

1. (1 ponto) O que é mostrado no navegador quando uma página com código abaixo é acessada?

```
<?php
function f(&$a,$b) {
    if ($b >= sizeof($a)) {
        echo current($a);
        return;
    }
    if (current($a) < $a[$b]) {
        echo ("X");
        next($a);
    } else {
        echo ("Y");
    }
    f($a,$b+1);
}
$a = array(1,3,2,7,3);
f($a,0);
?>
```

- (a) YXYXX1  
(b) YXYXY3  
(c) **YXYYY7**  
(d) XYYYX1  
(e) Nenhuma das opções anteriores
2. (1 ponto) Qual das asserções abaixo é *incorreta*?
- (a) A linguagem PHP é usada em servidores web para gerar páginas dinâmicas .  
(b) **Numa aplicação web, SGBDs podem ser acessados diretamente através do cliente.**  
(c) SQL é uma linguagem de consultas a bancos de dados relacionais.  
(d) Uma transação é um conjunto de operações que devem ser realizadas em bloco.  
(e) SGBDs implementam transações para permitir o acesso concorrente aos dados.
3. (1 ponto) Considere o seguinte comando SQL:

```
create table cliente (
    id int default 0 not null auto_increment,
    nome varchar(50) not null,
    endereco varchar(80),
    primary key (id),
    key nomecliente(nome)
);
```

O que *não podemos* dizer desse comando:

- (a) Cria uma tabela com 3 atributos, a saber: id, nome, endereco.
  - (b) O atributo id é chave primária da tabela.
  - (c) O atributo id sempre recebe o valor 0 quando uma nova linha é inserida na tabela sem especificar um valor explícito para esse atributo.
  - (d) nomecliente é um índice que permite acessar linhas da tabela cliente dado o valor do atributo nome.
  - (e) Algumas linhas da tabela podem possuir valores nulos para o atributo endereco.
4. (1 ponto) O que é mostrado no navegador quando uma página com código abaixo é acessada?

```
<?php
foreach (array("-ab", "+1abc1", "ab123", "abc123") as $x)
{
    if (ereg ("..(a|b|c)+[123]*", $x)) {
        echo "S";
    }
    else {
        echo "N";
    }
}
?>
```

- (a) SSSS
  - (b) SNNN
  - (c) NSNS
  - (d) SSNS
  - (e) Nenhuma das opções anteriores
5. (3 pontos) Implemente a função `formulario($campos, $legendas, $defaults)`, onde `$campos` é um array de strings contendo os campos de um formulário html que deve ser apresentado via web. Para cada elemento `c` desse array, a função `formulario` deve inserir uma legenda, cujo texto está em `$legendas[c]`, isto é, `$legendas` é um array cujas chaves são elementos de `$campos`. A função também insere um `tag input` com atributo `type` igual a 'text' com atributo `name` igual a `c`. O array `$defaults` contém valores default para alguns dos campos. Em particular, se o campo `c` tem um valor default, então este deve ser o valor do atributo `value` do input correspondente. Por exemplo, a chamada

```

formulario (array("nome", "ecivil"),
              array("nome" => "Nome do Consumidor",
                    "telefone" => "Telefone",
                    "ecivil" => "Estado Civil"),
              array("ecivil" => "Solteiro"));

```

gera o seguinte conteúdo html:

```

<form>
Nome do Consumidor: <input name = 'nome' type='text'><br>
Estado Civil: <input name = 'ecivil' type='text' value='Solteiro'><br>
</form>

```

que é mostrado por um navegador como :

Nome do Consumidor:

Estado Civil:

