

## Programação II

### Avaliação Presencial 1 - GABARITO

#### Parte I - PHP

- 1) Escreva um programa para validar a entrada do usuário utilizando expressões regulares. Assuma que as variáveis contendo os dados de entrada são:

a) \$email → endereço de email.

RESPOSTA:

```
<html>
<head>
<title> AP1: questão 1a </title>
</head>
<body>
<?php
    echo "Resposta à questão 1a:<br><br>";
    if (! ( empty($_POST["email"]) ) ) {
        $exprvalida = "^([0-9a-z~!#$%&_-]([.]?[0-9a-z~!#$%&_-])*)".
            "@([0-9a-z~!#$%&_-]([.]?[0-9a-z~!#$%&_-])*)$";
        echo "Usuário entrou com o email: ".
            $_POST["email"]."<br>";
        if ( ! eregi($exprvalida, $_POST["email"]) )
            echo "Email inválido.<br>";
        else
            echo "Obrigado. Email válido.";
    }
    else {
?>
        <form method="post" action="1a.php">
        Digite o email, por favor:
        <input type="text" name="email" value="" size="20"
            maxlength="20">
    <?php
    }
?>
</body>
</html>
```

b) \$placa → Placa de carro (três caracteres alfabéticos maiúsculos seguidos por 4 algarismos).

RESPOSTA:

```
<html>
<head>
<title> AP1: questão 1b </title>
</head>
<body>
<?php
    echo "Resposta à questão 1b:<br><br>";
    if (! ( empty($_POST["placa"]) ) ) {
        $exprvalida = "^([a-z]{3}[0-9]{4})$";
        echo "Usuário entrou com a placa: ".

```

```

        $_POST["placa"]."<br>";
        if ( ! eregi($exprvalida, $_POST["placa"]) )
            echo "Placa inválida.<br>";
        else
            echo "Obrigado. Placa válida.";
    }
    else {
?>
        <form method="post" action="1b.php">
        Digite o número da placa, por favor:
        <input type="text" name="placa" value="" size="7"
maxlength="7">
<?php
    }
?>
</body>
</html>

```

c) \$senha → Senha de usuário (mínimo de 8 caracteres, com pelo menos um algarismo ou um caractere não alfanumérico).

RESPOSTA:

```

<html>
<head>
<title> AP1: questão 1c </title>
</head>
<body>
<?php
    echo "Resposta à questão 1c:<br><br>";
    if ( ! ( empty($_POST["senha"]) ) ) {
        $exprvalida1 = "^ [0-9a-z~!#$%&_-]{8,}";
        $exprvalida2 = "^\.[0-9~!#$%&_-].*$";
        echo "Usuário entrou com a senha: ";
        $_POST["senha"]."<br>";
        if ( eregi($exprvalida1, $_POST["senha"]) &&
            eregi($exprvalida2, $_POST["senha"]) )
            echo "Obrigado. Senha válida.";
        else
            echo "Senha inválida. Senha deve conter ".
                "pelo menos um algarismo ou um caractere não ".
                "alfanumérico.<br>";
    }
    else {
?>
        <form method="post" action="1c.php">
        Digite a senha, por favor:
        <input type="password" name="senha" value="" size="20"
maxlength="20">
<?php
    }
?>
</body>
</html>

```

2) Assuma o seguinte vetor:

```

$produto['frios']['salame'] = 4.50;
$produto['frios']['presunto'] = 3.50;
$produto['confeitaria']['bolo'] = 20;

```

```
$produto['padaria']['paofrances'] = 0.30;
$produto['padaria']['brioche'] = 0.40;
```

a) Ordene este vetor pela chave e diga qual vai ser a nova organização de elementos.

