

PCI- Matrices

Profa. Mercedes Gonzales Márquez

Matrizes (Estrutura composta homogênea bidimensional)

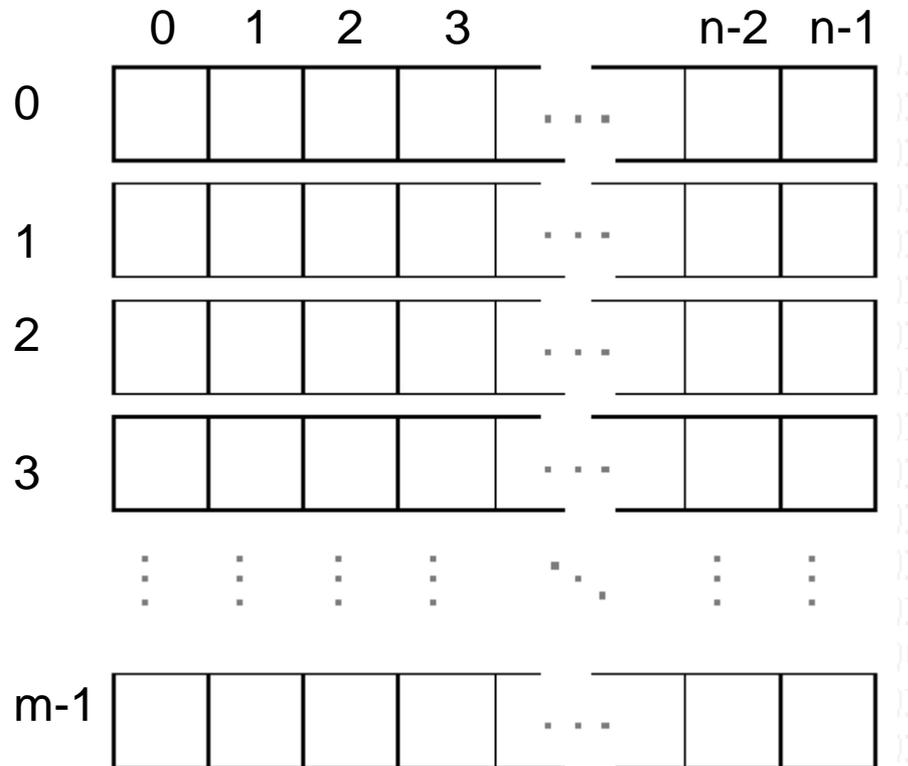
Uma matriz $m \times n$ representa um conjunto de m vetores de mesmo tamanho n .

- Declaração:

`<tipo> nome da matriz [m] [n]`

- Uma matriz possui $m * n$ elementos do tipo `<tipo>`.
- As linhas são numeradas de 0 a $m - 1$.
- As colunas são numeradas de 0 a $n - 1$.

Uma matriz de ***m*** linhas e ***n*** colunas



Estrutura composta homogênea multidimensional

- Há casos em que uma matriz é insuficiente para armazenar um conjunto de dados para um determinado programa. Nestes casos, é necessário definir uma estrutura de d -dimensões (Estrutura composta homogênea multidimensional).

- Declaração geral para d dimensões:

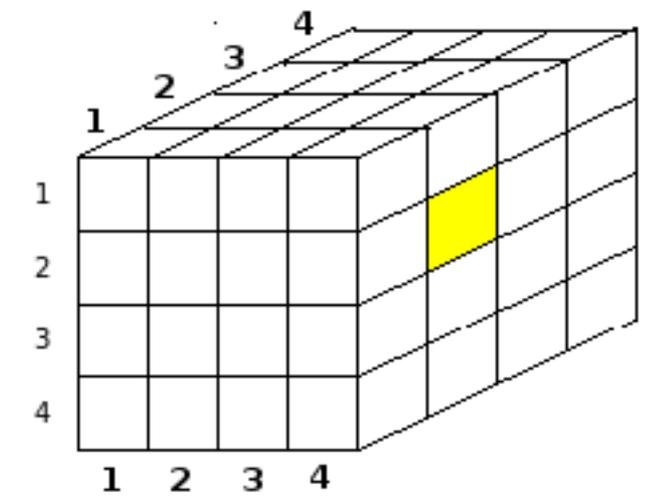
Tipo primitivo : variável[num_elem_ prim.dim]
[num_elem_seg_dim]...[num_elem_d_dim]

