

```
<?php

echo "LISTA DE EXERCÍCIOS 01" . "<br />"
."NOME 01;_____ " . "<br />"
."NOME 02;_____ " . "<p>";

/*****
* Exercício 01 *
Escreva uma função que receba 2 arrays (1 de notas, 1 de pesos) e mostre a média.
Para array(7, 8, 10) e array(2, 3, 5) o resultado deve ser 8.8.
*****/
echo "Exercício 01: " . "<br />";

function mediaDeArray($notas, $pesos) {
    $resultado = 0;

    for($x=0; $x<sizeof($notas); $x++) {
        $resultado += ($notas[$x] / 10) * $pesos[$x];
    }

    return $resultado;
}

echo mediaDeArray(array(7, 8, 10), array(2, 3, 5));

echo "<hr />";

/*****
* Exercício 02 *
Escreva uma função que retorne o array passado como parâmetro, de trás para frente.
*****/
echo "Exercício 02: " . "<br />";

function inverteArray($array) {
    $novoArray = array();

    for($x=sizeof($array) - 1; $x>=0; $x--) {
        $novoArray[sizeof($array) - 1 - $x] = $array[$x];
    }

    return $novoArray;
}

print_r(inverteArray(array("a", "b", "c", "d")));

echo "<hr />";

/*****
* Exercício 03 *
Escreva uma função que retorne o tamanho do array, sem utilizar as funções sizeof ou count.
*****/
echo "Exercício 03: " . "<br />";

function tamanhoDeArray($array) {
    $x = 0;
```

```
        foreach($array as $posicao) {
            $x++;
        }
        return $x;
    }
}

echo tamanhoDeArray(array(1, 2, 3));

echo "<hr />";

/*****
* Exercício 04 *
Construa uma versão própria da função array_push sem utilizar a função em seu
código.
*****/
echo "Exercício 04: " . "<br />";

function meu_array_push($array, $elemento) {

    /*
    * Para inserir um elemento no final do array basta informar a chave do novo
    * valor entre colchetes (neste caso foi utilizada uma chave numérica) e
    * atribuir nesta posição o valor ou elemento em questão.
    */
    $array[sizeof($array)] = $elemento;
    return $array;
}

$x = array(1, 2, 3);
$x = meu_array_push($x, 4);
print_r($x);

echo "<hr />";

/*****
* Exercício 05 *
Construa uma versão própria da função array_pop sem utilizar a função em seu
código.
*****/
echo "Exercício 05: " . "<br />";

function meu_array_pop($array) {
    $elemento = $array[sizeof($array) - 1];
    unset($array[sizeof($array) - 1]);
    return $elemento;
}

$x = array(1, 2, 3);
echo meu_array_pop($x) . "<br />";
print_r($x);

echo "<hr />";

/*****
* Exercício 06 *
Construa uma versão própria da função array_shift sem utilizar a função em seu
```

```

código.
*****/
echo "Exercício 06: " . "<br />";

function meu_array_shift($array) {
    $elemento = $array[0];
    unset($array[0]);
    return $elemento;
}

$x = array(1, 2, 3);
echo meu_array_shift($x) . "<br />";
print_r($x);

echo "<hr />";

/*****
 * Exercício 07 *
Construa uma versão própria da função array_unshift sem utilizar a função em seu
código.
*****/
echo "Exercício 07: " . "<br />";

function meu_array_unshift($array, $elemento) {

    /*
     * Antes de inserir o elemento na primeira posição, todos os atuais elementos
     * devem ser reposicionados. Por isto o comando for é montado invertido (do
     * fim para o início) reposicionando todos os elementos primeiramente.
     */
    for($x=sizeof($array); $x>0; $x--) {
        $array[$x] = $array[$x-1];
    }

    $array[0] = $elemento;
    return $array;
}

$x = array(1, 2, 3);
$x = meu_array_unshift($x, 4);
print_r($x);

echo "<hr />";

/*****
 * Exercício 08 *
Construa uma versão própria da função array_key_exists sem utilizar a função em seu
código.
*****/
echo "Exercício 08: " . "<br />";

function minha_array_key_exists($chaveBusca, $array) {
    $chaves = array_keys($array);

    foreach($chaves as $chave) {
        if($chave == $chaveBusca) {

```

```
        return TRUE;
    }
}
return FALSE;
}
$x = array("Componente" => "DSI", "Professor" => "Roberto");
echo minha_array_key_exists("Componente", $x);
echo minha_array_key_exists("DSI", $x);
```

?>