

Estruturas de Dados I

2021

MATRIZES

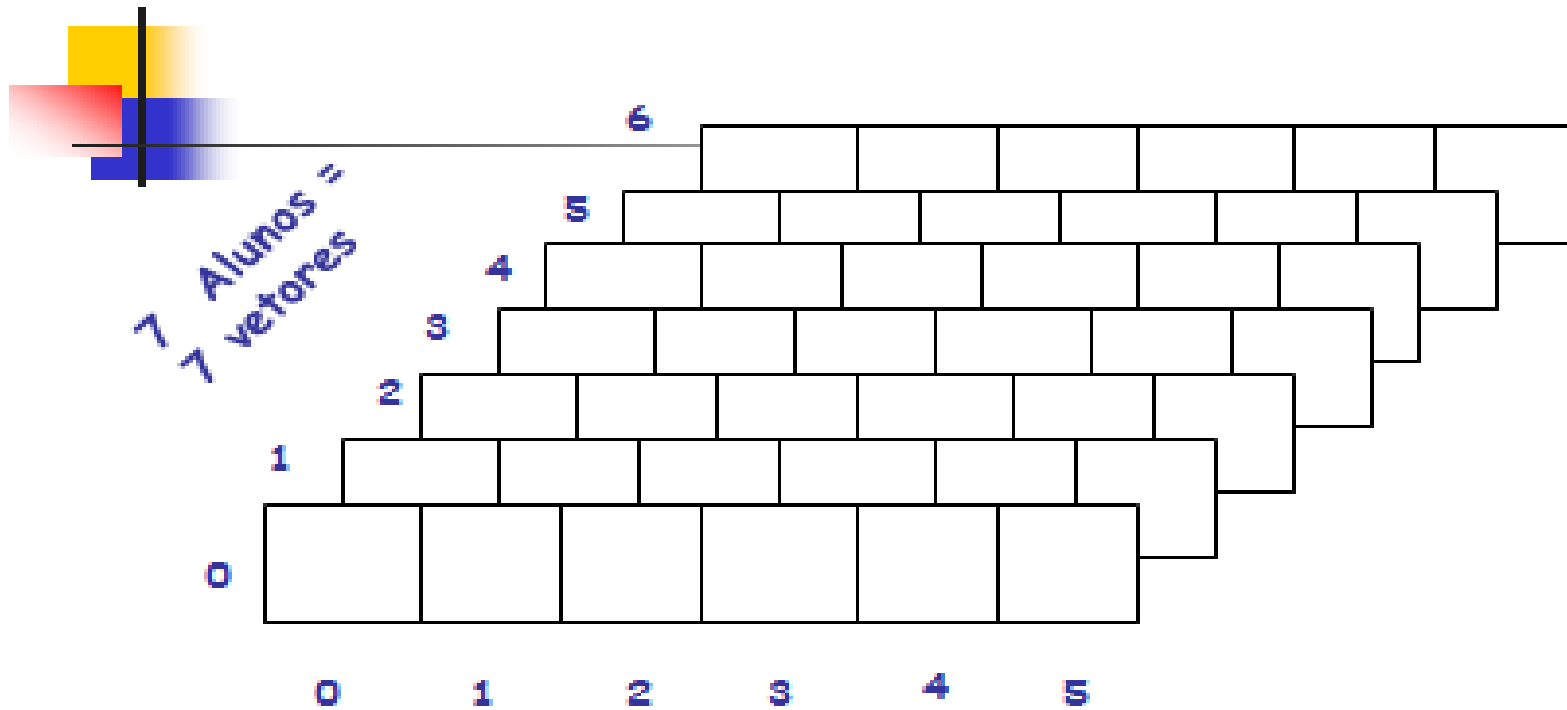
mauricioow@gmail.com

mauricio.wieler@mtac.org.br

# Recordando: Vetores

- Vetores são matrizes unidimensionais;
- Conjunto de vários elementos do mesmo tipo.;
- Armazenamento de dados seqüencial (tamanho  $[L]$  constante);
- Um elemento é acessado por meio de um  $[L]$  índice (posição);
- Exemplo: `float vetor[tamanho];`

## Vários vetores de mesmo tipo --> matriz de vetores



5 Notas + Média = 6 elementos por vetor



# Matrizes (*Arrays*)

- Generalização do conceito de vetores;
- Em C podemos criar matrizes multidimensionais;
- A forma mais simples e comum de matriz multidimensional é a matriz bidimensional.
- Uma matriz bidimensional nada mais é do que um vetor de vetores;
- Portanto, tendo aprendido como usar vetor estamos prontos para trabalhar com matrizes.

