GABARITO AD1

PROGRAMAÇÃO II

SEGUNDO SEMESTRE 2010

1. (3 pontos) Considere um tabuleiro do jogo "busca palavras". Implemente na linguagem PHP uma função `achaPalavras($tabuleiro,$palavras)` que permita encontrar as palavras no tabuleiro. Observe que $tabuleiro é um array bidimensional de caracteres, e $palavras é um array de strings. O resultado deve ser o tabuleiro original no qual as posições não contendo caracteres da lista de palavras são substituídas por asteriscos. Veja o exemplo abaixo.

```
x b m g x l v f m g c
e t e t v b a r a p j
i t l g w j n z r x p
j u a c n p a o a f v
l g o c b j n q c w o
w x h p a t a h u h j
q o o m x b b o j k n
w e p j t d a n a p o
t q t a n g e r i n a
w j m m o r a n g o k
f k r u d y b m o p b
```

banana
morango
abacate
maracuja
melao
tangerina

```
* * m * * * * * m * *
e * e * * * a * a * *
* t l * * * n * r * *
* * a * * * a * a * *
* * o c * * n * c * *
* * * * a * a * u * *
* * * * * b b * j * *
* * * * * * a * a * *
* * t a n g e r i n a
* * * m o r a n g o *
* * * * * * * * * * *
```

Resp.

```
function thereIsWord($table, &$result, $word, $wsz, $n, $m, $x, $y, $dir){
   $foundmsg = "$word encontrado na posiçao ".($y+1).", ".($x+1)." <br>";
   $lx = $x+$dir[0]*$wsz;
   $ly = $y+$dir[1]*$wsz;
   if ( ( $lx < $m || $lx > 0) && ( $ly < $n || $ly > 0) ){
      $i = $x;
      $j = $y;
      $can = true;
      for($c = 0; $c<$wsz; $c++){
         if (!($table[$i][$j] == $word[$c])){
            return false;
            break;
         }
         $i += $dir[0];
         $j += $dir[1];
      }

      if ($can){
         for($c = 0; $c<$wsz; $c++){
            $result[$x][$y] = $word[$c];
            $x += $dir[0];
            $y += $dir[1];
         }
      }
   }
   echo $foundmsg;
   return true;
```

```
}

function searchWord($table, &$result, $word){
   global $dirs;

   $m = count($table);
   $n = count($table[1]);

   $wsz = strlen($word);

   $found = false;

   while (!$found) {
      for ($i = 0; $i<$m; $i++)
      for ($j = 0; $j<$n; $j++)
      foreach($dirs as $dir){
         if (thereIsWord($table, $result, $word, $wsz, $n, $m, $i, $j, $dir)){
            $found = true;
            break;
         }
      }
   }
   return $found;
}

function searchWords($table, &$result, $words){
   foreach($words as $word){
      searchWord($table, $result, $word);
   }
}


$words = array("banana", "morango","abacate", "maracuja", "melao", "tangerina");
```

2. (3 pontos) Dado um *array* bidimensional (matriz) *M* de números inteiros, escreva uma função em PHP que imprima a sub-matriz cuja soma de seus elementos é a maior entre todas as sub-matrizes de M. Note que, se *M* tem dimensões $n \times m$, pode-se construir sub-matrizes de tamanhos $n \times (m-1)$, $n \times (m-2)$, ... , $n \times 1$, $(n-1) \times m$, $(n-1) \times (m-1)$, $(n-1) \times (m-2)$, ... , $(n-1) \times 1$, ... , 1×1. Por exemplo, dada a matriz

