Fundação CECIERJ - Vice Presidência de Educação Superior a Distância

Curso de Tecnologia em Sistemas de Computação
Disciplina: Programação II
AP2 - 2° semestre de 2013

---

1. (3 pontos) Considere o trecho de código abaixo e responda:
   (a) O que é retornado por `g(array(1,2,3,4))`?

   R: Um array com duas posições: a primeira posição contém um array com os elementos 2, 3 e 4, nessa ordem; a segunda posição contém o número 1.

   (b) O que retorna a função `g` em geral (uma frase curta apenas)?

   R: Um array com duas posições: a primeira contém um array com os elementos do array passado como parâmetro, da segunda posição em diante; a segunda contém o primeiro elemento desse mesmo array.

   (c) O que é retornado por `f(array(1,2,3,4,5), array(2,5))`?

   R: Um array de três posições com os elementos 4, 3 e 1, respectivamente.

   (d) O que retorna a função `f` em geral?

   R: Um array que contém os elementos de $a, excluindo os elementos de $b, em ordem contrária à encontrada no array $a original.

```
function g ($a) {
      $c = array(array());
      foreach ($a as $x) {
            if (count($c) == 1) $c[] = $x;
            else $c[0][] = $x;
      }
      return $c;
}

function f ($a, $b) {
      if (count($a) == 0) return array();
      $c = g($a);
      foreach ($b as $y) {
            if ($c[1] == $y) return f($c[0],$b);
      }
      $r = f($c[0],$b);
      $r[] = $c[1];
      return $r;
}
```

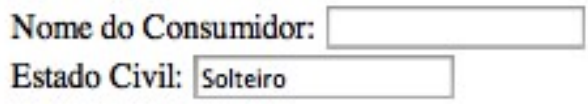
2. (3 pontos) Escreva em a função `formulario($campos,$legendas,$defaults)`, onde `$campos` é um array de strings contendo os campos de um formulário html que deve ser apresentado via web. Para cada elemento `c` desse array, a função formulário deve inserir uma legenda, cujo texto está em `$legendas[c]`, isto é, `$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"));
```

imprime 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:



Gabarito:

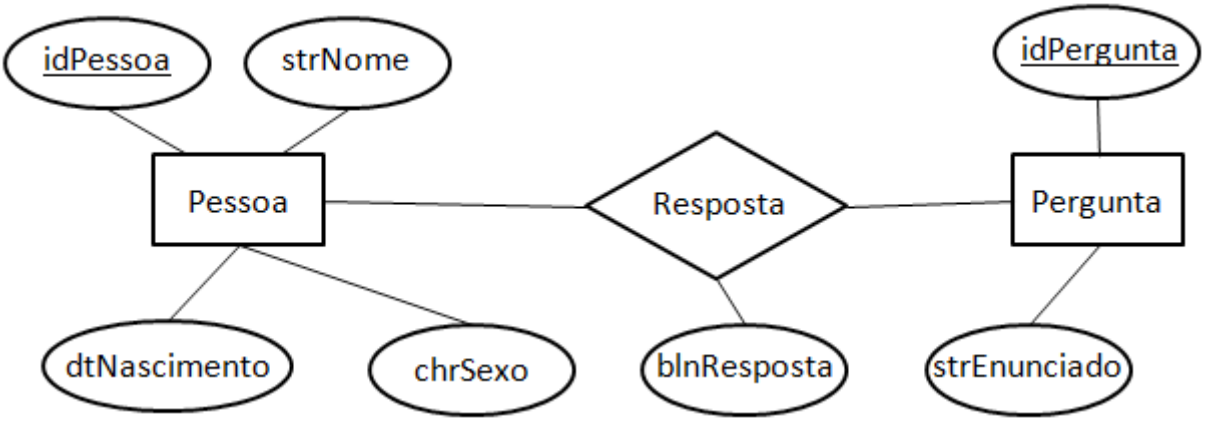
```php
<?php
function formulario($campos,$legendas,$defaults) {
    if (count($campos) > 0) {
        echo "<form>\n";
        foreach ($campos as $c) {
            $l = (array_key_exists($c, $legendas) ? $legendas[$c] : "");
            $d = (array_key_exists($c, $defaults) ? $defaults[$c] : "");