Cada elemento desta matriz necessita de dois índices para ser especificado --> notas é portanto uma matriz bidimensional

```
notas[3] [1]
```



Outra forma de apresentação de uma matriz bidimensional:

notas

notas[0][0]	notas[0][1]	...	notas[0][5]
notas[1][0]	notas[1][1]	...	notas[1][5]
notas[2][0]	notas[2][1]	...	notas[2][5]
	.		
	.		
	.		
notas[6][0]	notas[6][1]	...	notas[6][5]

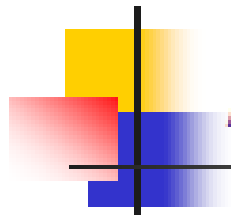
# Matrizes Bidimensionais: Sintaxe em C

```
tipo_da_variável nome_da_variável [M] [N]
```

**M:** número de linhas da matriz

**N:** número de colunas da matriz

**Bytes alocados:**  $M \times N \times \text{bytes do tipo}$



# Arranjos bidimensionais em C

---

- Primeiro índice : **linha;**
- Segundo índice : **coluna.**



**Exemplo:** Declarar uma **matriz bidimensional** de inteiros **mtx** de tamanho 10 x 20.

```
int mtx[10][20];
```

**Bytes alocados:**  $10 \times 20 \times 4 = 800$  bytes.

# Matrizes em C: Declaração

**Exemplo:** Declarar uma **matriz bidimensional** de inteiros **mtx** de tamanho 10 x 20.

```
int mtx[10][20];
```

**Limites da Matriz:** Nesse exemplo, os índices das linhas variam de 0 a 9 e o das colunas de 0 a 19.

# Exemplo: Matriz

```
#include <stdio.h>

main() {
    int mtx[3][4], i, j, k = 1;

    for (i=0; i<3; i++) {
        for (j=0; j<4; j++)
            mtx[i][j] = k++;
    }

    for (i=0; i<3; i++) {
        for (j=0; j<4; j++)
            printf("%4d", mtx[i][j]);
        printf("\n");
    }
}
```