Estrutura composta homogênea multidimensional

Exemplos

- tridimensional:

```
int matriz[4][4][4]
```



Visão multidimensional

- Matriz bi-dimensional



Indicador : VENDAS (R\$)

Dimensões : ZONAS DE VENDAS E PRODUTOS

Visão multidimensional

- Matriz tri-dimensional

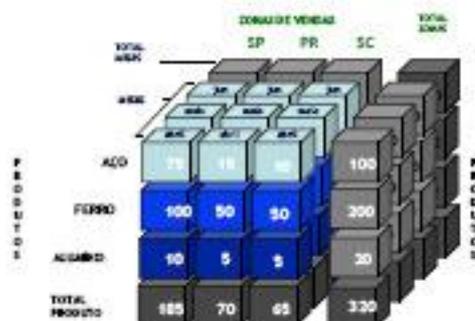


Indicador : VENDAS (R\$)

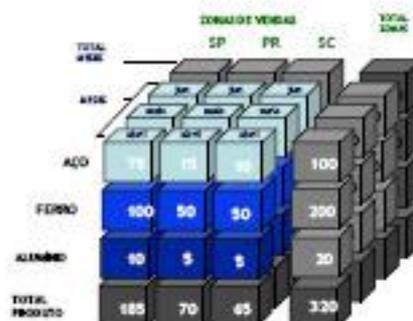
Dimensões : ZONAS DE VENDAS, PRODUTOS E MESES

Visão multidimensional

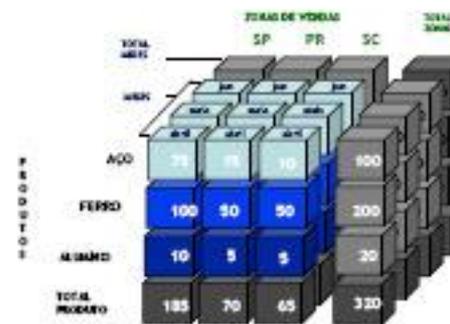
- Matriz multidimensional



Automobilística



Construção civil



Varejo



Estrutura Composta homogênea multidimensional

Analisando os gráficos que representam matrizes multidimensionais como você interpreta:

1. $M[2][2]$
2. $M[2][2][2]$
3. $M[2][2][2][2]$

Estrutura Composta homogênea multidimensional

- Para acessar um vetor, o inserimos em um único laço de repetição, fazendo com que haja variação em seu índice.
- Como em uma estrutura multidimensional temos mais de um índice, faz-se necessária a utilização de mais laços de repetição, no mínimo o número de dimensões da estrutura.
- As estruturas multidimensionais mais utilizadas são as bidimensionais (Matrizes).

Algumas Matrizes Especiais

-Matriz diagonal

Nesta matriz apenas os elementos da diagonal principal ou secundária são significativos. Exemplos:

$$A =$$

15.7	0	0	0
0	45.2	0	0
0	0	67.5	0
0	0	0	78.4

$$B =$$

0	0	0	89
0	0	67	0
0	3	0	0
45	0	0	0

Algumas Matrizes Especiais

-Matriz triangular

Nesse tipo de matriz, apenas os elementos da diagonal principal ou secundária e os abaixo (ou acima) possuem valores significativos.

