

Questão 1:

```
<?php
if (!isset($_POST["nome"])) {
?>
<html>
<head><title>Informações para Cadastro</title></head>
<body bgcolor="white">
<form method="post" action="questao_1.php">
<h1> Informações para Cadastro </h1>
<h3>Por favor, entre com a informação solicitada para se cadastrar. Campos mostrados em
<font color="red"> vermelho </font> são obrigatórios.</h3>
<table>
<col span="1" align="right">
<tr> <td><font color="red">Nome:</font></td>
    <td><input type="text" name="nome" size=50></td></tr>
<tr> <td><font color="red">Sobrenome:</font></td>
    <td><input type="text" name="sobrenome" size=50></td></tr>
<tr> <td><font color="red">Endereço:</font></td>
    <td><input type="text" name="endereco" size=50></td></tr>
<tr> <td><font color="red">Email/username:</font></td>
    <td><input type="text" name="email" size=50></td></tr>
<tr> <td><input type="submit" value="Submit"></td></tr>
</table></form></body>
</html>
<?php
} // fecha if
else {
    include 'error.inc';
    include 'clean.inc';

    $validaemail = "^([0-9a-z~!#%&_-])([.]?[0-9a-z~!#%&_-])*$".
        "@([0-9a-z~!#%&_-])([.]?[0-9a-z~!#%&_-])*$";
    $validastring = "^[A-Z].*";

    $errorString = "";
    foreach($_POST as $varname => $value)
        $formVars[$varname] = clean($value, 50);

    if (empty($formVars["nome"]))
        $errorString .= "\n<br>O campo nome deve ser digitado.";

    elseif (!ereg($validastring, $formVars["nome"], $partes))
        $errorString .= "\n<br>O nome não está no formato correto.";

    if (empty($formVars["sobrenome"]))
        $errorString .= "\n<br>O campo sobrenome deve ser digitado.";

    elseif (!ereg($validastring, $formVars["sobrenome"], $partes))
        $errorString .= "\n<br>O nome não está no formato correto.";

    if (empty($formVars["endereco"]))
        $errorString .= "\n<br>Você deve digitar pelo menos uma linha de endereço.";

    if (empty($formVars["email"]))
        $errorString .= "\n<br>O campo email é obrigatório.";

    elseif (!ereg($validaemail, $formVars["email"], $partes))
        $errorString .= "\n<br>O email não está no formato correto.";

    if (!empty($errorString))
    {
```

```

    // Há erros. Mostrar e sair.
?>
<html>
<head><title>Erro no cadastramento</title></head>
<body bgcolor="white">
<h1>Erro no Cadastramento</h1>
<?=$errorString?>
<br><a href="questao_1.php">Retornar para o form</a>
</body>
</html>

<?php
    exit;
}

```

Questão 2:

Ao termino do código da questão 1, escrever:

```

if (!$con = @ mysql_pconnect($ _SERVER["REMOTE_ADDR"], "aluno", "aluno"))
    die("Não pôde conectar ao BD");

if (!mysql_select_db("prog2", $con))
    ExibeErro();

// Criar uma consulta com o dado do usuário
$createTableCadastro = "create table if not exists cadastro (id int key auto_increment,
sobrenome varchar(100), nome varchar(50), endereco varchar(100), email varchar(80));"

if (!(@mysql_query($createTableCadastro,$con)))
    ExibeErro();

$verificador_de_email = "SELECT * FROM cadastro WHERE email = \"\" . $formVars["email"] .
\"\"";
$i = 0;
$verifica_email = mysql_query($verificador_de_email, $con);
while (mysql_fetch_row($verifica_email)){
    $i++;
}
if($i == 0){
    $consulta = "INSERT INTO cadastro
        set id = 0, \" .
        "sobrenome = \"\" . $formVars["sobrenome"] . "\", \" .
        "nome = \"\" . $formVars["nome"] . "\", \" .
        "endereco = \"\" . $formVars["endereco"] . "\", \" .
        "email = \"\" . $formVars["email"] . \"\"";
    // Rodar a consulta
    if (!(@ mysql_query ($consulta, $con)))
        ExibeErro();

    echo ("Registro inserido com sucesso !!!</b>");
} else {
    echo ("Este email já está cadastrado !!!</b></font>");
}

// Id do novo membro
    $ID = mysql_insert_id();
} // end else
?>

```

Questão 3:

Os array multidimensionais em php são definidos da seguinte forma:

`array["tipo1"]["tipo2"]...["tipon"]`, podendo ter quantos índices forem necessários. Os tipos dos arrays podem ser numéricos ou associativos. No caso de arrays numéricos, temos uma matriz de valores numéricos. No caso de uma array associativo, podemos ter, para cada tipo, uma associação de objetos a cada índice do vetor, como `$a["movei"]["mesa"]`, onde `$a["movei"]` é um vetor e `["mesa"]` é um índice do vetor `$a["movei"]`.