---

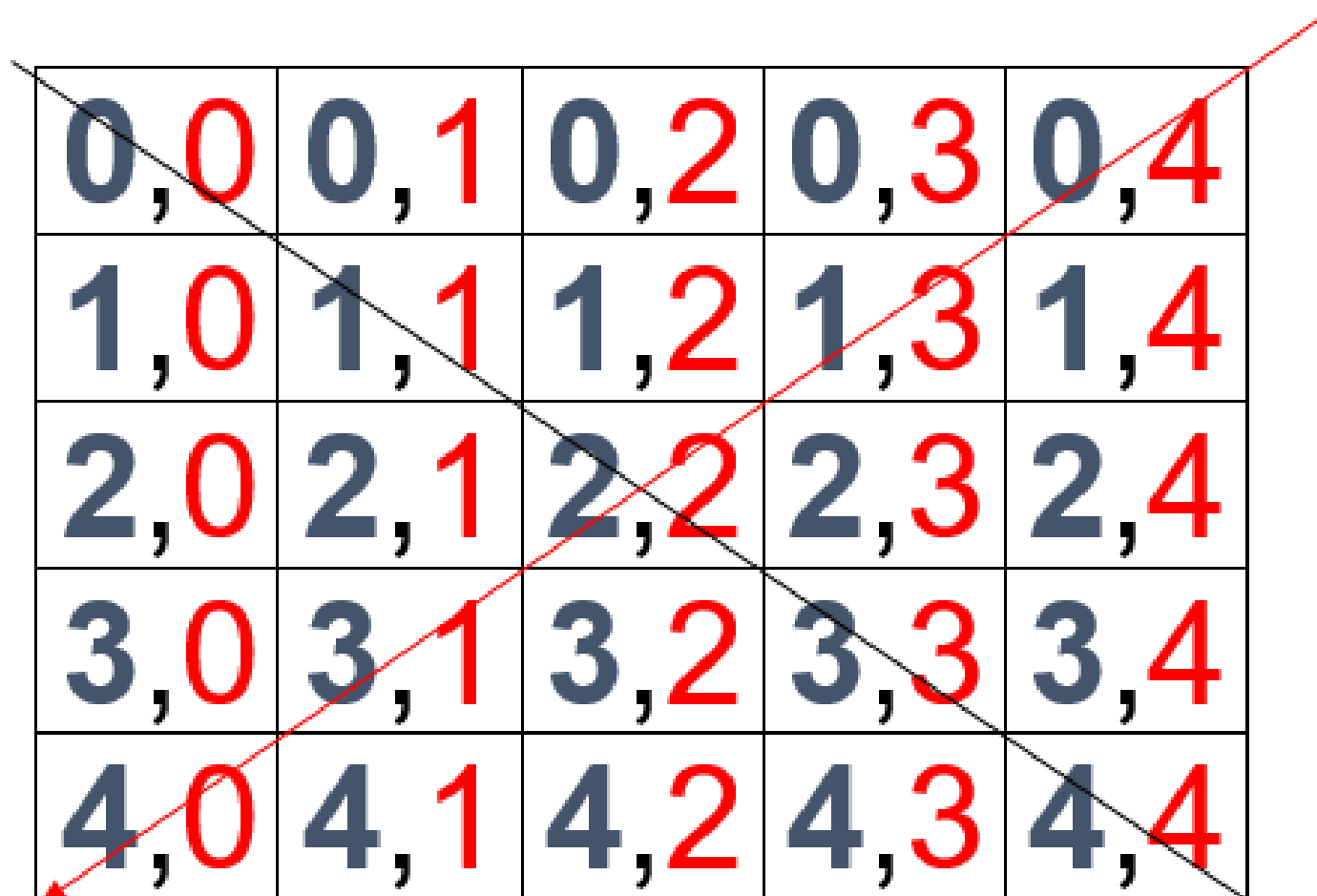
Preencher ou ler uma matriz no C o índice mais à direita varia mais rapidamente que o índice à esquerda.

# Exemplo: Esquematicamente

**mtx**

<b>mtx[0]</b>	→	mtx[0][0]=1	mtx[0][1]=2	mtx[0][2]=3	mtx[0][3]=4
<b>mtx[1]</b>	→	mtx[1][0]=5	mtx[1][1]=6	mtx[1][2]=7	mtx[1][3]=8
<b>mtx[2]</b>	→	mtx[2][0]=9	mtx[2][1]=10	mtx[2][2]=11	mtx[2][3]=12

# **Exercícios de Matrizes**



0,0	0,1	0,2	0,3	0,4
1,0	1,1	1,2	1,3	1,4
2,0	2,1	2,2	2,3	2,4
3,0	3,1	3,2	3,3	3,4
4,0	4,1	4,2	4,3	4,4

Diagonal Principal ( $i = j$ )

Diagonal Secundária ( $i + j = \text{tamanho da matriz} - 1$ )

Em azul = Linhas (i)

Em vermelho = Colunas (j)

No exemplo, o tamanho da matriz (por ser quadrada) é 5

1. Crie um programa em C que leia os elementos de uma matriz inteira de 5x5 e escreva os elementos da diagonal principal.

$$MAT = \begin{pmatrix} 2 & 5 & -1 & 7 & 3 \\ 1 & 4 & 6 & 9 & 0 \\ -3 & 8 & 12 & 5 & 4 \\ 3 & 0 & -3 & 4 & 1 \\ 6 & 2 & 1 & 5 & 8 \end{pmatrix}$$

