

# Exame Final de Algoritmos e Estruturas de Dados I

## 04/07/2017

### O que será avaliado?

- 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 serão especialmente observados.

### Introdução (Adaptado de GIMP.org):

Diversos software de edição de imagem utilizam **filtros** para aplicar efeitos na imagem, como desfocagem, nitidez, detecção de borda, dentre outros. Estes efeitos podem ser obtidos utilizando uma *matriz de convolução*.

Convolução é o processo de aplicar uma matriz (chamada de “núcleo” ou kernel), sobre outra matriz que contém a imagem (chamada de matriz primária). A imagem é uma matriz bi-dimensional de pixels e o núcleo a ser usado depende do efeito desejado. A ferramenta de processamento de imagens GIMP utiliza como núcleos matrizes  $5 \times 5$  ou  $3 \times 3$ . Nós consideraremos aqui apenas núcleos  $3 \times 3$ . Um exemplo de aplicação de uma matriz de convolução está ilustrado na Figura 1(a). A imagem da direita foi obtida aplicando o núcleo ilustrado na Figura 1(b).

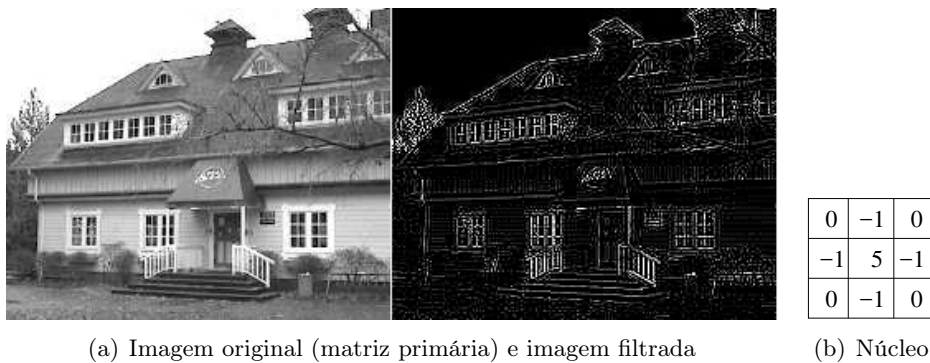
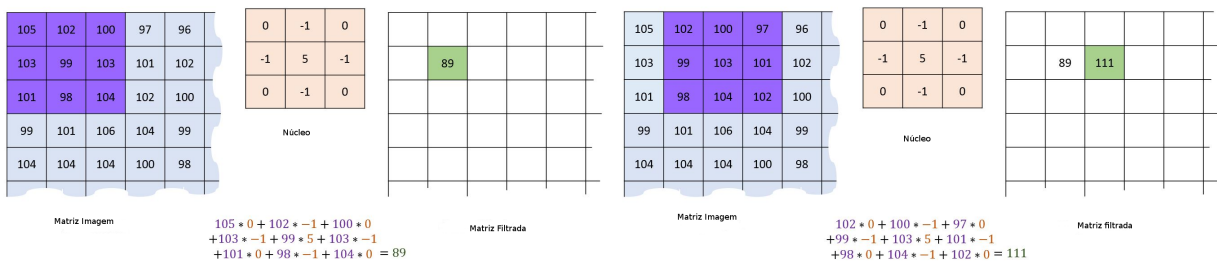


Figura 1: Exemplo de utilização de matriz de convolução

Para obter este efeito, cada pixel da imagem é tratado da seguinte forma. O novo valor do pixel é obtido multiplicando-se o valor dos pixels ao seu redor pelo valor correspondente na matriz núcleo. Esses nove valores são adicionados e se tornam o valor final daquele pixel. Um exemplo é apresentado na figura abaixo.



Isso é o que acontece: o filtro lê sucessivamente, da esquerda para a direita e de cima para baixo, todos os pixels da área de ação do núcleo. Ele multiplica o valor de cada um deles pelo valor correspondente do núcleo e depois soma os resultados. No exemplo à esquerda, o pixel central se tornou um 89:  $(105 \cdot 0) + (102 \cdot -1) + (100 \cdot 0) + (103 \cdot -1) + (99 \cdot 5) + (103 \cdot -1) + (101 \cdot 0) + (98 \cdot -1) + (104 \cdot 0) = 89$ . Observe que o filtro não altera diretamente a imagem original, mas sim produz uma nova imagem. Ou seja, os valores utilizados para o cálculo são os da imagem original e o resultado está em uma nova matriz, como mostra o exemplo à direita.

## 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 aplicar_filtro_no_pixel (var M, K: tpMatriz; N,x,y: integer): integer;`

Esta função aplica um filtro na matriz primária  $M_{N \times N}$  pela convolução com a matriz núcleo  $K_{3 \times 3}$ , conforme explicado na introdução. A convolução é aplicada na submatriz  $3 \times 3$  da matriz  $M$  com o pixel central na linha  $x$  e coluna  $y$ . A função devolve como resultado o inteiro resultante da operação. A função assume que a coordenada  $(x, y)$  está dentro dos limites da matriz nos quais é possível aplicar a operação. Por exemplo, não é possível a convolução de  $M$  com  $K$  na coordenada  $(1, 1)$  e nem  $(2, 1)$ , pois a parte da matriz núcleo usada na operação  $(3 \times 3)$  não “cabe” com o centro nestas coordenadas. Você não deve fazer o teste, cabe a quem chama a função garantir a corretude dos dados.

- `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 implementados e funcionais**. Faça um programa Pascal completo que utiliza as funções e procedimentos acima para resolver o seguinte problema. Ler um número  $N$  do teclado e uma matriz primária  $M_{N \times N}$ , bem como uma matriz núcleo  $K_{3 \times 3}$  e obter uma matriz  $R_{N \times N}$  que é a matriz resultante do processo de convolução com a matriz núcleo  $K$ . A chamada da função deve respeitar o fato de que a função não está preparada para aceitar qualquer coordenada, conforme explicado no funcionamento da função acima. O filtro deve ser aplicado em **todos os pixels** para os quais a função garantidamente funcione. Os pixels sobre os quais a função não pode ser aplicada devem manter o seu valor original. Esta parte central do programa deve estar contida no programa principal. Ao final, imprimir a matriz de entrada e a matriz filtrada.
2. Implemente as funções e procedimentos acima.

**Observação:** Você pode criar e usar outras funções e procedimentos se achar necessário, mas neste caso os códigos destes devem ser apresentados.