0	-2	-7	0
9	2	-6	2
-4	1	-4	1
-1	8	0	-2

A sub-matriz cuja soma de seus elementos é a maior entre todas as sub-matrizes é

9	2
-4	1
-1	8

Resp.

```
<?php
```

```php
function submatriz($M, $ini_i, $ini_j, $fin_i, $fin_j, &$sum){
   $sub_M = array();
   for($j = $ini_j; $j<$fin_j; $j++){
      $row = array();
      for($i = $ini_i; $i<$fin_i; $i++){
         $sum += $M[$j][$i];
         $row[] = $M[$j][$i];
      }
      $sub_M[] = $row;
   }

   return $sub_M;
}

function encontramaiormatriz($M){
   $m = count($M);
   $n = count($M[0]);

   $max = 0;
   $result = array();

   for($j = 0; $j<$m; $j++){
      for($i = 0; $i<$n; $i++){
         for($J = $j+1; $J<=$m; $J++){
            for($I = $i+1; $I<=$n; $I++){
               $curr = 0;
               $sub_M = submatriz($M, $i, $j, $I, $J, $curr);
               if ($curr>$max){
                  $max = $curr;
                  $result = $sub_M;
               }
            }
         }
      }
   }

   return $result;

}

$M = array(
         array(0, -2, -7, 0),
         array(9, 2, -6, 2),
         array(-4, 1, -4, 1),
         array(-1, 8, 0, -2));

$sub_M = encontramaiormatriz($M);

echo "<pre>";
print_r($sub_M);
echo "</pre>";

?>
```

3. Considere as seguintes tabelas, **“estoque”, “movimentação” e “itens”** respectivamente. Elas constituem o banco de dados de um sistema de estoque que permite operações de compra e venda de artigos. Note que o relacionamento entre as tabelas é dado por `id_estoque` e `id_mov`.

estoque

id_estoque	produto	valor_unit	Quantidade

1	Camisa X	10.00	12
2	Calça M	55.90	5
4	Bermuda L	29.50	9

movimentacao

id_mov	data	tipo
4	10-10-2008	compra
5	12-11-2008	venda
6	01-02-2009	compra
7	02-02-2009	venda

itens

id_estoque	id_mov	quantidade
1	4	15
1	5	3
4	6	10
2	6	5
4	7	1

Pede-se:

i. (1 ponto) Criar um modelo de Entidade-Relacionamento apropriado.

ii. (1 ponto) Escrever o modelo físico correspondente.

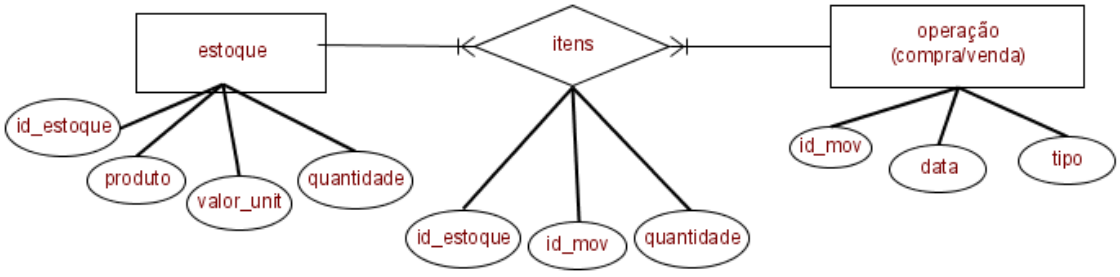
iii. (2 pontos) Escrever os comandos SQL para registrar as seguintes movimentações:

   a. venda de 4 “Camisa X” e 2 “Bermuda L”

   b. compra de 10 “Calça M”

Resp:

1.



2.

```
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`)
);
```

3.

Venda

cria uma nova movimentação

```
insert into movimentacao values (8, '2010-08-10', 'venda');
```

atualiza estoque e tabela de itens

```
update estoque set estoque.quantidade = estoque.quantidade - 4
where produto = 'Camisa X';
update estoque set estoque.quantidade = estoque.quantidade - 2
where produto = 'Bermuda L';

insert into itens values (1, 8, 4);
insert into itens values (4, 8, 2);
```

Compra

```
insert into movimentacao values (9, '2010-10-12', 'compra');
```

atualiza estoque e tabela de itens

```
update estoque set estoque.quantidade = estoque.quantidade + 10
where produto = 'Calça M';
insert into itens values (2, 9, 10);
```