



1)	2,0	
2)	3,5	
3)	1,0	
4)	3,5	

**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 AD1 – 1º Semestre de 2018**

**Estão sendo avaliados o uso das estruturas básicas de repetição e condição, a criação e uso de funções e manipulação de vetores na linguagem PHP.**

1) Considere o seguinte algoritmo simples para o cálculo de uma aproximação da raiz quadrada de um número  $n$ :

- Inicialize as variáveis *soma*, *incremento* e *raiz*, todas com o valor 1.
- Se *soma* é maior ou igual a  $n$ , retorne o valor de *raiz* e finalize o algoritmo.
- Incremente a variável *incremento* em duas unidades.
- Incremente a variável *raiz* em uma unidade.
- Atualize a variável *soma* para seu valor atual somado a *incremento*.
- Volte ao passo b)

Escreva uma função em PHP que implemente esse algoritmo.

**R: De forma comentada:**

```
<?php
//a)
$soma = $incremento = $raiz = 1;
//b) e f)
while(!($soma >= $n)) {
//c)
    $incremento += 2;
```

```
//d)
$raiz++;
//e)
$soma += $incremento;
//f)
}
//b)
return $raiz;
```

2) Suponha que você faça parte de um grupo que está realizando uma pesquisa sobre a desigualdade social no Brasil. Em particular, sua tarefa é determinar qual percentual das pessoas mais ricas detém 50% da riqueza total do país. Suponha que seu grupo já possua uma lista com todos os habitantes do Brasil e seu respectivo patrimônio em reais.

Escreva uma função em PHP que recebe como parâmetro um vetor **ordenado de forma decrescente** em que cada posição corresponde a um habitante e contém o valor, em reais, do patrimônio dessa pessoa. Sua função deverá processar esse vetor e retornar **o menor valor numérico X** tal que a soma dos patrimônios dos X% mais ricos (em relação à população total) seja **maior ou igual** à metade do somatório dos patrimônios de toda a população.

**R:**

```
function x_mais_ricos($ordenados) {
    $ordenados_mais_ricos = $x = 0;
    $populacao_total = count($ordenados);
    $soma_total = array_sum($ordenados);
    while($ordenados_mais_ricos < $soma_total/2)
        $ordenados_mais_ricos += $ordenados[$x++];
    return $x / $populacao_total;
}

$ordenados = array(15000, 9000, 8234, 2155, 1200, 900, 800, 800,
800, 150);
$x = x_mais_ricos($ordenados);
echo 'x = '. $x;
```

3) Embora o calendário possua tipicamente 365 dias, a Terra demora aproximadamente 365,2422 dias para dar uma volta completa em torno do Sol. A

existência dessa parte fracionária, faz com que o ano de 365 dias não corresponda exatamente ao tempo exato de translação (o que deveria determinar a duração de um ano). Por esse motivo, foram criados os anos bissextos: como regra geral, a cada 4 anos adiciona-se um dia a mais ao calendário para compensar a diferença de 0,2422 dias nos anos normais. Um fato menos conhecido, no entanto, é que essa regra não é tão simples. De fato, para compensar esse descasamento, a regra completa é:

- Anos múltiplos de 400 sempre são bissextos.
- Dos demais, anos múltiplos de 100 nunca são bissextos.
- Dos demais, anos múltiplos de 4 sempre são bissextos.

Escreva uma função em PHP que receba como argumento um ano e retorne o seu número de dias de acordo com essas regras.

R:

```
function bissexto($ano) {  
    return ($ano % 400 == 0) ||  
           (!($ano % 100 == 0) && ($ano % 4 == 0));  
}  
  
echo bissexto(2010) . ' - 2010<br />';  
echo bissexto(4) . ' - 4<br />';  
echo bissexto(100) . ' - 100<br />';  
echo bissexto(400) . ' - 400<br />';  
echo bissexto(2400) . ' - 2400<br />';  
echo bissexto(1996) . ' - 1996<br />';  
echo bissexto(1995) . ' - 1995<br />';  
echo bissexto(2010) . ' - 2010<br />';
```

4) Escreva uma função em PHP que receba como argumento uma string contendo uma expressão numérica (por exemplo, “3x4+1”) e retorne o valor resultante. Sua função, portanto, deverá identificar as constantes numéricas da expressão, bem como os sinais das operações e realizar o cálculo. Considere que:

- a) Apenas as quatro operações aritméticas básicas devem ser suportadas, sendo representadas pelos caracteres ‘+’, ‘-’, ‘x’ e ‘/’.
- b) Apenas valores inteiros deverão ser suportados (mas note que os valores podem ter múltiplos algarismos).
- c) Nenhum outro caractere deve aparecer nas expressões, além dos algarismos decimais e dos símbolos das operações. Caso haja outros caracteres na string recebida, **sua função deverá imprimir um erro e abortar a execução**.

- d) Sua função deverá ignorar as regras habituais de precedência das operações. Isto é, as operações deverão ser realizadas na ordem em que aparecem na expressão, independentemente do seu tipo (exemplo, sua função deve retornar o resultado 12 para a expressão “1+2x4”, ao invés do valor 9 que seria o resultado se considerássemos a precedência da multiplicação sobre a soma).

**R:**

```
function processa_conta($conta) {
    //opcao 2: if(!ereg('\d[+\-x\/]\d([+\-x\/]\d)*', $conta))
    if(!preg_match('/\d[+\-x\/]\d([+\-x\/]\d)*/', $conta))
        die('Entrada inválida');

    list($operando1, $conta) = consome_numero($conta);

    while(strlen($conta) > 0) {
        list($operador, $conta) = consome_operacao($conta);
        list($operando2, $conta) = consome_numero($conta);
        $operando1 = realiza_operacao($operando1, $operador,
$operando2);
    }

    return $operando1;
}

function consome_numero($str) {
    preg_match("~^(\d+)~", $str, $match);
    return array($match[1], substr($str, strlen($match[1])));
}

function consome_operacao($str) {
    preg_match("~^([+\-x\/])~", $str, $match);
    return array($match[1], substr($str, strlen($match[1])));
}

function realiza_operacao($operando1, $operador, $operando2) {
    switch($operador) {
        case '+':
            return $operando1 + $operando2;
        case '-':
```

```
        return $operando1 - $operando2;
    case 'x':
        return $operando1 * $operando2;
    case '/':
        return $operando1 / $operando2;
    }
    die('Entrada inválida');
}
```

```
echo '33 x 11 + 2 / 5<br />';
echo processa_conta('33x11+2/5');
echo ' <br />aaa + 2 <br />';
echo processa_conta('aaa+2');
```