15.7	0	0	0
44	45.2	0	0
66.5	17	67.5	0
88	23.9	90	78.4

Algumas Matrizes Especiais

-Matriz transposta

- É a matriz que resulta da troca de linhas por colunas em uma determinada matriz original.

$$A = \begin{bmatrix} 1 & 3 & -5 \\ 0 & -2 & 4 \\ 2 & 3 & 6 \end{bmatrix} \quad A^T = \begin{bmatrix} 1 & 0 & 2 \\ 3 & -2 & 3 \\ -5 & 4 & 6 \end{bmatrix}$$

- Uma matriz simétrica é toda matriz que é igual à sua transposta.

Exemplo 1:

Programa que realiza a leitura e a escrita de uma matriz quadrada.

Lendo uma matriz $n \times n$ do teclado:

```
/*Leitura*/  
for (i = 0; i < n; i++)  
    for (j = 0; j < n; j++) {  
        printf ("Matriz[%d][%d]: ", i, j);  
        scanf ("%d", &matriz[i][j]);  
    }
```

Exemplo 1

Escrevendo uma matriz $n \times n$ na tela:

```
/*Escrita*/  
for (i = 0; i < n; i++) {  
    for (j = 0; j < n; j++)  
        printf ("%d ", matriz[i][j]);  
    printf ("\n");  
}
```

Exemplo 2:

Programa que realiza a soma de duas matrizes $m \times n$ mat1 e mat2.

```
...  
printf("*** Somando os valores  
correspondentes das duas matrizes ***\n");  
for (i = 0; i < m; i++)  
    for (j = 0; j < n; j++)  
        mat3[i][j] = mat1[i][j] + mat2[i][j];
```

Exemplo 3:

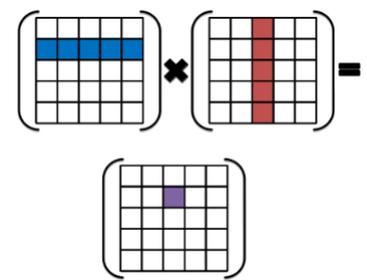
Elabore um programa que dada uma matriz 7×7 , calcule a sua matriz transposta e ainda diga se ela é simétrica ou não.

```
Algoritmo <matrizsimetrica>
inteiro:A[7][7],B[7][7],i,j,flag
inicio
  para (i ← 1; i<=7; i ←i+1)
    para (j ← 1; j<=7; j ←j+1)
      leia A[i][j]
      B[j][i] ← A[i][j]
    fim para
  fim para
fim para
```

```
flag ← 1
para (i ← 1; i<=7; i ←i+1)
  para (j ← 1; j<=7; j ←j+1)
    se (A[i][j]<>B[i][j])
      flag ← 0
    fim se
  fim para
fim para
Se (flag=1) então
  Escreva (“Simétrica”)
senão
  Escreva (“Não Simétrica”)
Fim se
fim
```

```
int i,j,n, A[7][7], B[7][7], simetrica=1;
scanf ("%d", &n);
for (i = 0; i < n; i++)
    for (j = 0; j < n; j++) {
        printf ("Matriz[%d][%d]: ", i, j);
        scanf ("%d", &A[i][j]);
        B[j][i]=A[i][j];
    }
printf("*** Matriz Original ***\n");
for (i = 0; i < n; i++) {
    for (j = 0; j < n; j++)
        printf ("%d ", A[i][j]);
    printf ("\n");
}
printf("*** Matriz Transposta ***\n");
for (i = 0; i < n; i++) {
    for (j = 0; j < n; j++)
        printf ("%d ", B[i][j]);
    printf ("\n");
}
```

```
printf("*** Verificando a Simetria da
Matriz A ***\n");
for (i = 0; i < n; i++) {
    for (j = 0; j < n; j++)
        if (A[i][j]!=B[i][j])
            simetrica=0;
}
if (simetrica)
    printf ("A matriz A eh simetrica");
else
    printf ("A matriz A nao eh simetrica");
}
```



