

Curso de Tecnologia em Sistemas de Computação
Disciplina Programação de Aplicações Web
Profs. Cristina Nader e Daniel de Oliveira
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"?
Porque em PHP as variáveis são declaradas sem a necessidade de se definir um tipo para estas e durante a execução do código as variáveis podem ser inicializadas com qualquer valor de qualquer tipo primitivo ou estruturado e podem assumir qualquer outro valor de qualquer outro tipo durante a execução.
- b) Dentre as características apresentadas da linguagem PHP, quais você considera as mais importantes? Por que?
Uma característica importante é a fácil integração com bancos de dados mysql devido a funções nativas da linguagem, fato que permite construir aplicações para web que interagem de forma mais interessante com o usuário.

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.

RESPOSTA:

```
<?php
//VARIABLES
$k = 0;
$j = 0;
$i = 0;
$soma = 9;
$pontuacao = 0;
$desloc = -1;
$horizontal = array();
$vertical = array();
$temp = array();
$pos = 0;
$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));

//FUNÇÃO
function acheSequencia($tabuleiro, $soma){
    global $k, $j, $i, $pontuacao, $desloc, $horizontal, $vertical, $temp, $pos;
    //horizontal
    for($i=0; $i<sizeof($tabuleiro); $i++){
```

```

while ($j< sizeof($tabuleiro)){
    $temp[$k] = $tabuleiro[$i][$j];
    $pontuacao+=$temp[$k];
    if($pontuacao == $soma){
        if (!in_array($temp, $horizontal)){
            $horizontal[$pos] = $temp;
            $pos+=1;
        }
        $pontuacao = 0;
        $temp = array();
        $k = -1;
    }else{
        if($pontuacao>$soma){
            $j = $desloc;
            $desloc+=1;
            $pontuacao = 0;
            $temp = array();
            $k = -1;
        }
    }
    $j+=1;
    $k+=1;
}
$j=0;
$desloc = -1;
if($k>$soma){
    $k =0;
}
}
echo "horizontal: ";
for($i=0; $i<sizeof($horizontal);$i++){
    for($j=0; $j<sizeof($horizontal[$i]); $j++){
        echo $horizontal[$i][$j]," ";
    }
    echo "-";
}
//Vertical
$i = 0;
$temp = array();
$desloc = -1;
$k = 0;
$pontuacao = 0;
$pos =0;
for($j=0; $j<sizeof($tabuleiro); $j++){
    while ($i< sizeof($tabuleiro)){
        $temp[$k] = $tabuleiro[$i][$j];
        $pontuacao+=$temp[$k];
        if($pontuacao == $soma){
            if (!in_array($temp, $vertical)){

```

```
$vertical[$pos] = $temp;
$pos+=1;
}
$pontuacao = 0;
$temp = array();
$k = -1;
}else{
    if($pontuacao>$soma){
        $i = $desloc;
        $desloc+=1;
        $pontuacao = 0;
        $temp = array();
        $k = -1;
    }
}
$i+=1;
$k+=1;
}
}
$i=0;
$desloc = -1;
if($k>$soma){
    $k = 0;
}
}
echo "\n", "vertical: ";
for($i=0; $i<sizeof($vertical);$i++){
    for($j=0; $j<sizeof($vertical[$i]); $j++){
        echo $vertical[$i][$j], " ";
    }
    echo "-";
}
}
//FIM FUNÇÃO
echo(acheSequencia($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".

RESPOSTA:

```

<?php

$Letras = array('a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','w','x','y','z');
(string)$saida = "";
$texto = "meu nome é daniel";
function cifraTexto($texto){
    global $Letras, $saida;
    for($i=0; $i<strlen($texto); $i++){
        strtolower($texto[$i]);
        if(in_array($texto[$i],$Letras)){
            $saida .= str_rot13($texto[$i]);
        }else{
            $saida .= $texto[$i];
        }
    }
    return $saida;
}
echo(strtoupper(cifraTexto($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 parecido são as letras contidas em posições correspondente 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

RESPOSTA:

```

<?php

$nomesOriginais = array("O Poderoso Chefao", "Minions", "Rei Leao", "Tropa de elite",
"Matrix");

```

```

$nomesReformulados = array("O Poderoso Melao","Milhons", "Rei Melao","Horta de
elite", "Maçatirix");
$Letras = array('a','b','c','d','e','f','g','h','i','j','k','l','m','n','o','p','q','r','s','t','u','v','y','x','w','z');
$primNome = " ";
$segNome = " ";

```

```

class similaridade{
    var $nomeFilme = " ";
    var $nomesSimilares = array();
    var $pontuacao = array();

    function DefinirNomeFilme($nomeF){
        $this->nomeFilme = $nomeF;
    }

    function ObterNomeFilme(){
        return @$this->nomeFilme;
    }

    function DefinirNomesSimilares($nomeDS,$posDS){
        $this->nomesSimilares[$posDS] = $nomeDS;
    }

    function ObterNomesSimilares($posOS){
        return @$this->nomesSimilares[$posOS];
    }

    function DefinirPontuacao($valor,$posDP){
        $this->pontuacao[$posDP] = $valor;
    }

    function ObterPontuacao($posOP){
        return @$this->pontuacao[$posOP];
    }
}

```

```

$soma = 0;
$saida = array();

```

```

function ComparacaoFilmes(){
    global $nomesOriginais, $nomesReformulados,
        $primNome, $segNome, $soma, $saida, $Letras;
    for ($i=0; $i<sizeof($nomesOriginais); $i++){
        $saida[$i] = new similaridade();
        $primNome = $nomesOriginais[$i];
        for ($j=0; $j<sizeof($nomesReformulados); $j++){
            $segNome = $nomesReformulados[$j];
            for ($k=0; $k<strlen($primNome); $k++){
                if($k>strlen($segNome)){

```

```

        break;
    }
    if((@$primNome[$k] == @$segNome[$k]) and
    (($primNome[$k] <> " ") and ($segNome[$k] <> " ")){
        $soma = $soma+10;
    }else{
        if (((@$primNome[$k] == strtolower(@$segNome[$k])) or
        (@$primNome[$k] == strtoupper(@$segNome[$k]))) and
        (($primNome[$k] <> " ") and ($segNome[$k] <> " ")){
            $soma = $soma+5;
        }
    }
    if(@$primNome[$k] <> @$segNome[$k]){
        if(in_array(strtolower(@$primNome[$k]),$Letras)
        and in_array(strtolower(@$segNome[$k]),$Letras)){
            $soma = $soma+2;
        }else{
            $soma = $soma+0;
        }
    }
} //for k
}
$saida[$i]->DefinirNomeFilme($primNome);
$saida[$i]->DefinirNomesSimilares($segNome, $j);
$saida[$i]->DefinirPontuacao($soma, $j);
$soma = 0;
} //for j
} //for i
for ($i=0; $i<sizeof($saida); $i++){
    echo "[ $i - ", $saida[$i]->ObterNomeFilme(), "]\n";
    for($j=0; $saida[$i]->ObterPontuacao($j) != null; $j++){
        echo $saida[$i]->ObterNomesSimilares($j), " ";
        echo $saida[$i]->ObterPontuacao($j), "\n";
    }
}
}
echo (ComparacaoFilmes());
?>

```