Deve imprimir na tela:

2 4 12 4 8

(deve imprimir como diagonal), ou seja:

2

4

12

4

8

```
#include <stdio.h>
#include <conio.h>
main()
{
    int MAT[5][5], i, j;
    printf("\nDigitando os elementos da matriz 5x5: ");
    //Rotina de leitura
    for (i=0;i<5;i++)
    {
        for (j=0;j<5;j++)
        {
            printf("\nDigite MAT[%d][%d]: ",i,j);
            scanf("%d",&MAT[i][j]);
        }
    }
    printf("\n\nDiagonal Principal da Matriz:\n\n");
    for (i=0;i<5;i++){
        for (j=0;j<5;j++)
        {
            if(i==j)
                printf("%d\t",MAT[i][j]);
            else
                printf("\t");
        }
        printf("\n");
    }
    getch();
}
```



2. Crie um programa em C que leia os elementos de uma matriz inteira de 5x5 e escreva todos os elementos, exceto os elementos da diagonal principal.

$$MAT = \begin{pmatrix} 2 & 5 & -1 & 7 & 3 \\ 1 & 4 & 6 & 9 & 0 \\ -3 & 8 & 12 & 5 & 4 \\ 3 & 0 & -3 & 4 & 1 \\ 6 & 2 & 1 & 5 & 8 \end{pmatrix}$$

E deve escrever na tela:

$$\begin{pmatrix} & 5 & -1 & 7 & 3 \\ 1 & & 6 & 9 & 0 \\ -3 & 8 & & 5 & 4 \\ 3 & 0 & -3 & & 1 \\ 6 & 2 & 1 & 5 & \end{pmatrix}$$

```
#include <stdio.h>
#include <conio.h>
main()
{
    int MAT[5][5], i, j;
    printf("\nDigitando os elementos da matriz 5x5: ");
    for (i=0;i<5;i++)
    {
        for (j=0;j<5;j++)
        {
            printf("\nDigite MAT[%d][%d]: ",i,j);
            scanf("%d",&MAT[i][j]);
        }
    }
    printf("\n\nMatriz Impressa sem a Diagonal Principal:\n\n");
    for (i=0;i<5;i++){
        for (j=0;j<5;j++)
        {
            if(i!=j)
            {
                printf("%d\t",MAT[i][j]);
            }
            else
            {
                printf("\t");
            }
        }
        printf("\n");
    }
    getch();
}
```