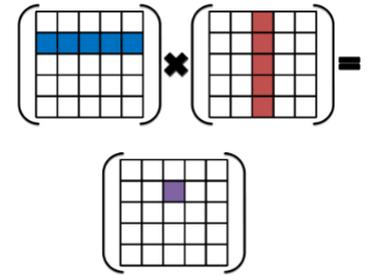
Matrizes - Produto

Exemplo4: Elabore um algoritmo que leia duas matrizes inteiras A e B de dimensão 3x3 e calcule em uma matriz R, a multiplicação delas.

$$A: \begin{array}{cccc} 4 & 6 & 2 & 1 \\ 9 & 0 & 0 & 2 \\ 8 & 7 & 3 & 9 \\ 1 & 2 & 3 & 4 \\ 0 & 1 & 0 & 1 \end{array}$$

$$B: \begin{array}{ccc} 2 & 3 & 4 \\ 5 & 0 & 0 \\ 0 & 7 & 7 \\ 1 & 0 & 9 \end{array}$$

$$\begin{aligned} c_{32} &= a_{31} \cdot b_{12} + a_{32} \cdot b_{22} + a_{33} \cdot b_{32} + a_{34} \cdot b_{42} \\ &= 8 \cdot 3 + 7 \cdot 0 + 3 \cdot 7 + 9 \cdot 0 \\ &= 45 \end{aligned}$$



Matrizes - Produto

Lembre-se que:

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} * \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} \\ R_{21} & R_{22} & R_{23} \\ R_{31} & R_{32} & R_{33} \end{bmatrix}$$

$$R_{11} = A_{11} * B_{11} + A_{12} * B_{21} + A_{13} * B_{31}$$

$$R_{12} = A_{11} * B_{12} + A_{12} * B_{22} + A_{13} * B_{32}$$

$$R_{13} = A_{11} * B_{13} + A_{12} * B_{23} + A_{13} * B_{33}$$

$$R_{21} = A_{21} * B_{11} + A_{22} * B_{21} + A_{23} * B_{31}$$

...

$$R_{32} = A_{31} * B_{12} + A_{32} * B_{22} + A_{33} * B_{32}$$

$$R_{33} = A_{31} * B_{13} + A_{32} * B_{23} + A_{33} * B_{33}$$

Exemplo 4:

```
Algoritmo <produtomatrizes>
inteiro:A[3][3],B[3][3],R[3][3],i,j,k
início
  para (i ← 1; i<=3; i ←i+1)
    para (j ← 1; j<=3; j ←j+1)
      leia (A[i][j],B[i][j])
    fim para
  fim para
  para (i ← 1; i<=3; i ←i+1)
    para (j ← 1; j<=3; j ←j+1)
      R[i][j] ← 0
      para (k ← 1; k<=3; k ←k+1)
        R[i][j] ←R[i][j]+A[i][k]* B[k][j]
      fim para
    fim para
  fim para
Fim
```

Exemplo 4:

```
do{
    printf ("Informe o numero de linhas
e colunas da matriz ");
    scanf ("%d", &n);
}while (n<1 || n>7);

printf("*** Lendo a Matriz A ***\n");
for (i = 0; i < n; i++)
    for (j = 0; j < n; j++) {
        printf ("A[%d][%d]: ", i, j);
        scanf ("%d", &A[i][j]);
    }
printf("* Lendo a Matriz B *\n");
for (i = 0; i < n; i++)
    for (j = 0; j < n; j++) {
        printf ("B[%d][%d]: ", i, j);
        scanf ("%d", &B[i][j]);
    }
```

```
for (i = 0; i < n; i++)
    for (j = 0; j < n; j++){
        C[i][j] = 0;
        for (k = 0; k < 3; k++)
            C[i][j] = C[i][j]+A[i][k]* B[k][j];
    }

printf("* Imprimindo a Matriz Produto C
*\n");
for (i = 0; i < n; i++) {
    for (j = 0; j < n; j++)
        printf ("%d ", C[i][j]);
    printf ("\n");
}
```

Exemplo 5:

