



Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação
Disciplina Programação de Aplicações Web
Profs. Cristina Nader e Daniel de Oliveira
AD1 2º semestre de 2016.

Nome: _____

Observações Importantes

1. Prova COM consulta.

Atenção: Como a avaliação à distância é individual, caso seja constatado que provas de alunos distintos são cópias umas das outras, independentemente de qualquer motivo, a todas será atribuída a nota ZERO. As soluções para as questões podem sim, ser buscadas por grupos de alunos, mas a redação final de cada prova tem que ser individual.

ADs enviadas pelo correio devem ser postadas cinco dias antes da data final de entrega estabelecida no calendário de entrega de ADs.

Questão 1. [1,0 ponto] Escreva um programa em PHP que identifique e imprima em uma página HTML todos os números primos entre 1 e 1000 separados por um espaço em branco. Após o último número primo deve haver um ponto final na *string*.

Gabarito

```
<?php
$limiteInferior = 1;
$limiteSuperior = 1000;
$strImpressao = "Números primos entre ".$limiteInferior." e ".$limiteSuperior." : ";
$contador = $limiteInferior;
while ($contador <= $limiteSuperior) {
    $primo = true;
    for($contadorPrimo = 2; $contadorPrimo < $contador; $contadorPrimo++){ // Varredura por
    todos os números menores que ele mesmo
        if ($contador % $contadorPrimo == 0) { // Se o número for divisível por algum menor
        que ele, não é primo
```

```

        $primo = false; // Marca o número como não-primo
    }
}
if ($primo) {
    $strImpressao .= $contador." "; // Adiciona o número primo na string
}
$contador++;
}
echo $strImpressao.".";
?>

```

Questão 2. [2,0 pontos] A Profa. Cristina Nader está elaborando uma planilha para controle de presença de seus alunos em sua disciplina de Computação Gráfica na UFF. A Profa. Cristina leciona todas segundas, quartas e sextas de 10:00 as 12:00 e precisa marcar a presença dos alunos em cada um dos dias entre 01/02/2016 e 03/06/2016. Logo, ela precisa que seja criada uma lista de todos os dias em que ela leciona para que seu controle seja facilitado. Vamos ajudar a Profa. Cristina? Escreva um programa em PHP que exiba uma *string* contendo todos os dias de aula da Profa. Cristina com o seguinte formato: “Aulas de Computação Gráfica: segunda-feira 18/07, quarta-feira 20/07, sexta-feira 22/07....” As datas devem ser impressas no formato segunda-feira OU quarta-feira OU sexta-feira DD/MM. Ao fim da *string* deve haver um ponto final e não uma vírgula!

Gabarito

```

<?php
$inicio = strtotime('01-02-2016');
$fim = strtotime('03-06-2016');
$qtdeSegundosDia = 24*60*60;
$stringTituloDias = "Aulas de Computação Gráfica:";

$aux = $inicio;
while ($aux <= $fim) {
    $diaDaSemana = date('w', $aux);
    if ($diaDaSemana==1){ // Significa que o dia é uma segunda-feira
        $stringTituloDias .= ' segunda-feira '.date('d/m',$aux).','; // Imprime o dia em questão
        $aux += $qtdeSegundosDia*2; // Avança dois dias na data para a próxima data válida
    } else if ($diaDaSemana==3){ // Significa que o dia é uma quarta-feira
        $stringTituloDias .= ' quarta-feira '.date('d/m',$aux).','; // Imprime o dia em questão
        $aux += $qtdeSegundosDia*2; // Avança cinco dias na data para a próxima data válida
    } else if ($diaDaSemana==5){ // Significa que o dia é uma sexta-feira
        $stringTituloDias .= ' sexta-feira '.date('d/m',$aux).','; // Imprime o dia em questão
        $aux += $qtdeSegundosDia*3; // Avança três dias na data para a próxima data válida
    }
}
echo $stringTituloDias.".";

```

```
}  
}  
echo rtrim($stringTituloDias,',').'.':;  
?>
```

Questão 3. [4,0 pontos] Considere uma turma de Programação de Aplicações Web do CEDERJ. Os professores da disciplina desejam calcular o desempenho da turma para fins estatísticos seguindo alguns critérios. Dado o cenário a apresentado, faça o que se pede a seguir:

- a. Escreva uma função PHP que recebe um *array* como parâmetro. Esse *array* descreve os pares [nomeDoAluno] => notaFinalDisciplina de uma determinada turma. Sua função PHP deve calcular a média de todos os alunos dessa turma **[1,0 ponto]**.