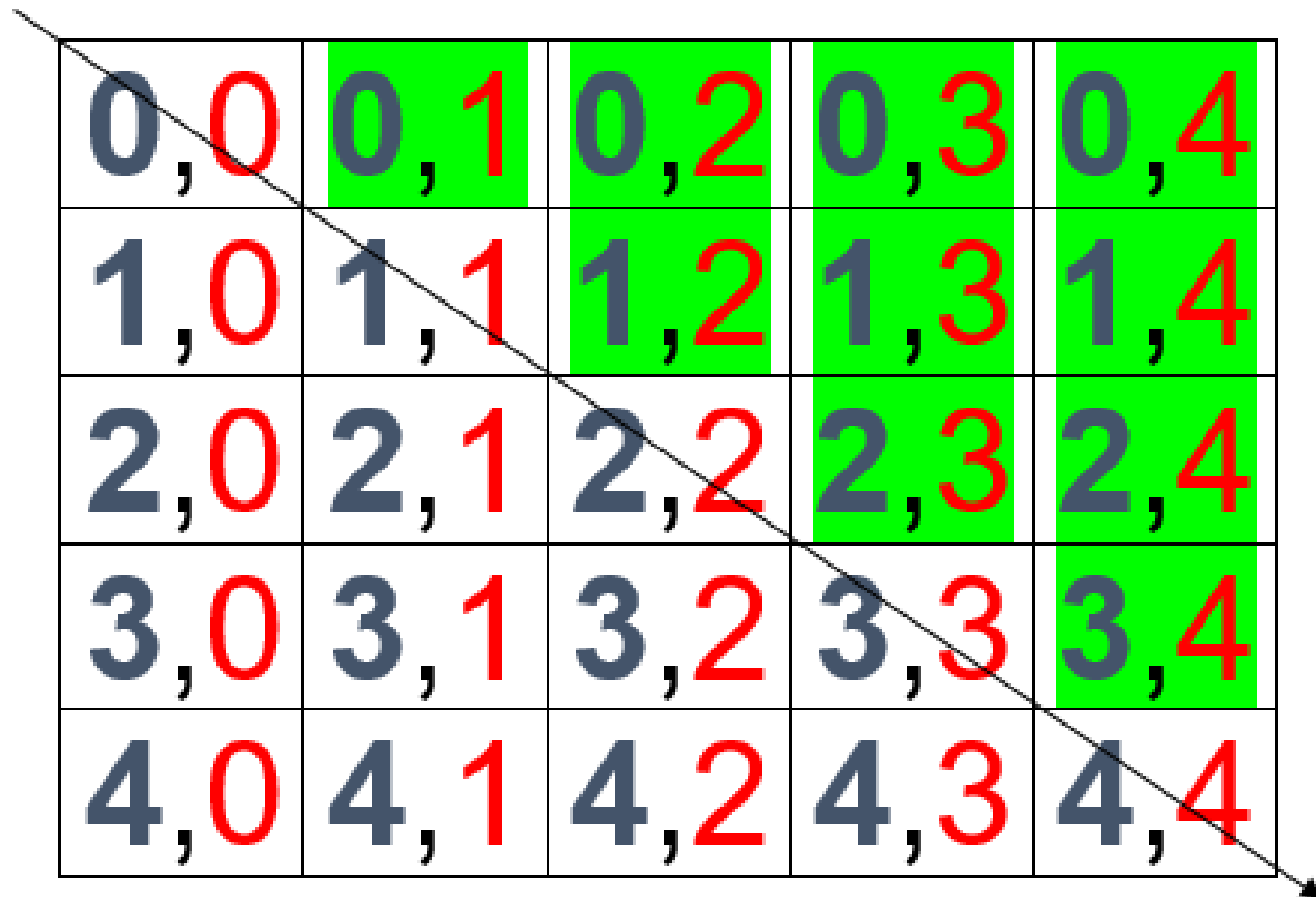
3. Crie um programa em C que leia os elementos de uma  $[SEP]$ matriz inteira de 5x5 e escreva todos os elementos que estão  $[SEP]$ acima da diagonal principal:

$$MAT = \begin{pmatrix} 2 & 5 & -1 & 7 & 3 \\ 1 & 4 & 6 & 9 & 0 \\ -3 & 8 & 12 & 5 & 4 \\ 3 & 0 & -3 & 4 & 1 \\ 6 & 2 & 1 & 5 & 8 \end{pmatrix}$$

E deve escrever na tela:

$$\begin{pmatrix} & 5 & -1 & 7 & 3 \\ & & 6 & 9 & 0 \\ & & & 5 & 4 \\ & & & & 1 \end{pmatrix}$$