Fazer um programa que:

- leia uma matriz inteira A de $M \times N$, onde os elementos de cada linha e os valores de M e N são fornecidos ($M \leq 20$, $N \leq 10$);
- imprima a matriz lida;
- calcule e imprima uma matriz modificada B ($M \times (N + 1)$), sendo que os elementos da (N+1)-ésima coluna são formados com o produto dos elementos da mesma linha - exercício 2.5.2.7 Farrer)

```
#include<stdio.h>
int main(){
    int A[20][10], i, j, B[20][11],m,n;
    do {
        printf ("| o numero de linhas de A: ");
        scanf("%d", &m);
    }while (m<1 || m>20);
    do {
        printf ("Informe o numero de colunas de A: ");
        scanf("%d", &n);
    }while (n<1 || n>10);
```

```
for (i=0; i<m; i++){
    B[i][n]=1;
    for(j=0; j<n; j++){
        printf("Informe A[%d][%d] =", i, j);
        scanf("%d", &A[i][j]);
        B[i][j]=A[i][j];
        B[i][n]*=B[i][j];
    }
}
printf ("Matriz A");
for (i=0; i<m; i++){
    printf ("\n");
    for(j=0; j<n; j++){
        printf(" %d",A[i][j]);
    }
}
```

```
printf ("\n Matriz B");
for (i=0; i<m; i++){
    printf ("\n");
    for(j=0; j<=n; j++){
        printf(" %d", B[i][j]);
    }
}
```

Exemplo 6:

Um grupo de pessoas respondeu a um questionário composto de 10 perguntas. Cada pergunta contém cinco opções ou respostas possíveis codificadas de 1 a 5. Cada pergunta é respondida com a escolha de apenas uma opção dentre as cinco opções possíveis.

São fornecidos os nomes das pessoas e suas respectivas respostas. A última informação utilizada, como flag, contém o nome da pessoa igual a "VAZIO".

Fazer um programa para ler e imprimir os dados lidos e calcular e imprimir o número de vezes que foram respondidas cada uma das cinco opções (exercício 2.5.2.9 Farrer)

```
#include<stdio.h>
#include <string.h>
int main(){
    int R[10], i, j, O[5], c=0;
    char N[20];
    for(j=0;j<5; j++)
        O[j]=0;
    printf("Digite o nome da pessoa 1: ");
    scanf ("%s",N);
```

```
while (strcmp(N,"vazio")){
    for(j=0;j<10;j++){
        printf("Digite a resposta da pergunta %d: ", j+1);
        scanf("%d", &R[j]);
        O[R[j]-1]++;
    }
    c++;
    printf ("\nRespostas da pessoa %d ==>", c);
    for(j=0;j<10;j++)
        printf("%d", R[j]);
    printf("\n\nDigite o nome da pessoa numero %d:", c+1);
    scanf ("%s", N);
}
printf ("\nFim dos dados\n");

for(j=0;j<5;j++)
    printf ("\n O numero de pessoas que responderam a opcao %d foi:
%d", j+1, O[j]);
}
```

Exemplo 6 (outra versão)

Um grupo de pessoas respondeu a um questionário composto de 10 perguntas. Cada pergunta contém cinco opções ou respostas possíveis codificadas de 1 a 5. Cada pergunta é respondida com a escolha de apenas uma opção dentre as cinco opções possíveis.

São fornecidos os nomes das pessoas e suas respectivas respostas. A última informação utilizada, como flag, contém o nome da pessoa igual a "VAZIO".

