



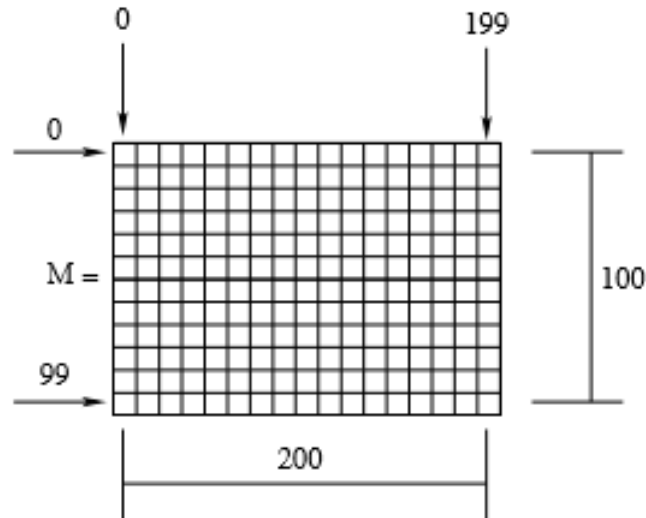
Matrizes

Prof. Lilian Berton



São José dos Campos, 2019

Matrizes

- Matrizes são estruturas - como ilustrado na figura abaixo, utilizadas para armazenar dados de um mesmo tipo: int, char, float ou double.
- São usados índices para acessar uma linha e uma coluna de uma matriz.
 - Os índices são números naturais.
 - O índice da primeira linha é sempre zero.
 - O índice da primeira coluna é sempre zero.



Declaração de matrizes

- A declaração de uma matriz é feita da seguinte forma:
 - `<tipo_da_matriz> <nome_da_matriz> [<numero_de_linhas>][<numero_de_colunas>];`
- Exemplos:
- `int M[100][200];`  100 é o número de linhas!
200 é o número de colunas!
- A declaração acima aloca uma matriz com 100 linhas e 200 colunas na memória. Cada casa da matriz guarda um int.
- `char M2[20][30];`  20 é o número de linhas!
30 é o número de colunas!
- A declaração acima aloca uma matriz com 20 linhas e 30 colunas na memória. Cada casa da matriz guarda um char.

Exemplo matrizes

```
# include <stdio.h>
```

```
int main () {
```

```
    int A[10][80], lin , col;
```

```
    A[1][2] = 4;          /* casa da linha 1 e coluna 2 recebe o inteiro 4 */
```

```
    lin = 2; col = 3;
```

```
    A[lin][col] = 5;      /* casa da linha 2 e coluna 3 recebe o inteiro 5 */
```

```
    A[A[lin-1][col-1] - 1][A[lin][col]] = 10; /* vc saberia dizer qual casa da
matriz A recebe o inteiro 10? */
```

```
    return 0;
```

```
}
```

	0	1	2	3	4	5		78	79
0	?	?	?	?	?	?	...	?	?
1	?	?	?	?	?	?	...	?	?
2	?	?	?	?	?	?	...	?	?
3	?	?	?	?	?	?	...	?	?
⋮	⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	⋮
9	?	?	?	?	?	?	...	?	?

Exemplo matrizes

```
# include <stdio.h>
```

```
int main () {
```

```
    int A[10][80], lin , col;
```

```
    A[1][2] = 4;      /* casa da linha 1 e coluna 2 recebe o inteiro 4 */
```

```
    lin = 2; col = 3;
```

```
    A[lin][col] = 5;   /* casa da linha 2 e coluna 3 recebe o inteiro 5 */
```

```
    A[A[lin-1][col-1] - 1][A[lin][col]] = 10; /* vc saberia dizer qual casa da  
    matriz A recebe o inteiro 10? */
```

```
    return 0;
```

```
}
```

	0	1	2	3	4	5		78	79
0	?	?	?	?	?	?	...	?	?
1	?	?	4	?	?	?	...	?	?
2	?	?	?	?	?	?	...	?	?
3	?	?	?	?	?	?	...	?	?
⋮	⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	⋮
9	?	?	?	?	?	?	...	?	?