Gabarito

```
function getMediaTurma($arrayNotas) {  
    $somaNotaAlunos = 0;  
    $qtdeAlunos = 0;  
    foreach ($arrayNotas as $aluno => $nota){  
        $somaNotaAlunos += $nota;  
        $qtdeAlunos++;  
    }  
    if ($qtdeAlunos > 0){  
        $mediaNotaAlunos = $somaNotaAlunos / $qtdeAlunos;  
        return $mediaNotaAlunos;  
    } else return "Média da Turma: Sem aluno(a)s na turma\n";  
}  
  
// Exemplo de uso  
  
$notaFinalDisciplina = array(  
    "Daniel" => 1,  
    "Pedro" => 4,  
    "Cristina"=> 8,  
);  
  
echo nl2br("Média da turma: ".getMediaTurma($notaFinalDisciplina)."\n");
```

- b. Escreva uma função PHP que recebe dois *arrays* como parâmetros. Esses *array* descrevem os pares [nomeDoAluno] => notaFinalDisciplina e [nomeDoAluno] => [sexoDoAluno] (onde M significa MASCULINO e F significa FEMININO) de uma determinada turma. Sua função PHP deve calcular a média de todos os alunos dessa turma por sexo de aluno **[1,0 ponto]**.

Gabarito

```

function getMediaTurmaPorSexo($arrayNotas, $arraySexo){
    $somaNotaAlunosM = 0;
    $somaNotaAlunosF = 0;
    $qtdeAlunosM = 0;
    $qtdeAlunosF = 0;
    foreach ($arrayNotas as $aluno => $nota){
        $sexo = $arraySexo[$aluno];
        if ($sexo == "M"){
            $somaNotaAlunosM += $nota;
            $qtdeAlunosM++;
        } else {
            $somaNotaAlunosF += $nota;
            $qtdeAlunosF++;
        }
    }
    if (($qtdeAlunosM > 0) && ($qtdeAlunosF > 0)){
        $mediaNotaAlunosM = $somaNotaAlunosM / $qtdeAlunosM;
        $mediaNotaAlunosF = $somaNotaAlunosF / $qtdeAlunosF;
        return "Média da turma (MASCULINO): ".$mediaNotaAlunosM."\n"."Média da turma (FEMININO): ".$mediaNotaAlunosF;
    }
    if (($qtdeAlunosM > 0) && ($qtdeAlunosF == 0)){
        $mediaNotaAlunosM = $somaNotaAlunosM / $qtdeAlunosM;
        return "Média da turma (MASCULINO): ".$mediaNotaAlunosM."\n"."Média da turma (FEMININO): 0";
    }
    if (($qtdeAlunosM == 0) && ($qtdeAlunosF > 0)){
        $mediaNotaAlunosF = $somaNotaAlunosF / $qtdeAlunosF;
        return "Média da turma (MASCULINO): 0"."\\n"."Média da turma (FEMININO): ".$mediaNotaAlunosF;
    }
}
// Exemplo de uso

$notaFinalDisciplina = array(
    "Daniel" => 1,
    "Pedro" => 4,
    "Cristina"=> 8,
);

$sexoDoAluno = array(
    "Daniel" => "M",
    "João" => "M",
    "Cristina" => "F",
);

```

```
echo nl2br(getMediaTurmaPorSexo($notaFinalDisciplina,$sexoDoAluno));
```

```
?>
```

- c. Escreva uma função PHP que recebe três *arrays* como parâmetros. Um primeiro *array* descrevendo os pares [nomeDoAluno] => notaFinalDisciplina, o segundo *array* descrevendo os pares [nomeDoAluno] => [sexoDoAluno] (onde M significa MASCULINO e F FEMININO) e o terceiro *array* descrevendo os pares [nomeDoAluno] => [presencaSuficienteInsuficiente] (onde I marca INSUFICIENTE e S a presença SUFICIENTE). Sua função PHP deve calcular a média de todos os alunos dessa turma por sexo de aluno e apresentar o resultado somente daqueles que tiveram presença SUFICIENTE. **[1,0 ponto]**

Gabarito

```
function getMediaTurmaPorSexo($arrayNotas, $arraySexo, $arrayFrequencia){
    $somaNotaAlunosM = 0;
    $somaNotaAlunosF = 0;
    $qtdeAlunosM = 0;
    $qtdeAlunosF = 0;
    foreach ($arrayNotas as $aluno => $nota){
        $sexo = $arraySexo[$aluno];
        $frequencia = $arrayFrequencia[$aluno];
        if ($frequencia == "S"){
            if ($sexo == "M"){
                $somaNotaAlunosM += $nota;
                $qtdeAlunosM++;
            } else {
                $somaNotaAlunosF += $nota;
                $qtdeAlunosF++;
            }
        }
    }
    if (($qtdeAlunosM > 0) && ($qtdeAlunosF > 0)){
        $mediaNotaAlunosM = $somaNotaAlunosM / $qtdeAlunosM;
        $mediaNotaAlunosF = $somaNotaAlunosF / $qtdeAlunosF;
        return "Média da turma (MASCULINO): ".$mediaNotaAlunosM."\n"."Média da turma (FEMININO): ".$mediaNotaAlunosF;
    }
    if (($qtdeAlunosM > 0) && ($qtdeAlunosF == 0)){
        $mediaNotaAlunosM = $somaNotaAlunosM / $qtdeAlunosM;
        return "Média da turma (MASCULINO): ".$mediaNotaAlunosM."\n"."Média da turma (FEMININO): 0";
    }
    if (($qtdeAlunosM == 0) && ($qtdeAlunosF > 0)){
```

```

        $mediaNotaAlunosF = $somaNotaAlunosF / $qtdeAlunosF;
        return "Média da turma (MASCULINO): 0"."\\n"."Média da turma (FEMININO):
".$mediaNotaAlunosF;
    }
}

// Exemplo de uso

$notaFinalDisciplina = array(
    "Daniel" => 1,
    "Pedro" => 4,
    "Cristina"=> 8,
);

$sexoDoAluno = array(
    "Daniel" => "M",
    "João" => "M",
    "Cristina" => "F",
);

$presencaSuficienteInsuficiente = array(
    "Daniel" => "S",
    "João" => "S",
    "Denise" => "S",
);

echo nl2br(getMediaTurmaPorSexo($notaFinalDisciplina,$sexoDoAluno,$
presencaSuficienteInsuficiente));

?>

```

- d. Escreva uma função PHP que receba um *array* descrevendo os pares [nomeDoAluno] => notaFinalDisciplina, um nome de aluno e uma nota. Sua função deve atualizar a nota do aluno no *array* recebido como parâmetro realizando a média da nova nota com a presente no *array*. Caso o aluno não esteja contido no *array*, deve ser adicionado com a nota informada. **[1,0 ponto]**

Gabarito

```

function atualizaNotaAluno($nomeAluno, $notaAluno, &$notaFinalDisciplina){
    if (array_key_exists($nomeAluno, $notaFinalDisciplina)){
        $notaFinalDisciplina[$nomeAluno] = ($notaFinalDisciplina[$nomeAluno] +
$notaAluno)/2;
    } else {
        $notaFinalDisciplina[$nomeAluno] = $notaAluno;
    }
}

```

}

Questão 4. [3,0 pontos] Vamos relembrar um pouco da disciplina de estruturas de dados? Considere que desejamos implementar uma lista encadeada composta por números inteiros sem repetição em PHP, utilizando como base um *array* global nativo da linguagem (chame este *array* de \$lista). Relembre seu funcionamento e implemente as seguintes funções de forma prover todas as possibilidades listadas a seguir (é permitido uso de funções nativas da linguagem para simplificar a implementação):

- `inserir($elemento)`: mapeia o *array* global e insere um elemento na frente da lista.
- `inserir_fim($elemento)`: insere o novo elemento como último elemento do *array* global.
- `esta_vazia()`: indica se a lista está vazia ou se possui algum elemento com um retorno booleano.
- `remover()`: remove o primeiro elemento da lista e retorna seu valor.
- `remover_fim()`: remove o último elemento da lista e retorna seu valor.
- `buscar($elemento)`: retorna a posição em que se encontra um elemento da lista (começando da posição 0).
- `proximo($elemento)`: retorna o valor do elemento na posição seguinte ao elemento buscado.
- `anterior($elemento)`: retorna o valor do elemento na posição anterior ao elemento buscado.

Observação MUITO IMPORTANTE

Atente para esta definição: “Uma lista encadeada é uma sequência de células na qual cada célula contém um objeto de algum tipo (no nosso caso números inteiros) e o endereço da célula seguinte”. Como em nossa descrição, o *array* não deve ter elementos repetidos, é preciso se preocupar com este aspecto durante a programação. No PHP há a possibilidade de manipulação explícita de ponteiros. A representação nesse exercício deve ser, portanto, uma aproximação da estrutura original, que expresse comportamento equivalente à uma implementação de mais baixo nível, por meio do resultado das chamadas de função definidas no exercício, aplicadas à um *array* comum.

Gabarito

```
<?php
$lista = array();

function inserir($elemento) {
    global $lista;
    if (is_int($elemento) && !in_array($elemento, $lista)) {
        array_unshift($lista, $elemento);
    } else die("Entrada invalida");
}
```

```
}

function inserir_fim($elemento) {
    global $lista;
    if (is_int($elemento) && !in_array($elemento, $lista)) {
        $lista[] = $elemento; //ou array_push
    } else die("Entrada invalida");
}

function esta_vazia() {
    global $lista;
    return empty($lista);
}

function remover() {
    global $lista;
    if (!esta_vazia()) {
        return array_shift($lista);
    }
}

function remover_fim() {
    global $lista;
    if (!esta_vazia()) {
        return array_pop($lista);
    }
}

function buscar($elemento) {
    global $lista;
    if (is_int($elemento)) {
        return array_search($elemento, $lista);
    } else die("Parametro de busca invalido");
}

function proximo($elemento) {
    global $lista;
    $tamanho_lista = count($lista);
    $indice_elemento = buscar($elemento);
    if ($indice_elemento >= 0 && $indice_elemento < $tamanho_lista - 1) {
        return $lista[$indice_elemento + 1];
    }
}

function anterior($elemento) {
    global $lista;
    $indice_elemento = buscar($elemento);
    if ($indice_elemento > 0) {
```



```
        return $lista($indice_elemento - 1);  
    }  
}  
?>
```