

3ª Prova de Algoritmos e Estruturas de Dados I

21/12/2011

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 fazer a prova a lápis?
R: prova é um documento, portanto deve ser feita 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.

Questões: Aonde está Wally?

Um dos formatos para representação de imagens em tons de cinza é o PGM. Este formato, simplificadaamente, consiste de um arquivo ASCII com o seguinte conteúdo:

- A primeira linha contém os caracteres "P2"
- A segunda linha contém a largura e a altura de uma matriz, isto é, respectivamente o número de colunas (*col*) e de linhas (*lin*) da imagem.
- A terceira linha contém o maior valor da matriz.
- as linhas seguintes contém *col* números $-1 \leq col \leq 255$ representando o valor de cada ponto da imagem. O valor -1 representa transparência.

1. (70 pontos) Faça um programa completo em free Pascal que leia da entrada padrão (teclado) duas imagens no formato PGM: imagem original (*imgO*) e a imagem do padrão (*imgP*). Em seguida, o programa deve procurar se a imagem *imgP* está contida na imagem *imgO* e imprimir na tela as coordenadas (*coluna*, *linha*) do canto superior esquerdo de **cada ocorrência** da imagem *imgP* encontrada na imagem *imgO*.

Observações:

- A imagem *imgP* pode aparecer mais de uma vez na imagem *imgO*;
- Na imagem *imgP*, pontos com o valor -1 devem ser ignorados, isto é, representam pontos transparentes do padrão e não devem ser comparados com a imagem *imgO*.
- Estructure seu código. A solução parcial ou a indicação de chamadas a funções não implementadas serão consideradas.

Exemplo:

- **Imagem original:**

```
P2
11 10
40
40 5 5 5 5 5 5 5 5 40 0
5 20 20 5 5 5 5 5 5 5 5
5 5 20 5 5 5 0 0 0 0 0
5 5 20 20 5 5 20 20 0 0 5
5 5 5 5 5 5 0 20 0 0 0
5 5 5 5 5 5 0 20 20 0 5
5 5 5 5 11 11 11 0 0 0 0
5 5 5 5 20 20 11 5 5 5 5
5 5 5 5 11 20 11 5 5 5 0
40 5 5 5 11 20 20 5 5 40 5
```

- **Imagem do padrão:**

```
P2
3 3
20
20 20 -1
-1 20 -1
-1 20 20
```

- **Resultado do Programa:**

```
2 2
7 4
5 8
```

2. (30 pontos) Modifique seu programa de forma que, ao invés de imprimir as coordenadas, a saída do programa seja uma nova imagem que consiste de uma cópia da imagem original *imgO* na qual as ocorrências da imagem *imgP* sejam preenchidas com o valor 0.

Exemplo da nova saída para a entrada original:

- **Imagem resultante:**

```
P2
11 10
40
40 5 5 5 5 5 5 5 5 40 0
5 0 0 0 5 5 5 5 5 5 5
5 0 0 0 5 5 0 0 0 0 0
5 0 0 0 5 5 0 0 0 0 5
5 5 5 5 5 5 0 0 0 0 0
5 5 5 5 5 5 0 0 0 0 5
5 5 5 5 11 11 11 0 0 0 0
5 5 5 5 0 0 0 5 5 5 5
5 5 5 5 0 0 0 5 5 5 0
40 5 5 5 0 0 0 5 5 40 5
```