

Terceira Prova de Algoritmos e Estruturas de Dados

I

24/11/2006

Perguntas comuns e suas respostas:

- P: Tenho uma dúvida na questão tal.
R: A compreensão do enunciado faz parte da prova.
- P: Se eu consultar algum material próprio ou de algum colega, o que acontecerá comigo?
R: A prova é individual e sem consulta. Qualquer tentativa de fraude acarretará abertura de processo administrativo na UFPR.
- P: Posso fazer a prova a lápis?
A prova é um documento, portanto deve ser feita à caneta.
- P: O que será corrigido?
R: A lógica, a criatividade, a sintaxe, o uso correto dos comandos, a correta declaração dos tipos, os nomes das variáveis, a indentação, uso equilibrado de comentários no código e, evidentemente, a clareza.

Questões (Valores entre parênteses. Total 100 pontos):

1. (50 pontos) Uma matriz B é dita inversa da matriz A quando $A \times B = I$, onde I é a matriz identidade e \times é a operação de multiplicação de matrizes. A matriz identidade é a matriz quadrada onde os elementos da diagonal principal são 1 e os demais 0 ($I[i, j] = 1$ se $i = j$ e $I[i, j] = 0$ se $i \neq j$). Escreva um programa em Free Pascal que leia duas matrizes e testa se a segunda é a inversa da primeira.
2. (50 pontos) Usando as estruturas de dados abaixo escreva um procedimento em Free Pascal que recebe como parâmetro uma estrutura do tipo **TAGENDA** e ordena de forma crescente o vetor **pessoa** dessa estrutura tomando como referência para a ordenação o campo **nome** da estrutura **TPESSOA**. Ou seja, ordena uma agenda pessoal de telefones e endereços em ordem crescente do nome das pessoas presentes na agenda. Você *deve* usar a função *compara(r, s)*, que recebe dois parâmetros do tipo **string** e retorna 0 se r e s são iguais, 1 se r é lexicograficamente maior que s e -1 se r é lexicograficamente menor que s . Um nome n_1 é lexicograficamente maior que um nome n_2 se n_1 aparece depois de n_2 numa ordenação alfabética crescente desses nomes.

Const

```
MAX = 1000;
```

Type

```
TPESSOA = record
```

```
    nome: string;
```

```
    telefone: string;
```

```
    endereco: string
```

```
end;
```

```
TAGENDA = record
```

```
    pessoa: array [1..MAX] of TPESSOA;
```

```
    tamanho: integer
```

```
end;
```