

# 2ª Prova de Algoritmos e Estruturas de Dados I

## 20/02/2013

### 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 entregar a prova a lápis?  
R: prova é um documento, portanto deve ser entregue a 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. A modularidade, correto uso de funções e procedimentos, incluindo passagem de parâmetros e bom uso de variáveis locais e globais serão especialmente observados.

#### 1. Questão 1 (40 pontos)

Escreva um programa em linguagem Pascal para ler, a partir do teclado, duas sequências com quantidades arbitrárias de valores inteiros positivos, as quais denotam dois conjuntos da matemática. Os valores de uma sequência nunca conterão duplicatas mas podem vir fora de ordem. Cada sequência termina com o valor 0 (que não é elemento de nenhum dos conjuntos). Depois da leitura dos dados, o programa deve determinar se os conjuntos possuem interseção vazia. Se isso ocorrer, a palavra “Sim” deve ser impressa. No caso contrário, a palavra “Nao” deve ser impressa. Por exemplo, considere a entrada de dados abaixo:

```
11 7 1 18 5 9 0 <ENTER>
22 4 5 6 0 <ENTER>
```

Ao final da leitura e do cálculo, o monitor de vídeo deverá mostrar:

Nao

Note que o programa não precisa determinar o conjunto-interseção em si, mas apenas se ele é vazio.

#### 2. Questão 2 (60 pontos)

Escreva um programa em linguagem Pascal para ler, a partir do teclado, uma sequência com uma quantidade arbitrária de valores. A sequência termina com o valor 0 (que não faz parte dela). Os valores lidos devem ser números inteiros entre 1 e 9 (incluindo 1 e 9). Os valores da sequência podem vir fora de ordem. Por definição, uma **subsequência** é aquela constituída por 2 (duas) ou mais ocorrências seguidas do mesmo **valor-base**. Por exemplo, o número 3 é o **valor-base** da **subsequência** 3 3. O programa deve determinar e imprimir o **valor-base** da **subsequência** mais frequente dos dados de entrada. Por exemplo, considere a entrada de dados abaixo:

```
2 5 5 5 5 8 8 2 5 5 5 5 3 8 8 5 1 1 2 8 8 2 5 2 8 8 2 5 5 5 5 7 0 <ENTER>
```

Ao final da leitura e do cálculo, como a **subsequência** 8 8 ocorre quatro vezes (todas as demais são menos frequentes), o monitor de vídeo deverá mostrar seu **valor-base** como saída do programa: