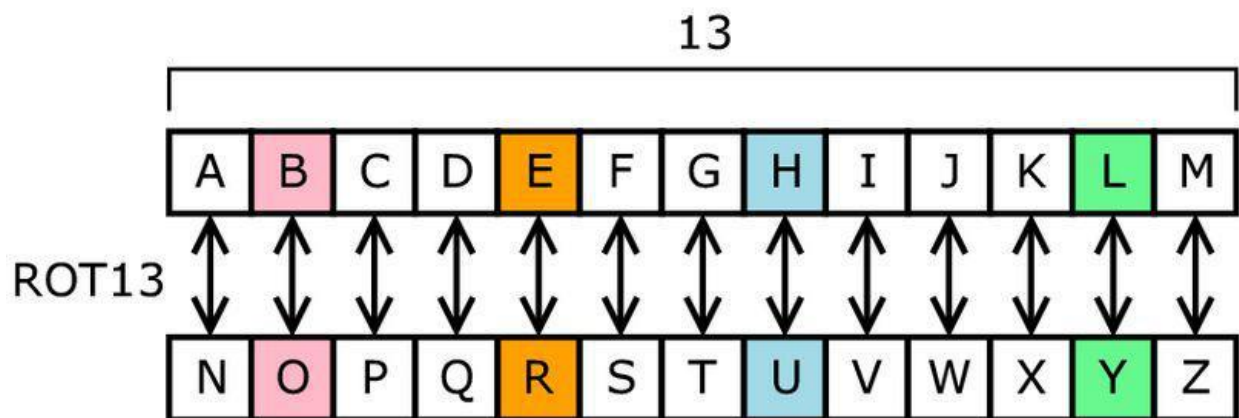
$tabuleiro = array(
    array(1, 3, 6, 7, 2, 5),
    array(4, 4, 6, 2, 2, 4),
    array(2, 3, 5, 1, 1, 5),
    array(5, 3, 5, 1, 5, 7),
    array(6, 2, 4, 2, 1, 3),
    array(3, 2, 3, 1, 7, 4));

```

```
$soma = 9;
```

```
acheSequencia($tabuleiro,$soma);
```

Questão 3. [2,0 pontos] Uma tarefa muito importante em sistemas de informação atualmente é a segurança do sistema. Em geral, a segurança envolve criptografia, protocolos de segurança, etc. Vamos implementar nesse exercício um algoritmo de criptografia bem simples, chamado de substituição simples. Uma substituição simples pode ser expressa escrevendo o alfabeto em uma ordem diferente, que se designa **alfabeto de substituição**. Em geral, ele é deslocado de um passo fixo ou embaralhado de forma mais complexa. Nesse exercício utilizaremos a cifra ROT13, conforme apresentado a seguir, onde cada letra é substituída por sua "equivalente":



Implemente uma função em PHP chamada **cifraTexto(\$texto)** que recebe um texto como parâmetro de entrada e gera como saída o texto cifrado de acordo com a substituição de caracteres. Por exemplo, se o texto "Meu nome é Daniel" fosse dado como entrada, a saída gerada seria "ZRH ABZR é QNAVRY".

R:

```
<?php
```

```
function cifraTexto($texto) {  
    $texto = strtoupper($texto);  
    $A = ord('A');  
    $Z = ord('Z');  
    $saida = '';  
    foreach(str_split($texto) as $char) {  
        if(ord($char) > $Z || ord($char) < $A)  
            $saida .= ' '  
        else if(ord($char) >= ($A + 13))  
            $saida .= chr(ord($char) - 13);  
        else  
            $saida .= chr(ord($char) + 13);  
    }  
}
```

```

        return $saida;
    }

    $texto = "Meu nome é Daniel";

    echo cifraTexto($texto);

```

Questão 4. [4,0 pontos] O sítio Hortiflix (<http://hortiflix.com.br/>) foi criado como publicidade para um famoso mercado do Rio de Janeiro. Nesse sítio nomes de filmes famosos são reformulados de forma que façam referência a alguma fruta, legume ou verdura (produto vendido pelo mercado). Por exemplo, “O Poderoso Melão” em referência ao “O Poderoso Chefão” e os “Milhons” em referência aos “Minions”. Como nem todas as pessoas conseguem entender de primeira a qual filme a propaganda se refere resolvemos pedir sua ajuda nessa tarefa. Escreva uma função em PHP chamada **comparacaoFilmes** que recebe dois arrays de **strings** contendo títulos de filmes. Cada **array** pode conter quantidades diferentes de **strings**. O primeiro **array** recebe o nome dos filmes originais e o segundo as versões do sítio Hortiflix. Sua função deve analisar o quão parecidas são as letras contidas em posições correspondentes para cada par de títulos de filmes fornecidos (um de cada **array**) e calcular um valor total que indica o índice de similaridade entre os títulos. É importante ressaltar que os títulos dos filmes nem sempre têm o mesmo comprimento (**strings** de mesmo tamanho). Ao fim, sua função deve apresentar os 3 títulos (original e modificado) que mais se assemelham, ou seja, que tem os maiores índices de similaridade. Utilize a regra de pontuação a seguir para comparar cada letra do primeiro título com a letra correspondente no segundo título.

1. Letras Iguais – 10 pontos
2. Mesma letra, mas em um título ela é maiúscula e no outro minúscula – 5 pontos
3. Letras diferentes – 2 pontos
4. Letra em uma **string** e número ou outro símbolo na outra – 0

Exemplo:

Comparação entre “Minions” e “Milhons”) -> calcula 10+10+2+2+10+10+10 = 54

R:

```
<?php
```

```

function comparacaoFilmes($filmes, $candidatos) {

    $tabela_filmes = array();
    $similaridades = array();

    for($i = 0; $i < count($filmes); $i++) {
        $texto1 = $filmes[$i];
        for($j = 0; $j < count($candidatos); $j++) {
            $texto2 = $candidatos[$j];
            $similaridade = comparaFilmes($texto1, $texto2);
            $similaridades[] = $similaridade;
            $tabela_filmes[] = array($filmes[$i],
                                    $candidatos[$j], $similaridade);
        }
    }
}

```

```

    }

    //Ordena tabela por uma coluna, estilo banco de dados
    array_multisort ($similaridades, SORT_DESC, $tabela_filmes);

    for($i = 0; $i < min(count($tabela_filmes), 3); $i++)
        echo ($i+1) . '): "' . $tabela_filmes[$i][0] . '" e "'
            . $tabela_filmes[$i][1] . '"<br />';
}

function comparaFilmes($texto1, $texto2) {

    $texto1 = str_split($texto1);
    $texto2 = str_split($texto2);

    $pontuacao = 0;

    for($i = 0; $i < min(count($texto1), count($texto2)); $i++) {
        $c1 = ord($texto1[$i]);
        $c2 = ord($texto2[$i]);
        if($c1 == $c2) {
            $pontuacao += 10;
        } else if(ord(ucfirst($texto1[$i])) ==
ord(ucfirst($texto2[$i]))) {
            $pontuacao += 5;

            } else if(($c1 == 195 || $c2 == 195) //195 usado para
char acentuado na funcao ord
                || (ctype_alpha($texto1[$i]) &&
ctype_alpha($texto2[$i]))) {
                $pontuacao += 2;
            }
        }

    return $pontuacao;
}

$filmes = array("O Silêncio dos Inocentes", "Minions", "O Poderoso
Chefão", "Avatar", "O Diabo veste Prada");
$candidatos = array("O Poderoso Melão", "Milhons", "O Rei Melão",
"O quiabo veste Prada");

comparacaoFilmes($filmes, $candidatos);

```