            echo "$l: <input name=\"$c\" type=\"text\" value=\"$d\"><br>\n";
        }
        echo "</form>\n";
    }
}
?>
```

3. (4 pontos) Deseja-se montar um serviço para relacionamentos onde pessoas podem se inscrever, responder a uma série de questões de um formulário visando montar um perfil e tentar encontrar parceiros com perfil semelhante. Além dos dados pessoais, tais como nome, data de nascimento e sexo, o formulário contém um grande número de perguntas que só podem ser respondidas com "Sim" ou "Não", embora a pessoa possa optar por não responder. Tais perguntas fazem parte de um cadastro que pode mudar com o tempo, isto é, o administrador do serviço pode incluir novas perguntas ou retirar perguntas do cadastro.
Pede-se:

(a) (1 ponto) Desenhe um diagrama de entidades e relacionamentos para este banco de dados.

Gabarito:



(b) (1 ponto) Escreva uma modelagem física do banco de dados **em SQL.**

Gabarito:

```
CREATE TABLE `Pessoa` (
    `idPessoa` INT NOT NULL AUTO_INCREMENT,
    `strNome` VARCHAR(45) NOT NULL,
    `chrSexo` CHAR NOT NULL,
    `dtNascimento` DATETIME NOT NULL,
    PRIMARY KEY (`idPessoa`)
);

CREATE TABLE `Pergunta` (
    `idPergunta` INT NOT NULL AUTO_INCREMENT,
    `strEnunciado` VARCHAR(45) NOT NULL,
    PRIMARY KEY (`idPergunta`)
);

CREATE TABLE `Resposta` (
  `idPessoa` INT NOT NULL,
  `idPergunta` INT NOT NULL,
  `blnResposta` TINYINT(1) NOT NULL,
  PRIMARY KEY (`idPessoa`, `idPergunta`),
);
```

(c) (1 ponto) Escreva uma consulta em SQL que retorne o nome de todas as pessoas que podem ser fumantes, isto é, que responderam “Sim” à pergunta “É fumante” ou não responderam à pergunta.

Gabarito:

```
SELECT p.strNome
FROM Pessoa p
WHERE
    Pessoa p NOT IN (
        SELECT p.idPessoa
        FROM
            Pessoa p INNER JOIN Resposta r USING idPessoa
                     INNER JOIN Pergunta q USING idPergunta
        WHERE
            q.strEnunciado = "É fumante" AND
            r.blnResposta = 1
    )
```

(d) (1 ponto) Escreva uma consulta em SQL que retorne todos os inscritos que têm sexo oposto à pessoa cujo nome é "Dagmar da Silva", seja mais jovem que ele/ela e que não tenha respondido à pergunta "Gosta de Futebol" de forma oposta à que Dagmar respondeu.

Gabarito:

```
SELECT p.strNome
FROM
    Pessoa p
WHERE
    p.chrSexo <> (SELECT chrSexo FROM Pessoa WHERE strNome = "Dagmar da Silva" LIMIT 1) AND
    DATEDIFF(p.dtNascimento,
             (SELECT dtNascimento FROM Pessoa WHERE strNome = "Dagmar da Silva" LIMIT 1)) < 0 AND
    p.idPessoa NOT IN (
            SELECT p.idPessoa
            FROM
                idPessoa p INNER JOIN Resposta r USING idPessoa
                           INNER JOIN Pergunta q USING idPergunta
            WHERE
                q.strEnunciado = "Gosta de Futebol" AND
                r.blnResposta <> (
                    SELECT blnResposta
                    FROM
                        Pessoa p INNER JOIN Resposta r USING idPessoa
                                 INNER JOIN Pergunta q USING idPergunta
                    WHERE
                        strNome = "Dagmar da Silva"
                    LIMIT 1
                )
        )
    )
```