No exemplo, o tamanho da matriz (por ser quadrada) é 5

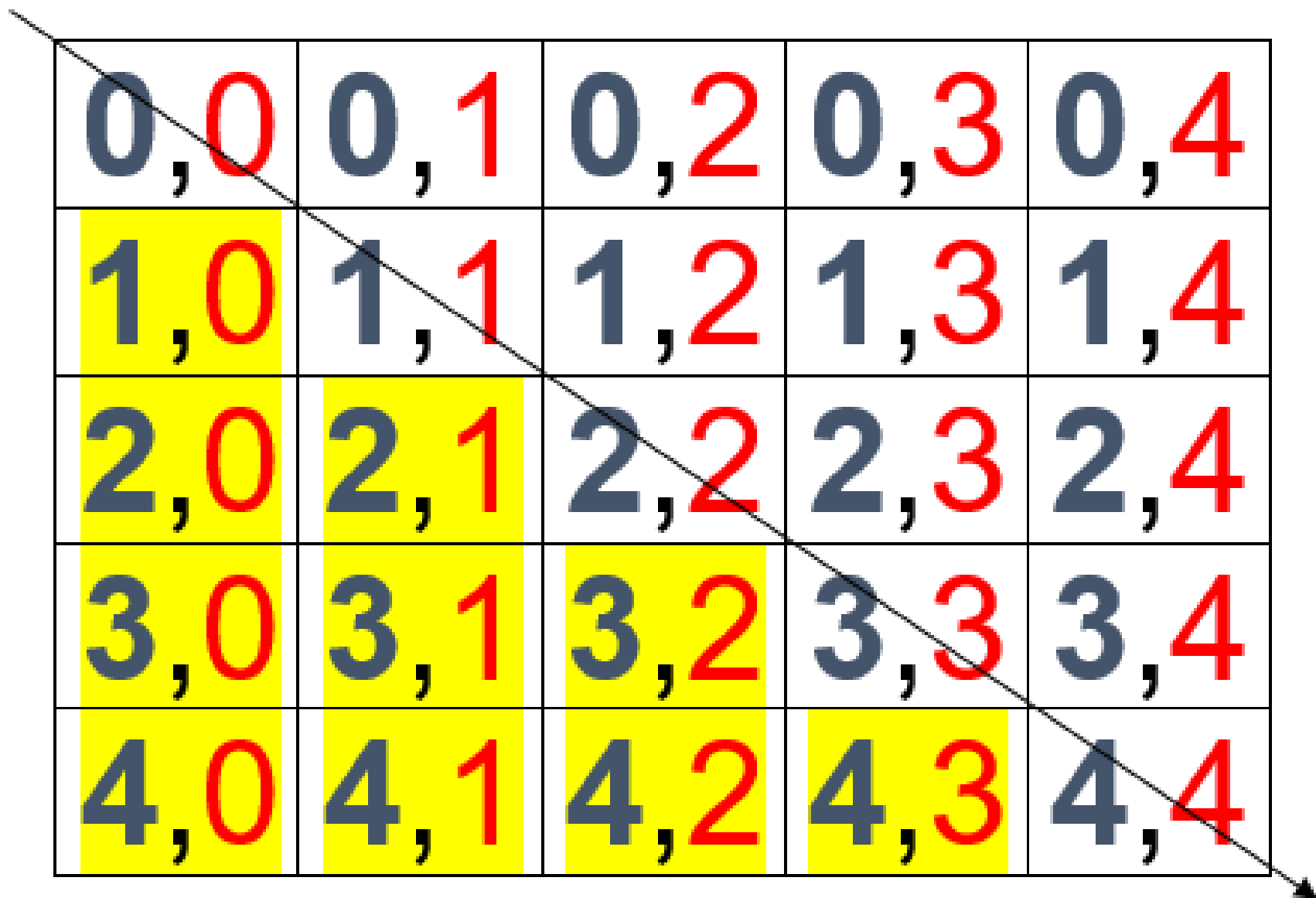


The diagram shows a 5x5 matrix with a diagonal arrow pointing from the top-left to the bottom-right. The cells are labeled with row and column indices (i, j). The cells where i < j are highlighted in green, representing the upper triangular part of the matrix.

0,0	0,1	0,2	0,3	0,4
1,0	1,1	1,2	1,3	1,4
2,0	2,1	2,2	2,3	2,4
3,0	3,1	3,2	3,3	3,4
4,0	4,1	4,2	4,3	4,4

**MATRIZ TRIANGULAR SUPERIOR**

Acima da diagonal principal  $\Rightarrow i < j$



0,0	0,1	0,2	0,3	0,4
1,0	1,1	1,2	1,3	1,4
2,0	2,1	2,2	2,3	2,4
3,0	3,1	3,2	3,3	3,4
4,0	4,1	4,2	4,3	4,4

