

2ª Prova de Algoritmos e Estruturas de Dados I

18/05/2011

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 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. 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 única (100 pontos)

Considere as seguintes definições:

```
const max = 20;
type cores = 1..30; (* faixa de numeros entre 1 e 30, subtipo de integer *)
    vetor = array [1..max] of cores;
```

Escreva, em *Pascal*, procedimentos ou funções para:

- Ler um vetor de n , $n \leq max$, posições de elementos do tipo *cores*, garantindo que todos os elementos pertencem a faixa válida e que não haja nenhum elemento repetido no vetor;
- Receba como parâmetros dois vetores do tipo *vetor* e conte quantas vezes os elementos do segundo vetor aparecem no primeiro, mas em posições diferentes;
- Receba como parâmetros dois vetores do tipo *vetor* e conte quantas vezes os elementos do segundo vetor aparecem no primeiro, mas nas mesmas posições;

Escreva um programa em *Pascal* que use estes subprogramas para realizar as seguintes tarefas:

- Ler um vetor S de n cores;
- Ler um conjunto de vetores V , de n cores cada um, e para cada um deles imprimir quantas cores estão no mesmo lugar do vetor S e quantas cores estão fora de lugar no vetor S .
- A leitura de vetores termina quando for dado como entrada um vetor de cores que tem exatamente as mesmas cores e na mesma ordem que o vetor S .
- Ao final, imprimir quantos vetores V foram lidos.

Exemplo:

```
Entrada: 7 3 5 6                (* vetor S *)
Entrada: 7 4 3 2
Saida  : cores no lugar certo: 1, cores no lugar errado: 1.
Entrada: 1 4 3 5
Saida  : cores no lugar certo: 0, cores no lugar errado: 2.
Entrada: 7 3 6 5
Saida  : cores no lugar certo: 2, cores no lugar errado: 2.
Entrada: 7 3 5 6
Saida  : cores no lugar certo: 4, cores no lugar errado: 0.
Foram lidos 4 vetores.
```