

Gabarito AD1

Programação II

1.

```
<?php
```

```
$N = 5;
while($N>1){
echo $N.", ";
if ($N%2)
    $N=$N*($N+1);
else
    $N= (int)($N/3);

}
```

```
?>
```

Para \$N=5 a saída é 5,30,10,3,12,4, isto se consideramos apenas valores inteiros para \$N.

Sem considerar a restrição de números inteiros a saída é

5,30,10,3.33333333333333,14.444444444444,4.8148148148148,1.6049382716049,4.180765127
2672,1.3935883757557,3.3356769367972,14.462417563478,4.8208058544927,1.60693528483
09,4.1891762944655,1.3963920981552,3.3463029899453,14.544046690462,4.8480155634874,
1.6160051878291,4.2274779549198,1.4091593183066,3.3948893026769,14.920162680107,4.9
733875600357,1.6577958533452,4.4060829447139,1.4686943149046,3.6257573055378,16.77
1873344198,5.5906244480662,36.845706167381,12.281902055794,4.0939673519312,1.36465
57839771,3.2269411927192,13.640090653987,199.69216370297,40076.652408077,13358.884
136026,4452.9613786752,1484.3204595584,494.77348651947,164.92449550649,54.97483183
5496,18.324943945165,6.1083146483885,2.0361048827962

2.

```
<?php
```

```
$matriz = array(array(1, 3, 9, 0),
                  array(5, 2, 6, 4),
                  array(4, 5, 4, 1),
                  array(7, 8, 0, 2));
```

```
function soma($arr){
    $res = 0;
    foreach($arr as $val){
        $res += $val;
    }

    return $res;
}
```

```

function f($matriz, &$col, &$row){
    // criar arrays para as colunas
    $cols = array();
    for($j = 0; $j<count($matriz);$j++){
        $cols[] = array();
        for($i = 0; $i<count($matriz[$j]);$i++){
            $cols[$j][] = $matriz[$i][$j];
        }
    }

    $tmp = 0;
    // encontra a coluna com a maior soma de elementos
    $val = -1;
    foreach($cols as $c){
        $tmp = soma($c);
        if ($val < $tmp){
            $val = $tmp;
            $col = $c;
        }
    }

    // encontra a linha com a maior soma de elementos
    $val = -1;
    foreach($matriz as $l){
        $tmp = soma($l);
        if ($val < $tmp){
            $val = $tmp;
            $row = $l;
        }
    }
}

$col=null;
$row=null;
f($matriz, $col, $row);

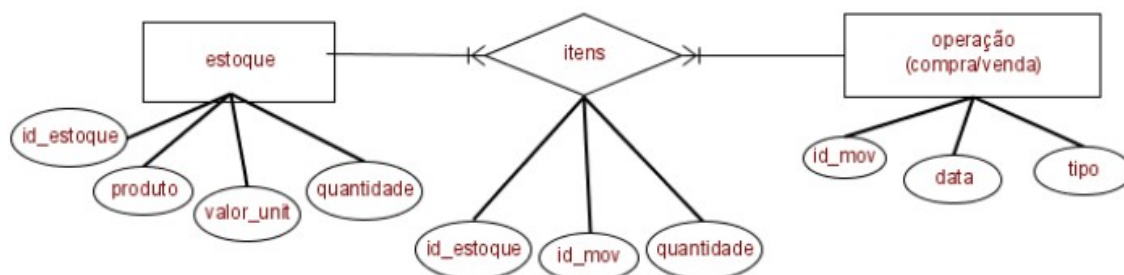
print_r($col);
print_r($row);

```

?>

3.

i.



ii.

```
CREATE TABLE `estoque` (  
  `id_estoque` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `produto` varchar(200) NOT NULL,  
  `valor_unit` decimal(10,2) unsigned NOT NULL DEFAULT '0.00',  
  `quantidade` tinyint(3) unsigned DEFAULT NULL,  
  PRIMARY KEY (`id_estoque`)  
);  
CREATE TABLE `movimentacao` (  
  `id_mov` int(10) unsigned NOT NULL AUTO_INCREMENT,  
  `data` date NOT NULL,  
  `tipo` varchar(50) NOT NULL,  
  PRIMARY KEY (`id_mov`)  
);  
CREATE TABLE `itens` (  
  `id_estoque` int(10) unsigned NOT NULL,  
  `id_mov` int(10) unsigned NOT NULL,  
  `quantidade` int NOT NULL,  
  PRIMARY KEY (`id_estoque`, `id_mov`)  
);
```

iii.

cria uma nova movimentação

```
insert into movimentacao values (8, '2011-02-01', 'compra');
```

atualiza estoque e tabela de itens

```
update estoque set estoque.quantidade = estoque.quantidade + 10  
where produto = 'Casaco P';
```

```
update estoque set estoque.quantidade = estoque.quantidade + 5  
where produto = 'Calça M';
```

```
insert into itens values (4, 6, 10);
```

```
insert into itens values (2, 6, 5);
```

ERRATA

Houve um erro de escrita na questão 3.3, a movimentação 6 se refere a uma “compra” e não uma “venda”, mas se foi considerado como uma “venda” as consultas de atualização mudam pouco.

```
insert into movimentacao values (8, '2011-02-01', 'venda');  
update estoque set estoque.quantidade = estoque.quantidade - 10  
where produto = 'Casaco P';  
update estoque set estoque.quantidade = estoque.quantidade - 5  
where produto = 'Calça M';
```