

**CCM0118 — Computação I**

CURSO DE CIÊNCIAS MOLECULARES — TURMA 22 — SEGUNDO SEMESTRE DE 2012

Quarto Exercício-Programa

Prazo de entrega: até **4 de dezembro de 2012**.Filtro da Mediana

Seja  $A$  uma matriz de inteiros positivos com  $m$  linhas e  $n$  colunas, e sejam  $p$  e  $q$  dois inteiros positivos ímpares. Dada um par de índices  $(i, j)$ , com  $p/2 \leq i < m - p/2$  e  $q/2 \leq j < n - q/2$ , a vizinhança de tamanho  $p \times q$  em torno de  $(i, j)$  é a submatriz de  $A$  com  $p$  linhas,  $q$  colunas e centro em  $(i, j)$ . Essa submatriz é denotada por  $A_{i,j}$ .

Por exemplo, dada a seguinte matriz  $5 \times 5$

$$A = \begin{bmatrix} 9 & 4 & 5 & 0 & 8 \\ 10 & 3 & 2 & 1 & 7 \\ 9 & 1 & 6 & 3 & 15 \\ 0 & 3 & 8 & 10 & 1 \\ 1 & 16 & 9 & 12 & 7 \end{bmatrix}$$

a vizinhança  $3 \times 3$  em torno de  $(1, 1)$  é a submatriz

$$A_{1,1} = \begin{bmatrix} 9 & 4 & 5 \\ 10 & 3 & 2 \\ 9 & 1 & 6 \end{bmatrix}$$

Note que a vizinhança não está definida para os pares de índices muito próximos à “borda” da matriz (por exemplo, em  $(0, 0)$ ).

**1 O filtro da mediana**

Filtro da mediana é uma transformação bastante usada para suavizar ruído do tipo impulsivo em sinais e imagens digitais.

Uma imagem digital pode ser representada por uma matriz. Dada uma matriz  $A$  de inteiros positivos, com  $m$  linhas e  $n$  colunas, e dados dois inteiros positivos e ímpares  $p$  e  $q$ , o filtro da mediana produz uma matriz transformada  $M$ , com as mesmas dimensões que  $A$ , definida da seguinte maneira: para cada par de índices  $(i, j)$ , o elemento  $M(i, j)$  da matriz transformada é a mediana dos elementos de  $A_{ij}$  (a vizinhança  $p \times q$  em torno de  $(i, j)$ ).

No caso do exemplo anterior, os números em torno de  $(1, 1)$  são 9, 4, 5, 10, 3, 2, 9, 1, 6. Logo,  $M(1, 1) = 5$ . Quando a vizinhança de uma par de índices  $(i, j)$  não estiver bem definida, usaremos a convenção  $M(i, j) = 0$ .

No caso da matriz  $A$  acima, o resultado do filtro da mediana com uma vizinhança  $3 \times 3$  é a seguinte matriz:

$$M = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 5 & 3 & 5 & 0 \\ 0 & 3 & 3 & 6 & 0 \\ 0 & 6 & 8 & 8 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

## 2 O que o seu programa deve fazer

Você deverá escrever um programa que:

- lê uma imagem de um arquivo e a armazena em uma matriz (mais detalhes na seções 3 e 4),
- calcula o resultado do filtro da mediana,  $M$ , conforme descrito na seção 1, e
- grava a imagem  $M$  em um arquivo (mais detalhes nas seções 3 e 4)

O seu programa deverá também imprimir as matrizes sempre que as duas dimensões (o número de linhas e o de colunas) dessas matrizes forem menores ou iguais a 16. Veja um exemplo de saída do programa na seção 5.

## 3 Formato PGM

Neste EP utilizaremos o formato PGM (*Portable Gray Map*) para armazenar imagens em arquivos. Esse formato é voltado para imagens em tons de cinza. Um arquivo no formato PGM deve conter um cabeçalho seguido da matriz correspondente à imagem. Veja exemplo a seguir.

```
P2
5 4
16
9 4 5 0 8
10 3 2 1 7
9 1 6 3 15
1 16 9 12 7
```

A primeira linha do arquivo contém a palavra-chave “P2”, que é obrigatória. A segunda linha contém dois números que correspondem ao número de colunas e linhas da matriz, respectivamente. A terceira linha contém um valor denominado *maxval*, o qual deve ser maior ou igual ao maior elemento da matriz. Para fins deste EP, *maxval* é no máximo 255. Os demais números do arquivo correspondem aos tons de cinza da imagem armazenados em forma de uma matriz de inteiros. Cada tom de cinza é um número entre 0 e *maxval*, com 0 indicando “negro” e *maxval* indicando “branco”.

