

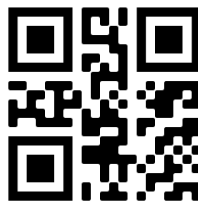
MC202AB - Estrutura de Dados

Lab 2 - Matrizes

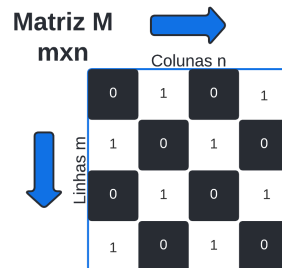
Data da Primeira Chance: 5 de setembro de 2022

Peso: 1

QR Codes são códigos de barra formados por padrões em uma imagem binária que podem ser escaneados na câmera do celular. Eles contêm informações, como link de sites, imagens e textos.



Imagens podem ser vistas como matrizes, onde cada elemento da matriz corresponde ao valor de intensidade do píxel. No caso de imagens binárias, os elementos assumem valores 1 (branco) ou 0 (preto).



Considerando isso,

(a) Inicialmente, defina uma imagem binária A de dimensão $N \times N$, com listras verticais de espessura de 3 colunas, começando com branco. Defina também uma matriz B iniciada igual à matriz A .

(b) Elabore uma operação TRANSPOSTA que dado uma matriz de entrada A , transpõem a matriz.

(c) Elabore uma operação SOMA que calcula a soma de duas matrizes de entrada $N \times N$ (e.g., A e B). Lembrando que como a imagem binária só assume valores de 0 e 1, considere que $1 + 1 = 1$.

(d) Elabore uma operação MULTI_ELEM de multiplicação elemento a elemento de duas matrizes de entrada NxN (e.g., A e B) e calcula o resultado, onde $C_{M \times N} = A_{M \times N} \times B_{M \times N}$, por exemplo:

$$\begin{bmatrix} 1 & 4 \\ 9 & 16 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \times \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

Lembrando que como a imagem binária só assume valores de 0 e 1, considere que $1 + 1 = 1$.

(e) Elabore uma operação MULTI_MAT de multiplicação de matrizes com duas matrizes de entrada NxN (e.g., A e B) e calcula o resultado, onde $C = A.B$, por exemplo:

$$\begin{bmatrix} 7 & 10 \\ 15 & 22 \end{bmatrix} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}$$

Lembrando que como a imagem binária só assume valores de 0 e 1, considere que $1 + 1 = 1$.

(f) Bordas são transições na imagem que definem objetos, onde dada uma imagem de entrada X, a imagem Y representa as bordas do objeto.

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad X \quad \rightarrow \quad \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad Y$$

Elabore uma operação BORDAS que define as bordas da matriz de entrada NxN (e.g A).

Entrada

A primeira entrada é o número inteiro 'N' de linhas e colunas para as matrizes ($6 \leq N \leq 99$ e múltiplo de 3) da matriz A. A segunda entrada é o valor inteiro 'O' que representa a quantidade de operações a serem feitas. Em seguida, recebe-se uma lista de operações a serem feitas com suas devidas entradas, onde a saída é guardada em um acumulador B a ser utilizado na próxima operação.

Lembre-se que as matrizes A e B começam como definida no item (a) enunciado, porém podem ser usada como um "acumulador", logo seu valor pode ser alterado após as operações.

As operações são:

- TRANSPOSTA X Y // faz a transposta de X e guarda em Y;
- SOMA X Y Z // Soma X e Y e guarda em Z;
- MULTI_ELEM X Y Z // Multiplica X e Y elemento a elemento e guarda em Z;
- MULTI_MAT X Y Z // Multiplica X e Y e guarda em Z;
- BORDAS X Y// Define as bordas de X e guarda em Y;

Saída

A saída é formada pela impressão da imagem resultante de cada operação com um espaço em branco (" ") entre píxeis até depois do último píxel da linha e pulando uma linha antes de começar a imagem.

Exemplos

Exemplo 1:

Entrada

```
9
2
TRANSPOSTA A B
SOMA A B B
```

Saída

```
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 0 0 0 1 1 1
1 1 1 0 0 0 1 1 1
1 1 1 0 0 0 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
```

Exemplo 3:

Entrada

```
9
5
TRANSPOSTA A B
SOMA A B A
BORDAS B B
MULTI_ELEM A B B
BORDAS B A
```

Saída

```
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 0 0 0 1 1 1
1 1 1 0 0 0 1 1 1
1 1 1 0 0 0 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1

1 1 1 1 1 1 1 1 1
1 0 0 0 0 0 0 0 1
1 1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0
```

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1
1	0	0	0	0	0	0	0	0	1
1	1	1	1	1	1	1	1	1	1

Regras e Avaliação

Seu código será avaliado não apenas pelos testes do CodePost, mas também pela qualidade. Dentre os critérios subjetivos de qualidade de código iremos analisar: o uso apropriado de funções, de comentários, e de structs; a escolha de bons nomes de funções, variáveis e de structs e seus campos; o correto uso de Tipos Abstratos de Dados e a correta separação em vários arquivos; a ausência de diversos trechos de código repetidos, e o tempo de execução e uso de memória dos algoritmos projetados. Note, porém, que essa não é uma lista exaustiva, pois outros critérios podem ser analisados dependendo do código apresentado visando mostrar ao aluno como o código poderia ser melhor.

Submissão

Você deverá submeter no CodePost, na tarefa Lab 2 - Matrizes, um arquivo com o nome `lab02.c`. Após o prazo estabelecido para a atividade, será aberta uma tarefa Lab 2 - Segunda Chance, com prazo de entrega até o fim do semestre.