Exemplo matrizes

```
# include <stdio.h>
```

```
int main () {
```

```
    int A[10][80], lin , col;
```

```
    A[1][2] = 4;          /* casa da linha 1 e coluna 2 recebe o inteiro 4 */
```

```
    lin = 2; col = 3;
```

```
    A[lin][col] = 5;      /* casa da linha 2 e coluna 3 recebe o inteiro 5 */
```

```
    A[A[lin-1][col-1] - 1][A[lin][col]] = 10; /* vc saberia dizer qual casa da
matriz A recebe o inteiro 10? */
```

```
    return 0;
```

```
}
```

	0	1	2	3	4	5		78	79
0	?	?	?	?	?	?	...	?	?
1	?	?	4	?	?	?	...	?	?
2	?	?	?	5	?	?	...	?	?
3	?	?	?	?	?	?	...	?	?
⋮	⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	⋮
9	?	?	?	?	?	?	...	?	?

Exemplo matrizes

```
# include <stdio.h>
```

```
int main () {
```

```
    int A[10][80], lin , col;
```

```
    A[1][2] = 4;          /* casa da linha 1 e coluna 2 recebe o inteiro 4 */
```

```
    lin = 2; col = 3;
```

```
    A[lin][col] = 5;      /* casa da linha 2 e coluna 3 recebe o inteiro 5 */
```

```
    A[A[lin-1][col-1] - 1][A[lin][col]] = 10; /* vc saberia dizer qual casa da
    matriz A recebe o inteiro 10? */
```

```
    return 0;
```

```
}
```

	0	1	2	3	4	5		78	79
0	?	?	?	?	?	?	...	?	?
1	?	?	4	?	?	?	...	?	?
2	?	?	?	5	?	?	...	?	?
3	?	?	?	?	?	10	...	?	?
⋮	⋮	⋮	⋮	⋮	⋮	⋮	...	⋮	⋮
9	?	?	?	?	?	?	...	?	?

A[lin-1][col-1]=A[1][2]=4

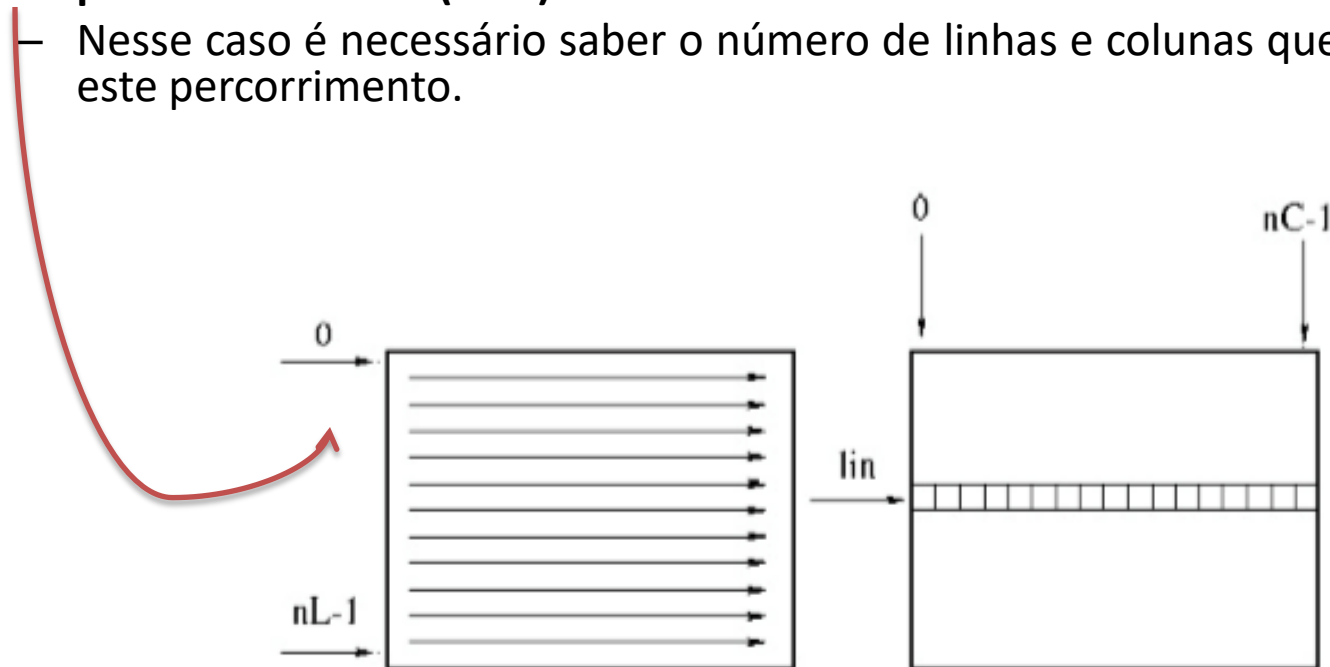
A[lin][col]=A[2][3]=5.