O formato PGM também permite a colocação de comentários na imagem. A partir de uma ocorrência do caractere ‘#’, todos os caracteres (incluindo o ‘#’) até o próximo fim de linha (caractere ‘\n’) formam um comentário e portanto devem ser ignorados. Exemplo de imagem com comentários:

```
P2
# imagem: exemplo.pgm
5 4
16
9 4 5 0 8
10 3 2 1 7
9 1 6 3 15
1 16 9 12 7
```

## 4 Arquivo de imagens: leitura e escrita

O programa a seguir mostra como fazer a leitura do cabeçalho de um arquivo PGM.

```

#include <stdio.h>
#include <string.h>

#define MAX_NAME 256 /* tamanho maximo para nome de arquivo */
#define MAX      512 /* dimensao maxima para matrizes */

int main()
{
    FILE *arq;
    char fname[MAX_NAME];
    char key[128];
    int m, n, maxval, a[MAX][MAX];
    int aux, i, j;

    /* leitura do nome do arquivo de entrada */
    printf("Digite o nome do arquivo de entrada: ");
    scanf("%s", fname);

    /* abre arquivo para leitura */
    arq = fopen(fname, "r");
    if (arq == NULL) {
        printf("Erro na abertura do arquivo %s\n", fname);
        return 0;
    }

    /* le dados do cabeçalho */
    aux = fscanf(arq, "%s", key);
    if (aux != 1) {
        printf("Erro na leitura do arquivo %s\n", fname);
        fclose(arq);
        return 0;
    }
    if (strcmp(key, "P2") != 0) {
        printf("Formato desconhecido\n");
        fclose(arq);
        return 0;
    }
    aux = fscanf(arq, "%d %d %d", &n, &m, &maxval); /* note que o numero de colunas n
                                                    vem antes do numero de linhas m */
    if (aux != 3) {
        printf("Formato incorreto\n");
        fclose(arq);
        return 0;
    }

    /* le a matriz m x n que vem depois do cabeçalho
        ...
        ...
    */

    fclose(arq); /* fecha arquivo */
    return 0;
}

```

Usando esse exemplo como ponto de partida, escreva uma função para ler um arquivo no formato PGM e outra para escrever um arquivo no formato PGM. Essas funções devem ter os seguintes protótipos:

```
/* -----  
Funcao que le um arquivo no formato PGM.  
  fname   : nome do arquivo PGM  
  a       : matriz com a imagem lida  
  *m      : numero de linhas da matriz  
  *n      : numero de colunas da matriz  
  *maxval : valor correspondente ao tom branco  
----- */  
int read_pgm(char fname[], int a[][MAX], int *m, int *n, int *maxval);  
  
/* -----  
Funcao que escreve um arquivo no formato PGM.  
  fname   : nome do arquivo PGM  
  a       : matriz com a imagem a ser gravada  
  m       : numero de linhas da matriz  
  n       : numero de colunas da matriz  
  maxval  : valor correspondente ao tom branco  
----- */  
int write_pgm(char fname[], int a[][MAX], int m, int n, int maxval);
```

A idéia é que a função `read_pgm` faça tudo o que for preciso para ler o arquivo PGM cujo nome é dado pelo parâmetro `fname`. Ela deve abrir esse arquivo para leitura (por meio de uma chamada a `fopen`), ler o conteúdo do arquivo e depois fechar o arquivo. Além de preencher a matriz `a`, a função `read_pgm` deve colocar em `*m`, `*n` e `*maxval` as dimensões da matriz e o valor de `maxval` lido do arquivo PGM.

A leitura de um arquivo PGM pode ser bem sucedida ou não. (Por exemplo: a chamada a `fopen` pode falhar porque não existe o arquivo `fname`.) O `int` devolvido pela função `read_pgm` deve ser usado para informar se a leitura correu bem ou não.

Analogamente, a idéia é que a função `write_pgm` faça tudo o que for preciso para escrever um arquivo PGM com nome dado pelo parâmetro `fname`. Ela deve abrir esse arquivo para escrita (por meio de uma chamada a `fopen`), escrever o conteúdo do arquivo e depois fechar o arquivo. O `int` devolvido por essa função deve ser usado para informar se a escrita foi bem sucedida ou não.

## 5 Exemplo

Digite o nome do arquivo de entrada: exemplo.pgm

Matriz original:

```
 9  4  5  0  8  
10  3  2  1  7  
 9  1  6  3 15  
 0  3  8 10  1  
 1 16  9 12  7
```

Digite o tamanho da vizinhanca (p q): 3 3

Calculando mediana, aguarde...

Matriz mediana:

```
0 0 0 0 0
0 5 3 5 0
0 3 3 6 0
0 6 8 8 0
0 0 0 0 0
```

Digite o nome do arquivo de saída: sai.pgm

## 6 Sobre a organização do programa

- Não devem ser usadas variáveis globais.
- As funções `main`, `read_pgm` e `write_pgm` não precisam (e nem devem!) ser as únicas funções do seu programa. A recomendação, aqui, é a mesma do EP3: “Organize seu programa em funções. Use funções para dividir as tarefas do seu programa em tarefas menores e mais simples. Idealmente cada função deve ser curta e fácil de entender. Se alguma função parecer longa ou complexa, crie uma ou mais funções auxiliares de modo a dividir o trabalho realizado por aquela função!”

## 7 Outras informações

Em <http://www.ime.usp.br/~reverbel/ccm118-12/#EPs> estão disponíveis dois exemplos bem simples de arquivos PGM e alguns arquivos de entrada (contendo imagens “sujas”) para você testar o seu programa.

Para visualizar uma imagem no formato PGM, use qualquer visualizador de imagens que entenda este formato. No Linux você pode usar o `eog` ou o `kview`, dois visualizadores geralmente presentes nas distribuições de Linux. Ambos aceitam arquivos PGM.

Um visualizador de imagens que roda no Windows e aceita arquivos PGM é o IrfanView, disponível gratuitamente no sítio <http://www.irfanview.com/>. Você pode instalar o IrfanView no seu computador e usá-lo para visualizar tanto as imagens de entrada como as de saída.

**Bom trabalho!**