A ordenação de vetores multidimensionais pode ser feita pelos valores dos vetores, ou pelas chaves. Para ordenar vetores pelos valores, podemos usar as funções `sort` (chaves numéricas) que modificam as chaves, ou `asort` (chaves alfanuméricas) que não modificam as chaves. Para ordenar vetores pelas chaves podemos usar a função `ksort`. A ordenação funciona da mesma forma que em arrays unidimensionais. A ordenação pode ser feita utilizando o comando `foreach` dentro de outro `foreach`, onde percorremos cada índice do vetor fazendo as operações necessárias.

Questão 4:

Este comando header permite enviar cabeçalhos html para o cliente. Deve ser chamado na primeira linha do arquivo .php. O seguinte exemplo mostra o que não se deve fazer:

```
<html>
<? header("Location: http://www.php.net"); ?>
</html>
```

Este código causará um erro, já que tentou-se enviar um header depois de ter sido enviado um texto ("`<html>\n`").

O comando header é utilizado para diversos propósitos, entre os mais comuns são:

//redirecionar para outra página:

```
header("Location: http://www.php.net");
```

// Definir o script como uma mensagem de erro:

```
header("http/1.0 404 Not Found");
```

// Definir a expiração da página:

```
header("Cache-Control: no-cache, must-revalidate"); // HTTP/1.1 header("Pragma: no-cache");
// HTTP/1.0
```

Exmplo: index.php

```
<?php
if(isset($_POST['url'])){
    if(ereg("^http://", $_POST['url']) && $_POST['url']!="")
        header("Location: ".$_POST['url']);
    else{
        header("http/1.0 404 Not Found");
        echo "Error! Url não válida <br/>";
    }
}
?>
```

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
<title>Teste Headers</title>
<form action='index.php' method='post'>
Ir: <input name="url" size="80" type="text">
<input type="submit">
</form>
</html>
```

Questão 5:

Locks são fechaduras, isto é, no mysql podemos imaginar que existe uma porta de entrada ao banco de dados.

Os locks são úteis ou necessários quando se está salvando dados.

Então, locks são variáveis especiais que permitem restringir o acesso ao banco de dados.

Existe lock de leitura, o qual permite que múltiplos usuários possam ler o banco de dados e apenas um usuário possa escrever; e lock de escrita o qual restringe que apenas um usuário possa manipular o banco de dados, neste caso pode-se pensar como se um usuário entra no banco de dados e tranca a porta, não deixando entrar mais ninguém, e destranca a porta apenas quando ele fez todas as operações que tinha que fazer.

exemplo: reajuste de preços em 5%

```
LOCK TABLES produtos WRITE;
```

```
UPDATE produtos SET preco = preco*1.05;
```

```
UNLOCK TABLES;
```

Questão 6:

Método 1.

// arquivo: a.php

```
<?
    $login="";
    if (isset($_POST['login'])){
        $login = $_POST['login'];
        // Enviar um novo valor para o cookie
        setcookie("login", $login);
        header("Location: b.php");
    }
}
```

?>

```
<html>
<head>
    <title> Acesso </title>
</head>
<body>
<form enctype="multipart/form-data" action="a.php" method="post">
Usuario
<br><input type="text" name="login" size=20>
<br><input type="submit" value="Login">
<form>
</body>
</html>
```

//arquivo: b.php

```
<?
    $login = "";
    if (!isset($_COOKIE['login'])){
        header("Location: a.php");
    }
    $login = $_COOKIE['login'];
```

?>

```
<html>
<body>
<? echo "Bem vindo, $login!"; ?>
</body>
</html>
```

Método 2.

```
// arquivo: a.php
<?
session_start();
if (!isset($_SESSION["login"])){
    if (isset($_POST['user'])){
        $_SESSION["login"] = $_POST['user'];
        header("Location: b.php");
    }
} else
    header("Location: b.php");
?>

<html>
<head>
    <title> Acesso </title>
</head>
<body>
<form enctype="multipart/form-data" action="c.php" method="post">
Usuario
<br><input type="text" name="user" size=20>
<br><input type="submit" value="Login">
</form>
</body>
</html>

//arquivo: b.php
<?
require("e.php");
session_start();
if (isset($_POST['Sair'])){
    session_destroy();
    header("Location: a.php");
}
?>

<html>
<body>
<? echo "Bem vindo, ".$_SESSION["login"]."!"; ?>
<form enctype="multipart/form-data" action=" b.php" method="post">
<input type="submit" name = "Sair" value="Sair">
</form>
</body>
</html>
```