

# Programação Imperativa

## LEI + LCC (1º ano)

### 5ª Ficha Prática

Ano Lectivo de 2006/07

## Objectivos

O objectivo desta ficha é treinar o aluno no processamento de imagens recorrendo à utilização de matrizes.

## Exercícios

O PNM é um formato de imagem muito simples, fácil de ler e de escrever de/para um ficheiro. Actualmente uma imagem PNM pode pertencer a uma de três famílias:

- PBM (Portable BitMap file format) — P1/P4 — para imagens a preto e branco;
- PGM (Portable GrayMap file format) — P2/P5 — para imagens em tons de cinza;
- PPM (Portable PixMap file format) — P3/P6 — para imagens a cores (RGB).

Cada família tem duas representações: uma *textual* com 1 caracter por byte (P1, P2 ou P3) e uma *compactada* (P4, P5 ou P6).

Embora não seja um requisito da especificação PNM, existe a convenção de que a imagem deve ser armazenada de cima para baixo e da esquerda para a direita.

Relativamente à família PBM, que corresponde a imagens contendo apenas duas cores, branco (1) e preto (0), descreve-se a seguir o formato do ficheiro que contem cada imagem:

- qualquer linha iniciada pelo carácter # é um comentário; os comentários só podem aparecer nas linhas do cabeçalho, ou seja, antes dos valores dos pontos da imagem;
- a primeira linha contem o identificador do tipo da imagem: P1 (ou P4);
- a segunda linha contem um par de valores que definem respectivamente: o decimal correspondente ao número de colunas (largura da imagem em pixels), por exemplo 9; e o decimal correspondente ao número de linhas (altura da imagem em pixels), por exemplo 7;
- as linhas restantes contêm uma lista de caracteres (correspondentes aos valores decimais), em que cada valor corresponde a um pixel da imagem e estão organizadas de acordo com os valores definidos nas linhas anteriores. Com os exemplos dados nos itens anteriores teríamos 7 linhas cada uma com 9 valores separados por um espaço (cada valor corresponde a um byte, 0-1, no caso das imagens do tipo P1).  
No formato P4, cada grupo de 8 bytes (caracteres) com 0 ou 1 é compactado em 1 só byte.

Para a família PGM a lógica é a mesma, mas agora cada pixel é representado por um caracter com um valor que vai do 0 = preto ao 255 = branco.

1. Desenvolva um programa, em linguagem C, para converter uma imagem do tipo P1 (formato textual) numa imagem do tipo P4 (formato compactado) e para fazer a conversão inversa (o próprio algoritmo deve decidir o sentido da conversão em função do tipo da imagem que lhe forma passada).  
O ficheiro contendo a imagem a converter deve ser indicado na linha de comandos à frente do nome do programa.

Exemplo de uma imagem do tipo P1 e respectiva conversão para P4:

P1	P4
# PBM example	# PBM example
9 7	9 7
0 0 0 0 0 0 0 0	0 0
0 1 1 1 1 0 0 1	121 0
0 1 0 0 1 0 0 1	73 0
0 1 1 1 1 0 0 1	121 0
0 1 0 0 0 0 0 1	65 0
0 1 0 0 0 0 0 1	65 0
0 0 0 0 0 0 0 0	0 0

A imagem em causa é:

2. Desenvolva um programa, em linguagem C, semelhante ao anterior, mas agora para converter uma imagem do tipo P2 numa imagem do tipo P1 (considere para já que a escolha entre preto e branco se faz em torno do valor 128).
3. Desenvolva um algoritmo, e codifique-o em linguagem C, para rodar uma imagem P1 90 graus à direita.
4. Investigue os restantes tipos de formato (PGM e PPM) e desenvolva um algoritmo, e codifique-o em linguagem C, para converter uma imagem do tipo P2/P5/P6/P3 numa imagem do tipo P1/P4 (conversão *cores* → *preto-branco*).

Como valor de *threshold* para a *binarização*, pode considerar o valor médio absoluto entre 0 e 255, ou uma média real (Max/2). Alternativamente a este último, e de modo a fazer uma *binarização adaptativa* mais precisa, poderá considerar o valor médio da diferença entre o valor máximo e a média dos pixels da imagem ((Max-Média)/2).

Experimente também a influência que tem na qualidade da imagem obtida, tomar como *threshold* para *binarização* um valor específico para cada pixel que depende dos pixel na sua vizinhança:

- (a) Média dos valores dos seus vizinhos da esquerda e direita (*nvizinhos=2*);
- (b) Média dos valores dos seus vizinhos da esquerda, direita, cima e baixo (*nvizinhos=4*);
- (c) Média dos valores dos seus vizinhos anteriores e ainda os quatro das diagonais (*nvizinhos=8*).