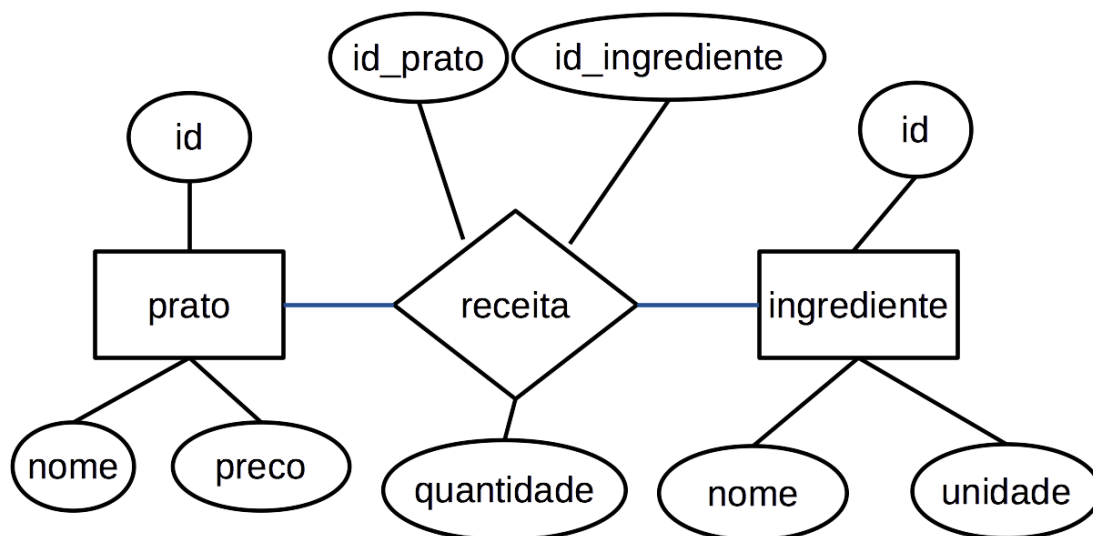
**R:**

```

<?php
function formulario($campos,$legendas,$defaults) {
    echo("<form>\n");
    foreach ($campos as $k) {
        echo "$legendas[$k]: <input name = '$k' type='text'";
        if(array_key_exists($k,$defaults))
            echo " value='$defaults[$k]'";
        echo "><br>\n";
    }
    echo("</form>\n");
}

```

As próximas questões referem-se a um banco de dados simples para armazenar receitas de pratos de um restaurante que foi modelado como no diagrama E-R abaixo. Observe que o atributo “unidade” se refere às unidades usadas em receitas que contêm um dado ingrediente. Assim, por exemplo, “manteiga” é um ingrediente cuja quantidade é dada em unidades de “colheres”, “carne” é um ingrediente cuja quantidade é dada em unidades de “gramas”, etc.



6. (1 ponto) Escreva uma modelagem física para o banco de dados em SQL.

R:

```

CREATE TABLE IF NOT EXISTS prato (
  id int(11) NOT NULL auto_increment,
  nome varchar(50) NOT NULL,
  preco float(10,2) NOT NULL,
  primary key (id)
);
  
```

```

CREATE TABLE IF NOT EXISTS receita (
  id_prato int(11) NOT NULL,
  id_ingrediente int(11) NOT NULL,
  quantidade float(10,2) default 1.0,
  primary key (id_prato,id_ingrediente)
);
  
```

```

CREATE TABLE IF NOT EXISTS ingrediente (
  id int(11) NOT NULL auto_increment,
  nome varchar(40) NOT NULL,
  unidade varchar (20),
  primary key (id)
);
  
```

7. (1 ponto) Escreva comandos INSERT que podem ser usados para registrar uma receita de “bife a cavalo”, um prato de 30 reais que leva 300 gramas de carne e 1 ovo. Você deve assumir que o banco de dados está inicialmente vazio.

R:

```

INSERT INTO prato (nome,preco) VALUES
  ("Bife a cavalo", 30),
  ("Espagete com tomate", 25),
  ("Peixe assado", 50);
  
```

```
INSERT INTO ingrediente (nome,unidade) VALUES
    ("carne", "gramas"),
    ("ovo", ""),
    ("peixe", ""),
    ("molho de tomate", "xicara"),
    ("sal", "pitada"),
    ("manteiga", "colher de sopa");
```

```
INSERT INTO receita (id_prato, id_ingredient, quantidade) VALUES
    ((select id from prato where nome = "Bife a cavalo"),
    (select id from ingrediente where nome = "carne"),
    300),
    ((select id from prato where nome = "Bife a cavalo"),
    (select id from ingrediente where nome = "ovo"),
    1);
```

8. (1 ponto) Escreva um comando SELECT para listar todos os ingredientes do prato “espaguete com tomate”.

**R:**

```
SELECT * FROM ingrediente WHERE id IN (SELECT id_ingredient FROM
receita WHERE id_prato = (SELECT id FROM prato WHERE nome =
"espaguete com tomate"));
```