A[A[lin-1][col-1]-1][A[lin][col]]=

A[4-1][5]=A[3][5]=10

Percorrendo matrizes

- Percorrer uma matriz significa visitar cada elemento da matriz (ou um subconjunto de elementos) de casa em casa em uma determinada ordem.
- Por exemplo:
 - **Percorrer todos os elementos de uma matriz retangular, linha a linha, a partir da linha 0 (zero), e para cada linha, visitar os elementos de cada coluna, a partir da coluna 0 (zero).**
 - Nesse caso é necessário saber o número de linhas e colunas que se deve fazer este percorrimento.



Percorrendo matrizes – uma linha

- Para percorrer uma linha da matriz A usamos **um comando de repetição (no caso, vamos usar o comando for)** com uma variável inteira col para o índice das colunas da matriz A:

```
int A[20][30];  
int lin, col, nL = 10, nC = 5, cont = 0;  
lin = 0;  
for (col=0; col < nC; col++) {  
    A[lin][col]=cont;  
    cont++;  
}
```

um usuário não necessariamente irá usar todas as linhas e colunas disponíveis da matriz. Deve-se sempre ter duas variáveis inteiras que diz quantas linhas e colunas da matriz estão sendo usadas (por exemplo, nL e nC associadas a matriz A)

	0	1	2	3	4
0	0	1	2	3	4
1					
2					
3					
4					
5					
6					
7					
8					
9					

Percorrendo matrizes – uma coluna

- Para percorrer uma coluna da matriz A usamos **um comando de repetição (no caso, vamos usar o comando for)** com uma variável inteira lin para o índice das linhas da matriz A:

```
int A[20][30];  
int lin, col, nL = 10, nC = 5, cont = 0;  
col = 0;  
for (lin=0; lin < nL; lin++) {  
    A[lin][col]=cont;  
    cont++;  
}
```

	0	1	2	3	4
0	0				
1	1				
2	2				
3	3				
4	4				
5	5				
6	6				
7	7				
8	8				
9	9				

Percorrendo matrizes – linhas e colunas

- Para percorrer completamente a matriz A (isto é, as nL linhas e as nC colunas) **usamos dois comandos de repetição (no caso, vamos usar o comando for)** com duas variáveis inteiras lin e col, um para percorrer as linhas e a outra para percorrer as colunas da matriz A:

```
int A[20][30];  
int lin, col, nL = 10, nC = 5, cont = 0;  
  
for (lin=0; lin < nL; lin++) {  
    for (col=0; col < nC; col++) {  
        A[lin][col] = cont;  
        cont++;  
    }  
}
```

	0	1	2	3	4
0	0	1	2	3	4
1	5	6	7	8	9
2	10	11	12	13	14
3	15	16	17	18	19
4	20	21	22	23	24
5	25	26	27	28	29
6	30	31	32	33	34
7	35	36	37	38	39
8	40	41	42	43	44
9	45	46	47	48	49

Percorrendo matrizes – diagonal principal

- Como na **diagonal principal** temos que a linha é igual a coluna, um padrão para percorrer a diagonal principal de A é usar dois comandos de repetição (no caso, vamos usar o comando for) com duas variáveis inteira lin e col para os índices das linhas e colunas da matriz A. Quando lin e col for igual indicaria a diagonal principal.

