

ALGORITMOS E LINGUAGENS DE PROGRAMAÇÃO

PROFESSOR JEFFERSON CHAVES

jefferson.chaves@ifpr.edu.br

ATIVIDADE AVALIATIVA

A proposta da atividade é formalizar os conceitos estudados acerca do uso de repetições e listas no PHP.

Avaliação:

Os passos realizados para execução dos algoritmos deverão ser registrados por meio dos espaços correspondentes. SIGA AS INSTRUÇÕES DO PROFESSOR!

Para esta atividade, cada detalhe será avaliado (representações dos valores em memória, comando de pular linha, espaços, traços e etc.).

Exercício 1

```
$numeros[3] = [1, 22, 333];
```

```
for($i=0; $i < 3; $i++){  
    print(numeros[$i], "-");  
}
```

memória

valor de i

Console

Exercício 2

```
$sete_maravilhas[7] = [  
    "Cristo Redentor",  
    "Chichén Itzá",  
    "Coliseu",  
    "Muralha da China",  
    "Petra",  
    "Machu Picchu",  
    "Taj Mahal"  
];
```

```
for($i=0; $i < 7; $i++){  
    print($i+1, ": ", $sete_maravilhas[$i], "\n");  
}
```

memória



valor de i

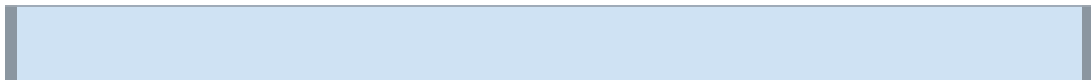
Console



Exercício 3

```
$sete_maravilhas[7] = [  
    "Cristo Redentor",  
    "Chichén Itzá",  
    "Coliseu",  
    "Muralha da China",  
    "Petra",  
    "Machu Picchu",  
    "Taj Mahal"  
];  
  
$valor_buscado = "Coliseu";  
  
for($i=0; $i < 7; $i++){  
    if ($sete_maravilhas[$i] == $valor_buscado) {  
        print("o valor buscado está entre as maravilhas");  
        break;  
    }  
}  
}
```

memória




valor de i

Console

Exercício 4

```
$mega_sena[6] = [33, 4, 47, 7, 11, 25];  
$aux = 0;  
  
for($i=0; $i < 6; $i++){  
    for($j=0; $j < 5; $j++){  
  
        if($mega_sena[$j] > $mega_sena[$j + 1]){  
            $aux = $mega_sena[$j]  
            $mega_sena[$j] = $mega_sena[$j + 1]  
            $mega_sena[$j + 1] = $aux  
        }  
    }  
}  
  
for($i=0; $i < 6; $i++){  
    print($mega_sena[$i], " - ")  
}  
}  
memória
```



valor de **i**

Console



5. Escreva um algoritmo que leia uma matriz de inteiros com 3 linhas e 3 colunas, com valores já inicializados e mostre-os na tela, separando as linhas.



6. Escreva um algoritmo que leia uma matriz de inteiros com 3 linhas e 3 colunas, com valores já inicializados, e mostre a soma na tela.



7. Escreva um algoritmo que leia uma matriz de inteiros com 3 linhas e 5 colunas com valores informados pelo usuário. Ao final, mostre: quais são os números pares digitados e a posição (linha, coluna) em que eles foram armazenados.

8. Ao se executar o código abaixo, o que será mostrado no console?

```
$lista_frutas = ['banana', 'abacaxi', 'melão', 'morango'];

print "\nTeste 0: ";
var_dump(count($lista_frutas));

print "\nTeste 1: ";
var_dump(in_array('banana', $lista_frutas));

print "\nTeste 2: ";
var_dump(in_array('alface', $lista_frutas));

print "\nTeste 3: ";
var_dump(array_search('morango', $lista_frutas));

print "\nTeste 4: ";
var_dump(sort($lista_frutas));

print "\nTeste 5: ";
var_dump(shuffle($lista_frutas));
```

9. Ao se executar o código abaixo, o que será mostrado no console?

```
$pessoa = [  
    'nome' => 'John Doe',  
    'idade' => 54,  
    'altura' => 1.8,  
    'formacao' => 'Físico',  
    'profissao' => 'Astronauta',  
];  
  
print "Teste 0: ";  
var_dump(count($pessoa));  
  
print "\nTeste 1: ";  
var_dump(in_array('Físico', $pessoa));  
  
print "\nTeste 2: ";  
var_dump(in_array('Detetive', $pessoa));  
  
print "\nTeste 3: ";  
var_dump(array_search('nome', $pessoa));  
  
print "\nTeste 4: ";  
var_dump(array_key_exists('profissao', $pessoa));  
  
print "\nTeste 5: ";  
var_dump(array_key_first($pessoa));  
  
print "\nTeste 6: ";  
var_dump(array_key_last($pessoa));  
  
print "\nTeste 7: ";  
var_dump(sort($pessoa));
```

10. Manipulação de parâmetros por referência

Dizemos que em PHP, os parâmetros são passados “por valor”, o que significa que uma cópia do valor da variável é passada para a função. No entanto, você pode simular a passagem por referência usando o operador `&`. Quando você passa um parâmetro com `&`, você está permitindo que a função modifique diretamente a variável original.

Aqui está um exemplo simples em PHP:

```
<?php
// Função que modifica o valor da variável original
function modificarPorReferencia(&$x) {
    $x = $x * 2;
}

// Variável original
$valor = 5;
echo "Valor antes da função: " . $valor . "\n";

// Chamando a função com a variável como argumento por
referência
modificarPorReferencia($valor);

echo "Valor após a função: " . $valor . "\n";
```


Neste exemplo, a função `modificarPorReferencia` recebe um parâmetro por referência (`&$x`), o que significa que qualquer alteração feita em `$x` dentro da função afetará diretamente a variável `$valor` fora da função. Ao chamar a função, você verá que o valor de `$valor` é modificado de acordo.

Essa abordagem permite que você altere o valor original da variável sem precisar retornar um novo valor da função. No entanto, tenha em mente que o uso excessivo de passagem por referência pode tornar o código mais difícil de entender e manter, então é geralmente recomendável usá-lo com cuidado.

Atividade:

Programa uma função em PHP que receba uma string e um caractere como parâmetros. A função deve **modificar a string original**, removendo todas as ocorrências desse caractere. O número de vezes que o caractere aparece na string deve ser contado. Tanto a String quanto o caractere devem ser recebidos como referência.