

2ª Prova de Algoritmos e Estruturas de Dados I

25/05/2016

Perguntas comuns e suas respostas:

- P: Tenho uma dúvida na questão tal.
R: A compreensão do enunciado faz parte da prova.
- 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.

Questão única (100 pontos): Jogo *Mastermind*

Considere as seguintes definições:

```
const MAX = 20;   CORINI=1;   CORFIM=30;
type cores = CORINI..CORFIM; (* faixa de valores, subtipo de integer *)
   vetor = array [1..MAX] of cores;
```

1. Escreva, em *Pascal*, procedimentos ou funções para:
 - (a) Ler um vetor com $n \in [2..MAX]$ elementos do tipo **cores**, garantindo que todos os elementos pertencem ao intervalo $[CORINI, CORFIM]$ e que não haja nenhum elemento repetido no vetor lido;
 - (b) Receba como parâmetros dois vetores do tipo **vetor** e conte quantas vezes os elementos do segundo vetor aparecem no primeiro, mas em posições diferentes;
 - (c) Receba como parâmetros dois vetores do tipo **vetor** e conte quantas vezes os elementos do segundo vetor aparecem no primeiro, mas nas mesmas posições;
2. Escreva um programa estruturado em *Pascal* que use estes subprogramas (e outros que forem necessários) para:
 - (a) Ler n e um vetor S de n cores;
 - (b) Ler um conjunto de vetores V , de n cores cada um, e para cada um deles imprimir quantas cores estão no mesmo lugar e quantas cores estão fora de lugar em relação ao vetor S .
 - (c) Entradas que não satisfizerem as restrições do item “??” devem ser ignoradas;
 - (d) 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 .
 - (e) Ao final, imprimir quantos vetores V foram lidos.

Exemplo:

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