RESPOSTA:

```
<html>
<head>
<title> AP1: questão 2a </title>
</head><pre>
<body>
<?php
    $produto['frios']['salame'] = 4.50;
    $produto['frios']['presunto'] = 3.50;
    $produto['confeitaria']['bolo'] = 20;
    $produto['padaria']['paofrances'] = 0.30;
    $produto['padaria']['brioche'] = 0.40;

    echo "Resposta à questão 2a, AP1<br><br>";
    echo "Array sem ordenação: <br>";
    foreach ($produto as $categoria) {
        foreach ($categoria as $i => $preco)
            echo "$preco<br>";
    }
    print_r($produto);

    ksort($produto);

    echo "Array com ordenação por chave de primeira dimensão:
<br>";
    foreach ($produto as $categoria) {
        foreach ($categoria as $i => $preco)
            echo "$preco<br>";
    }
    print_r($produto);
?>
</body></pre>
</html>
```

b) Ordene este vetor por valor e diga qual vai ser a nova organização de elementos.

RESPOSTA:

```
<html>
<head>
<title> AP1: questão 2b </title>
</head><pre>
<body>
<?php
    $produto["frios"]["salame"] = "4.50";
    $produto["frios"]["presunto"] = "3.50";
    $produto["confeitaria"]["bolo"] = "20";
    $produto["padaria"]["paofrances"] = "0.30";
    $produto["padaria"]["brioche"] = "0.40";

    // $iproduto["salame"]["frios"] = 4.50;
    // $iproduto["presunto"]["frios"] = 3.50;
```

```
// $Iproduto["confeitaria"]["bolo"] = "20";
// $Iproduto["padaria"]["paofrances"] = "0.30";
// $Iproduto["padaria"]["brioche"] = "0.40";

// array_multisort($Iproduto);

// print_r($Iproduto);

echo "Resposta à questão 2b, AP1<br><br>";
echo "Array sem ordenação: <br>";
foreach ($produto as $categoria) {
    foreach ($categoria as $i => $preco)
        echo "$preco<br>";
}
print_r($produto);

echo "Array com ordenação por valor: <br>";

array_multisort($produto);
foreach ($produto as $categoria) {
    foreach ($categoria as $i => $preco)
        echo "$preco<br>";
}
echo "Note que a ordenação realizada por array_multisort faz
uma ordenação por chave de primeira dimensão,<br> depois por
segunda dimensão e assim por diante. <br>Os valores de preço
não são ordenados neste array, <br>porque todas as chaves de
segunda dimensão são distintas. <br> Ou seja, se o array
tivesse sido definido da forma:<br><br>\$produto[salame][frios]
= 4.50;<br>\$produto[presunto][frios] = 3.50;
<br>\$produto[confeitaria][bolo] = 20; <br>
\$produto[padaria][paofrances] = 0.30;<br>
\$produto[padaria][brioche] = 0.40;<br><br> os valores 4.5 e
3.5 seriam ordenados porque a chave de segunda dimensão é a
mesma.";
print_r($produto);
?>
</body></pre>
</html>
```

- c) Escreva um pequeno trecho de código que reproduza os elementos deste vetor em uma tabela de categorias.

RESPOSTA:

SOLUÇÃO 1:

```
<html>
<head>
<title> AP1: questão 2c </title>
</head><pre>
<body>
<?php
    $produto['frios']['salame'] = 4.50;
    $produto['frios']['presunto'] = 3.50;
    $produto['confeitaria']['bolo'] = 20;
    $produto['padaria']['paofrances'] = 0.30;
    $produto['padaria']['brioche'] = 0.40;

    echo "Resposta à questão 2c, AP1<br><br>";
    echo "<table border=1>";
    list($chave) = each($produto);
```

```

        foreach ($produto as $categoria) {
            foreach ($categoria as $i => $preco) {
                $chave = key($produto);
                echo "<tr><td> $chave </td>
                    <td> $i: </td>
                    <td> R\$ $preco</td></tr>";
            }
            list($chave) = each($produto);
        }
    ?>
</body></pre>
</html>

```

SOLUÇÃO 2:

```

<html>
<head>
<title> AP1: questão 2c </title>
</head><pre>
<body>
<?php
    $produto['frios']['salame'] = 4.50;
    $produto['frios']['presunto'] = 3.50;
    $produto['confeitaria']['bolo'] = 20;
    $produto['padaria']['paofrances'] = 0.30;
    $produto['padaria']['brioche'] = 0.40;

    echo "Resposta à questão 2c, AP1<br><br>";
    echo "<table border=1>";
    while (list($key, $value) = each($produto)) {
        while (list($key2, $value2) = each($value)) {
            echo "<tr><td>$key</td>
                <td>$key2</td>
                <td>$value2</td></tr>";
        }
    }
    ?>
</body></pre>
</html>

```

- 3) Escreva uma função que retorne o somatório de elementos de um vetor. Se o usuário não passar parâmetros, o resultado retornado deve ser nulo.

```

<html>
<head>
<title> AP1: questão 3 </title>
</head>
<body>
<?php
function soma($v)
{
    if ( ! isset($v) ) return 0;
    foreach ($v as $i => $valor)
    {
        $soma += $valor;
    }
    return $soma;
}

```

```

        echo "Resposta à questão 3:<br><br>";
        // inicializar um array com 100 elementos atribuídos
        aleatoriamente
        for ( $i = 0; $i < 100; $i++ ) {
            $array[] = rand(1,1000);
        }
        // escrever valores do array
        echo "Valores do vetor: <br>";
        foreach ( $array as $i => $valor )
        {
            echo "$valor ";
        }
        echo "<br><br>";
        $soma = soma($array);
        echo "Soma dos valores do vetor = $soma<br>";
        $soma = @soma();
        echo "Soma dos valores do vetor quando a função soma é
        chamada sem parâmetros = $soma<br>";
        ?>
    </body>
</html>

```

## Parte II – MySQL

- 4) Considere as relações T1 e T2 listadas abaixo. Escreva o resultado das seguintes consultas:
- select nome, grau from T1 inner join T2 where T1.cor = T2.cor
  - select nome, grau from T1 left join T2 where T1.cor = T2.cor
  - select nome, grau from T1 right join T2 where T1.cor = T2.cor

T1

<i>nome</i>	<i>cor</i>
Joao	preto
Pedro	rosa
Maria	vermelho
Rosa	verde

T2

<i>cor</i>	<i>grau</i>
vermelho	2
azul	4
verde	5
vermelho	6

RESPOSTA:

a)

<i>nome</i>	<i>grau</i>
Maria	2
Maria	6
Rosa	5

b)

<i>nome</i>	<i>grau</i>
Joao	NULL
Pedro	NULL
Maria	2
Maria	6
Rosa	5

c)

<i>nome</i>	<i>grau</i>
Maria	2
NULL	4
Rosa	5
Maria	6

- 5) Considere que você deseja montar um banco de dados de receitas culinárias. Cada receita tem um nome (string), um texto de descrição (string) e uma lista de ingredientes. Cada ingrediente tem um nome (string) e uma quantidade (decimal). Pede-se:
- Escreva um esquema para seu banco de dados.
  - Que consulta SQL poderia ser usada para saber os nomes das receitas que usam ovos?
  - Que consulta SQL poderia ser usada para saber quantas receitas não usam margarina?

RESPOSTA:

- a) 

```
create table receita (id int not null,
                      nome varchar(40),
                      descricao varchar (120),
                      primary key (id));
create table ingrediente (id int not null,
                          nome varchar (40),
                          primary key (id));
create table receita_ingrediente (idreceita int not null,
                                  idingrediente int not null,
                                  quantidade decimal);
```
- b) 

```
select receita.nome
from receita, ingrediente, receita_ingrediente
where receita.id = idreceita
and ingrediente.id = idingrediente
and ingrediente.nome = 'ovo';
```
- c) 

```
select (select count(*) from receita) -
(select count(*)
 from receita, ingrediente, receita_ingrediente
 where receita.id = idreceita
 and ingrediente.id = idingrediente
 and ingrediente.nome = 'margarina')
```