

3ª Prova de Algoritmos e Estruturas de Dados I

15/06/2018

Perguntas comuns e suas respostas:

- P: O que será avaliado?

R: A lógica, a criatividade, a sintaxe, o uso correto dos comandos, a correta declaração dos tipos, os nomes das variáveis, a indentação, o uso equilibrado de comentários no código e, evidentemente, a clareza. A modularidade, o correto uso de funções e procedimentos, incluindo passagem de parâmetros, e o bom uso de variáveis locais e globais e também a eficiência do seu algoritmo serão especialmente observados.

QUESTÃO: (100 pontos)

Considere as seguintes constantes e tipos assim definidos:

```
CONST MAX = 200;  
TYPE tpMatriz = array [1..MAX,1..MAX] of integer;
```

Considere os seguintes protótipos de funções e procedimentos, junto da descrição do seu funcionamento:

- **function** achar_X (var M: tpMatriz; N, linIni, colIni, tam: integer) : boolean;

Esta função deve procurar em uma matriz $M_{N \times N}$ por um conjunto de elementos cuja disposição forma visualmente um “X” constituído de números iguais, conforme pode ser visto na figura da esquerda abaixo.

A função recebe a matriz, uma coordenada ($linIni, colIni$) – linha e coluna respectivamente – e um número inteiro tam ($2 \leq tam \leq N$) que representa o tamanho do “X”. A função retorna **TRUE** se os elementos contidos no “X” de tamanho tam , cujo canto superior esquerdo está na coordenada ($linIni, colIni$), possuem todos o mesmo valor, e retorna **FALSE** caso contrário. O teste da existência do “X” não deve ultrapassar os limites da matriz.

Exemplo: A matriz $M_{6 \times 6}$ da figura abaixo, na **esquerda**, possui um “X” de tamanho 4 constituída de elementos “1” iniciando na coordenada (2,2). Assim, se a função receber esta matriz com $N = 6$, $linIni = 2$, $colIni = 2$ e $tam = 4$, deve retornar **TRUE**. Se receber a mesma matriz com $N = 6$, $linIni = 1$, $colIni = 1$ e $tam = 6$, deve retornar **FALSE** (pois esse “X” não existe). Se receber a mesma matriz com $N = 6$, $linIni = 4$, $colIni = 4$ e $tam = 4$, deve retornar **FALSE** (pois ultrapassa os limites da matriz).

- **procedure** inverte_cor (var M: tpMatriz; N: integer);

Este procedimento recebe uma matriz $M_{N \times N}$. O objetivo é inverter as cores da matriz. A inversão de cor funciona ao obter o maior número da matriz **max** e, para cada coordenada da matriz, subtrair de **max** o valor da coordenada.

Exemplo: Para a matriz $M_{6 \times 6}$ da esquerda abaixo, o resultado será a matriz da direita.

- **procedure** ler_matriz (var M: tpMatriz; N: integer);

Este procedimento lê do teclado uma matriz $M_{N \times N}$ de números inteiros.

- **procedure** imprimir_matriz (var M: tpMatriz; N: integer);

Este procedimento imprime na tela uma matriz $M_{N \times N}$ de números inteiros.

O QUE DEVE SER FEITO (NESTA ORDEM):

1. Suponha, para este item 1, que as funções e procedimentos acima **já estão implementadas e funcionais**. Faça um programa Pascal completo que utiliza as funções e procedimentos acima para resolver o seguinte problema. O programa deve ler N e uma matriz $N \times N$ de números inteiros e imprimir a matriz resultante do seguinte processo. Se a matriz contém um “X” de **qualquer** tamanho tam (onde $2 \leq tam \leq N$), então deve-se imprimir a matriz com as cores invertidas. Caso contrário, deve-se imprimir a matriz sem modificações.
2. Implemente as funções e procedimentos acima.

```
2 3 1 5 2 3
4 1 3 2 1 1
2 2 1 1 3 2
3 5 1 1 2 3
2 1 4 4 1 4
5 5 2 3 4 3
```

```
3 2 4 0 3 2
1 4 2 3 4 4
3 3 4 4 2 3
2 0 4 4 3 2
3 4 1 1 4 1
0 0 3 2 1 2
```