Código 1: usando dois for

```
int A[5][5];
int lin, col, cont = 0;
for (lin = 0; lin < 5; lin++) {
    for(col = 0; col < 5; col++) {
        if(lin == col) {
            A[lin][col] = cont;
            cont++;
        }
    }
}
```

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

Percorrendo matrizes – diagonal principal

- Como na **diagonal principal** temos que a linha é igual a **coluna**, outro padrão para percorrer a diagonal principal de A é usar um comando de repetição (no caso, vamos usar o comando for) com uma variável inteira lin para o índice das linhas e colunas da matriz A.

Código 2: usando um for

```
int A[5][5];
int lin, cont = 0;
for (lin=0; lin<5; lin++) {
    A[lin][lin] = cont;
    cont++;
}
```

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

Percorrendo matrizes – diagonal secundária

- Como na diagonal secundária temos que a soma da linha com a coluna é igual a $n-1$, um padrão para percorrer a diagonal secundária de A é usar dois comandos de repetição (no caso, vamos usar o comando for) com duas variáveis inteiras lin e col para o índice das linhas e colunas da matriz A, e quando a soma delas for igual $n-1$ indicaria a diagonal secundária:

Código 1 : usando dois for

```
int A[5][5];
int lin, col, cont = 0;
for (lin=0; lin<5 lin++) {
    for(col = 0; col < 5; col++) {
        if (lin + col == 5-1) {
            A[lin][col] = cont;
            cont++;
        }
    }
}
```

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

Percorrendo matrizes – diagonal secundária

- Como na diagonal secundária temos que a soma da linha com a coluna é igual a $n-1$ (para uma linha lin , a coluna deve ser $n-1-i$), outro padrão para percorrer a diagonal secundária de A é usar um comando de repetição (no caso, vamos usar o comando `for`) com uma variável inteira lin para o índice das linhas e colunas da matriz A :

Código 2: usando um for

```
int A[5][5];
int lin, cont = 0;
for (lin=0; lin<5; lin++) {
    A[lin][5 - 1 - lin] = cont;
    cont++;
}
```

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


Leitura de matrizes

```
1  # include <stdio.h>
2  # define MAX_L 100
3  # define MAX_C 200
4
5  int main () {
6      float A[MAX_L][MAX_C];
7      int lin, col, nL, nC;
8
9      printf ("Entre com 0<nL<%d: ", MAX_L);
10     scanf ("%d" &nL);
11
12     printf ("Entre com 0<nC<%d: ", MAX_C);
13     scanf ("%d" &nC);
14
15     /* percorrer a matriz A elemento a elemento
16      * colocando o valor lido pelo teclado */
17     for (lin=0; lin < nL; lin++) {
18         for (col=0; col < nC; col++) {
19             printf ("Entre com A[%d][%d] = ", lin, col);
20             scanf ("%f", &A[lin][col]);
21         }
22     }
23
24     return 0;
25 }
```

← scanf ("%f",&A[lin][col]);

Impressão de matrizes

```
1  # include <stdio.h>
2
3  # define MAX_L 100
4  # define MAX_C 200
5
6  int main () {
7      float A[MAX_L][MAX_C];
8      int lin, col, nL, nC;
9
10     printf ("Entre com 0<nL<%d: ", MAX_L);
11     scanf ("%d" &nL);
12
13     printf ("Entre com 0<nC<%d: ", MAX_C);
14     scanf ("%d" &nC);
15
16     /* percorrer a matriz A elemento a elemento
17      * imprimindo o valor de cada casa */
18     for (lin=0; lin<nL; lin++) {
19         for (col=0; col<nC; col++) {
20             printf ("%f ", A[lin][col]);
21         }
22         printf ("\n");
23     }
24
25     return 0;
26 }
```

 `printf ("%f",A[lin][col]);`

Matrizes como Parâmetro de Funções

- A forma de declaração de uma matriz como parâmetro de função é a mesma que vimos para declarar vetores, ou seja, indicando o nome da matriz, e entre colchetes o número de linhas e o número de colunas.

```
float soma_diagonal (float B[300][300], int n) {  
    int i;  
    float r = 0;  
    for (i=0; i<n; i++) {  
        r = r + B[i][i];  
    }  
    return r;  
}
```