## MATRIZ TRIANGULAR INFERIOR

Abaixo da diagonal principal  $\Rightarrow i > j$

```
#include <stdio.h>
#include <conio.h>
main()
{
    int MAT[5][5], i, j;
    printf("\nDigitando os elementos da matriz 5x5: ");
    for (i=0;i<5;i++)
    {
        for (j=0;j<5;j++)
        {
            printf("\nDigite MAT[%d][%d]: ",i,j);
            scanf("%d",&MAT[i][j]);
        }
    }
    printf("\n\nMatriz Acima da Diagonal Principal:\n\n");
    for (i=0;i<5;i++){
        for (j=0;j<5;j++)
        {
            if(i<j)
                printf("%d\t",MAT[i][j]);
            else
                printf("\t");
        }
        printf("\n");
    }
    getch();
}
```

4. Crie um programa em C que leia os elementos de uma matriz inteira de 5x5 e imprima a soma dos elementos que estão acima da diagonal principal:

$$MAT = \begin{pmatrix} 2 & 5 & -1 & 7 & 3 \\ 1 & 4 & 6 & 9 & 0 \\ -3 & 8 & 12 & 5 & 4 \\ 3 & 0 & -3 & 4 & 1 \\ 6 & 2 & 1 & 5 & 8 \end{pmatrix}$$

E deve escrever na tela:

$$\text{Soma} = 5 + (-1) + 7 + 3 + 6 + 9 + 0 + 5 + 4 + 1 = 39.$$

```
#include <stdio.h>
#include <conio.h>
main()
{
    int MAT[5][5], i, j, soma=0;
    printf("\nDigitando os elementos da matriz 5x5: ");
    for (i=0;i<5;i++)
        {
            for (j=0;j<5;j++)
            {
                printf("\nDigite MAT[%d][%d]: ",i,j);
                scanf("%d",&MAT[i][j]);
                soma+=MAT[i][j];
            }
        }
    printf("\n\nSoma dos Elementos da Matriz Acima da Diagonal Principal:%d",soma);
    for (i=0;i<5;i++){
        for (j=0;j<5;j++)
        {
            if(i<j)
                printf("%d\t",MAT[i][j]);
            else
                printf("\t");
        }
        printf("\n");
    }
    getch();
}
```



5. Crie um programa em C que leia os elementos de uma matriz inteira de 5x5 e escreva os elementos da diagonal secundária.

$$MAT = \begin{pmatrix} 2 & 5 & -1 & 7 & 3 \\ 1 & 4 & 6 & 9 & 0 \\ -3 & 8 & 12 & 5 & 4 \\ 3 & 0 & -3 & 4 & 1 \\ 6 & 2 & 1 & 5 & 8 \end{pmatrix}$$

E deve escrever na tela:

3  
9  
12  
0  
6

```
#include <stdio.h>
#include <conio.h>
main()
{
    int MAT[5][5], i, j;
    printf("\nDigitando os elementos da matriz 5x5: ");
    for (i=0;i<5;i++)
    {
        for (j=0;j<5;j++)
        {
            printf("\nDigite MAT[%d][%d]: ",i,j);
            scanf("%d",&MAT[i][j]);
        }
    }
    printf("\n\nDiagonal Secundária:\n\n");
    for (i=0;i<5;i++){
        for (j=0;j<5;j++)
        {
            if(i+j==4)
                printf("%d\t",MAT[i][j]);
            else
                printf("\t");
        }
        printf("\n");
    }
    getch();
}
```

6. Crie um programa em C que leia os elementos de uma matriz inteira de 5x5 e imprima:

- Toda a matriz.
- A raiz quadrada da soma dos quadrados de todos os números ímpares localizados abaixo da diagonal secundária.

$$MAT = \begin{pmatrix} 2 & 5 & -1 & 7 & 3 \\ 1 & 4 & 6 & 9 & 0 \\ -3 & 8 & 12 & 5 & 4 \\ 3 & 0 & -3 & 4 & 1 \\ 6 & 2 & 1 & 5 & 8 \end{pmatrix}$$

