

1. (3 pontos) Matrizes são muito usadas para representar sistemas lineares de equações. Se uma matriz quadrada de n linhas e n colunas representa um sistema linear, então, o sistema não é alterado quando duas linhas i e j são trocadas. Pede-se: escreva uma função que recebe um array bidimensional (matriz) e o ordena por linha. Considere que uma linha i é menor que uma linha j se $\mathbf{M}[i][k] == \mathbf{M}[j][k]$ para $0 \leq k \leq m$ e $\mathbf{M}[i][m] < \mathbf{M}[j][m]$ para algum $m \leq n$. Por exemplo: a função `ordenamatriz ($M)` ordena a matriz \mathbf{M} do sistema de equações

$$\begin{array}{rcl} x + y + 2z & = & 6 \\ 2y + z & = & 4 \\ 2x + y + z & = & 7 \end{array} \quad \mathbf{M} = \begin{bmatrix} 1 & 1 & 2 \\ 0 & 2 & 1 \\ 2 & 1 & 1 \end{bmatrix}$$

, que resulta em

$$\begin{bmatrix} 0 & 2 & 1 \\ 1 & 1 & 2 \\ 2 & 1 & 1 \end{bmatrix}$$

Resp.

```
<?php
```

```
function menor($A, $B){
    $size = sizeof($A);
    for($i = 0; $i<$size; $i++ ){
        if ($A[$i] == $B[$i]) continue;
        if ($A[$i] > $B[$i]) return true;
        if ($A[$i] < $B[$i]) return false;
    }
    return false;
}

function ordenamatriz(&$mat){
    $rows = sizeof($mat);

    for($j = 1; $j<$rows; $j++){
        for($i = 0; $i<$j; $i++){
            if (menor($mat[$i], $mat[$j])){
                $tmp = $mat[$j];
                $mat[$j] = $mat[$i];
                $mat[$i] = $tmp;
            }
        }
    }
}
```

?

2. (3 pontos) Considere que o texto de um livro, guardado numa variável string `$s`, tem seus capítulos encabeçados por “CAPÍTULO *xxx*”, onde *xxx* é um número em algarismos romanos. Escreva uma função `substituiromanos(&$s)` para alterar `$s` de tal forma a reescrever os números de capítulos em arábicos.

Resp.

```

<?php

function romano_decimal($var)
{
    $var=strtoupper($var);
    $var=ereg_replace("[^IVXLCDM]", "", $var);

    $algarismos=array("M"=>1000, "D"=>500, "C"=>100,
                     "L"=>50, "X"=>10, "V"=>5, "I"=>1);

    $max=1001;
    $tmp = 0;
    for($i=0;$i<strlen($var);$i++)
    {
        $parcial=substr($var,$i,1);
        $parcial=$algarismos[$parcial];
        if($parcial<=$max)
        {
            $tmp += $parcial;
        }else{
            $tmp +=($parcial-(2*$max));
        }
        $max=$parcial;
    }
    return $tmp;
}

function substituiromanos(&$s) {
    while(ereg("(.*)(CAPÍTULO[:space:]+)([IVXLCDM]+)(.*)", $s, $ret) >0)
    {
        $s = $ret[1].$ret[2].romano_decimal($ret[3]).$ret[4];
    }
}

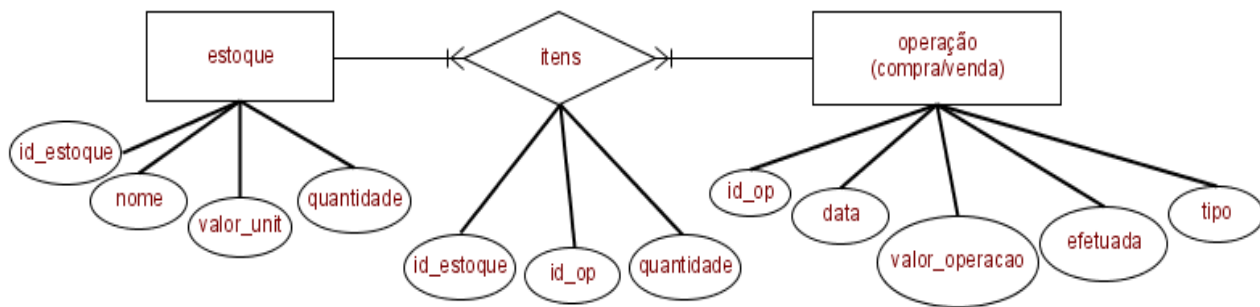
echo substituiromanos ("xxxx CAPÍTULO XXX xxxxx xxxx CAPÍTULO XXXI
xxxxx");

?>

```

3. (4 pontos) Deseja-se modelar um sistema de estoque onde a cada tipo de item corresponde uma descrição, um valor unitário e uma quantidade em estoque. Ao mesmo tempo, no banco de dados pode-se registrar compras (que agregam itens ao estoque) e vendas (que retiram itens do estoque). Compras e vendas são marcadas com um atributo lógico “efetuada” que, se verdadeiro, indica que a compra ou venda foi efetivamente realizada. Quando uma compra ou uma venda é realizada, as quantidades correspondentes devem ser atualizadas no estoque. Observe que cada pedido de compra ou venda pode conter diversos tipos de itens. Pede-se:
- Crie uma modelagem entidade-relacionamento para o banco de dados.
 - Crie um modelo físico para o banco de dados.
 - Que comandos SQL devem ser utilizados para registrar uma compra de 10 unidades do item “Camisa Xadrez”.
 - Que consulta poderia ser utilizada para abastecer o estoque com 12 unidades do item “Camisa Xadrez”

Resp a.



Resp b.

```

CREATE TABLE estoque(
    id_estoque integer not null auto_increment,
    nome varchar(50) not null,
    valor_unitario decimal,
    quantidade integer,
    primary key(id_estoque)
);
  
```

```

CREATE TABLE operacao(
    id_op integer not null auto_increment,
    data datetime not null,
    valor_operacao decimal,
    tipo varchar(1) not null,
    efetuada varchar(1) not null,
    primary key(id_op)
);
  
```

```

CREATE TABLE itens(
    id_estoque integer not null,
    id_op integer not null,
    quantidade integer not null,
    primary key(id_estoque, id_op)
);
  
```

Resp c.

Cadastramos o produto no estoque

```

INSERT INTO estoque VALUES(1, 'Camisa Xadrez', 0.0, 10);
  
```

Criamos uma movimentação de tipo compra ('C')

```

INSERT INTO operacao VALUES(1, '2008-09-20 10:15:59', 0.00, 'C', '1');
  
```

Criamos os itens da compra

```

INSERT INTO itens VALUES(1, 1, 10);
  
```

Resp d.

Atualizamos o estoque

```

UPDATE estoque SET quantidade = quantidade + 12 WHERE id = 1;
  
```