Matrizes como Parâmetro de Funções

```
1      # define MAX 200
2
3
4      float f (float M[MAX][MAX]) {
5          float s;
6          /* declaração da função f */
7          ...
8          M[i][j] = 4;
9          ...
10         return s;
11     }
12
13     int main () {
14         float a, A[MAX][MAX]; /* declaração da variável a e da matriz A */
15         ...
16         /* outras coisas do programa */
17
18         a = f (A);    /* observe que a matriz é passada apenas pelo nome */
19
20         ...
21
22         return 0;
23     }
24
```

M aponta para A[0][0].

Nessa Linha 8, dentro da função f, estamos mudando o conteúdo da casa de linha i e coluna j da matriz A que foi passada por parâmetro para a função .

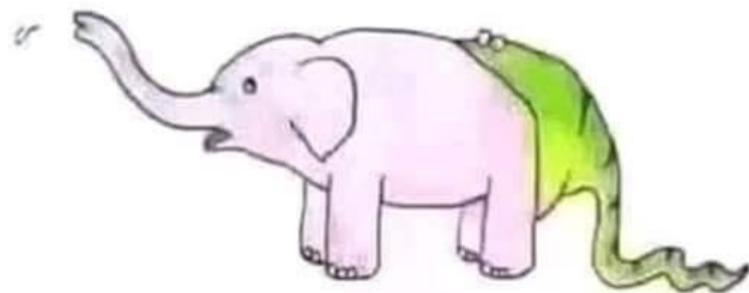
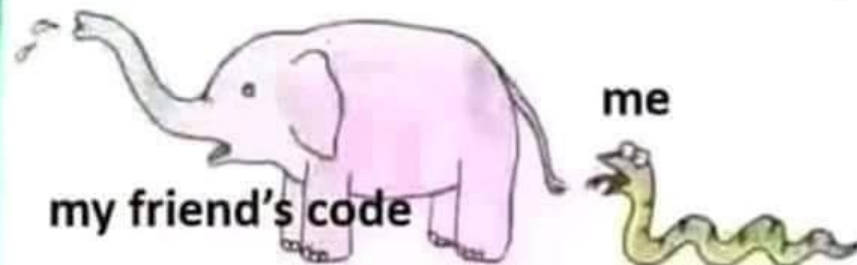
Erros comuns

- Ao desenvolver seus programas com matrizes, preste atenção aos seguintes detalhes:
- **Índices inválidos:** tome cuidado, especialmente dentro de um while ou for, de não utilizar índices negativos ou maiores que o tamanho máximo designado para as linhas e colunas da matriz.
- **A definição do tamanho das linhas e colunas da matriz** se faz na declaração da matriz. Isso significa que podemos estar “desperdiçando” algum espaço da memória por não estar usando todas as casas da matriz.

Exercícios

1. Escreva um programa que, dadas duas matrizes $A_{m \times n}$ e $B_{m \times n}$, calcula a matriz $C_{m \times n}$ que é **a soma de A e B**. Note que, para ler as matrizes, é necessário primeiro ler os seus tamanhos m, n .
2. Escreva um programa que, dadas duas matrizes $A_{m \times n}$ e $B_{n \times p}$, calcula a matriz $D_{m \times p}$ que é **o produto de A por B**. Note que, para ler as matrizes, é necessário primeiro ler os seus tamanhos m, n , e p .
3. Desenhe na tela um retângulo de 60 colunas (a partir da coluna 1 da tela) e 10 linhas (a partir da linha 1 da tela), sendo que a borda deste retângulo será formada pelo caractere '+'. Lembre que somente a primeira e última linha deverão ter todas as colunas preenchidas com o caractere '+'. As demais linhas (entre 2 e 9) só terão as colunas 1 e 60 preenchidas. A aparência deste retângulo deve ser parecida com a figura abaixo:





my code

恐龙是怎么来的