Deve imprimir na tela:

$$S = \sqrt{(5)^2 + (-3)^2 + (1)^2 + (1)^2 + (5)^2} = \sqrt{61} = 7,18$$

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
main()
{
    float MAT[5][5],soma_quad=0,valor=0;
    int i,j;
    printf("\nDigitando os elementos da matriz 5x5: ");
    for (i=0;i<5;i++)
    {
        for (j=0;j<5;j++)
        {
            printf("\nDigite MAT[%d][%d]: ",i,j);
            scanf("%f",&MAT[i][j]);
        }
    }
    printf("\n\nImpressao da Matriz:\n\n");
    for (i=0;i<5;i++){
        for (j=0;j<5;j++)
        {
            printf("%.0ft",MAT[i][j]);
            if(i>j){
                soma_quad += MAT[i][j]*MAT[i][j];
                valor=sqrt(soma_quad);
            }
        }
        printf("\n");
    }

    printf("Valor da raiz da soma dos quadrados da diag sec: %f",valor);
    getch();
}
```

7. Crie um programa em C que leia os elementos de duas matrizes inteiras A e B, de 5x5 e imprima a matriz SOMA, que será a soma das duas matrizes, elemento a elemento, como segue:

$$A = \begin{pmatrix} 2 & 5 & -1 & 7 & 3 \\ 1 & 4 & 6 & 9 & 0 \\ -3 & 8 & 12 & 5 & 4 \\ 3 & 0 & -3 & 4 & 1 \\ 6 & 2 & 1 & 5 & 8 \end{pmatrix} \quad B = \begin{pmatrix} 5 & 7 & 9 & 1 & 7 \\ 2 & 0 & 1 & 1 & 2 \\ -2 & 1 & 2 & 6 & -4 \\ 7 & 4 & 4 & 3 & 2 \\ 0 & 3 & 4 & 1 & 1 \end{pmatrix}$$

Deve imprimir na tela:

$$SOMA = \begin{pmatrix} 7 & 12 & 0 & 8 & 10 \\ 3 & 4 & 7 & 10 & 2 \\ -5 & 9 & 14 & 11 & 0 \\ 10 & 4 & 1 & 7 & 3 \\ 6 & 5 & 5 & 6 & 9 \end{pmatrix}$$

8. Crie um programa em C que leia uma matriz  $A_{2 \times 3}$  e gere e imprima a matriz transposta  $A_t$ , sabendo que a matriz transposta é gerada trocando linha por coluna, como mostra o exemplo:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix} \Rightarrow A_t = \begin{pmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{pmatrix}$$

9. Crie um programa em C que preencha uma matriz 3x3 com números inteiros e mostre uma mensagem dizendo se a matriz digitada é simétrica.

Uma matriz se diz simétrica se  $A(i,j) = A(j,i)$

$$A_{3 \times 3} = \begin{bmatrix} 3 & 7 & 9 \\ 7 & 4 & 6 \\ 9 & 6 & 2 \end{bmatrix}$$

é uma matriz simétrica.

```
#include <stdio.h>
#include <conio.h>
main()
{
    int MAT[3][3],l,c;
    int simetria;
    printf("\nDigitando os elementos da matriz 3x3: ");
    for (l=0;l<3;l++)
    {
        for (c=0;c<3;c++)
        {
            printf("\nDigite MAT[%d][%d]: ",l,c);
            scanf("%d",&MAT[l][c]);
        }
    }
    simetria=0;//verdadeiro
    printf("\n\nImpressao da Matriz:\n\n");
    for (l=0;l<3;l++){
        for (c=0;c<3;c++){
            {
                printf("%d\t",MAT[l][c]);
                if(MAT[l][c]!=MAT[c][l]){
                    simetria=1;//falso
                }
            }
        }
        printf("\n");
    }
    if(simetria==0)
        printf("\nMatriz Simetrica");
    else
        printf("\nMatriz Assimetrica");
    getch();
}
```



10. Crie um programa em C que leia uma matriz  $A_{4 \times 5}$  e calcule e imprima a soma de todos os seus elementos:

$$A = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 16 & 17 & 18 & 19 & 20 \end{pmatrix}$$

Deve imprimir:

$$\text{Soma} = 1 + 2 + 3 + 4 + \dots + 19 + 20 = 210.$$