Fazer um programa para ler e imprimir os dados lidos e calcular e imprimir o número de pessoas que responderam a cada uma das cinco opções (exercício 2.5.2.9 Farrer)

```
#include<stdio.h>
#include <string.h>
int main(){
    int R[10], i, j, c=0,O[5],cp=0;
    char N[20];
    printf("Digite o nome da pessoa 1: ");
    scanf ("%s",N);
```

```
while (strcmp(N,"vazio")){
    for(j=0;j<5; j++)
        O[j]=0;
    for(j=0;j<10; j++){
        printf("Digite a resposta da pergunta %d: ", j+1);
        scanf("%d", &R[j]);
        O[R[j]-1]++;
    }
    c++;
    printf ("\nRespostas da pessoa %d ==>", c);
    for(j=0;j<10; j++)
        printf("%d", R[j]);
    if (O[0] && O[1] && O[2]&&O[3]&&O[4])
        cp++;
    printf("\n\nDigite o nome da pessoa numero %d:", c+1);
    scanf ("%s", N);
}
printf ("\nFim dos dados\n");
printf ("\n O numero de pessoas que responderam todas as opcoes
foi: %d", cp);
}
```

Exemplo 7:

Uma matriz triangular inferior preenchida com valores inteiros não negativos é denominada de **mapa triangular**. Um mapa triangular é utilizado em uma certa brincadeira, na qual os participantes caminham ao longo das casas, recolhendo prêmios ao longo do trajeto. O valor do prêmio coletado corresponde ao número no interior de cada posição do mapa. No início do jogo, o participante se encontra na posição $(1, 1)$ e a cada rodada deve se locomover para uma casa na linha imediatamente abaixo da sua posição atual. Entretanto, somente dois movimentos são permitidos. Um deles consiste em o jogador se mover para a posição de coluna igual a atual, na linha de baixo. O outro corresponde a se mover para a posição uma coluna à direita da posição atual na linha de baixo.

Isto é, um participante que se encontre na posição (i, j) pode optar por se mover para a posição $(i + 1, j)$ ou para a posição $(i + 1, j + 1)$. A brincadeira se encerra quando o participante atinge qualquer posição na última linha. Sua tarefa é elaborar um programa que leia um mapa triangular e retorne a quantidade máxima de prêmios que um participante pode coletar. Além disso, o seu programa deve informar o caminho, partindo da posição inicial, que leva à coleta máxima de prêmios.

```
#include <stdio.h>
int main(){
int M[20][20], m,n, i,j, cam_i[20],cam_j[20],premio=0;

/* Valida a dimensão da matriz*/
do{
    printf("Informe o numero de linhas e colunas da matriz (>0 e <=20)");
    scanf ("%d %d", &m, &n);
}while (m>20 || m<=0 || n>20 || n<=0);

/* Informa os elementos da matriz*/

for (i=0; i<m; i++)
    for (j=0; j<n; j++){
        printf ("Informe M[%d][%d]:", i+1,j+1);
        scanf ("%d", &M[i][j]);
    }
```

```
/* Imprime a Matriz */
for (i=0; i<m; i++){
    for (j=0; j<n; j++)
        printf ("%d\t",M[i][j]);
    printf ("\n");
}
/* Determina o caminho de premio máximo */
premio= M[0][0];
cam_i[0]=cam_j[0]=i=j=0;
while (i<m-1){
    i++;
    if (M[i][j]<M[i][j+1])
        j++;
    premio+=M[i][j];
    cam_i[i]=i;
    cam_j[i]=j;
}
```

```
/* Imprime o caminho*/  
    printf ("Caminho :\n");  
    for (i=0; i<m;i++)  
        printf ("%d %d \t",cam_i[i],cam_j[i]);  
        printf ("Valor do premio: %d", premio);  
}
```

Questione: Este programa está correto? Está incompleto?
Se estiver, complete-o!

Exemplo 8:

A matriz abaixo representa o Triângulo de Pascal de ordem 6:

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
```

Os elementos extremos de cada linha são iguais a 1. Os outros elementos são obtidos somando-se os dois elementos que aparecem imediatamente acima e à esquerda na linha anterior. Assim, $10 = 4 + 6$.

Escreva um programa que, dado $n \leq 20$, construa e imprima o Triângulo de Pascal de ordem n , utilizando:

- (a) Uma matriz
- (b) Apenas um vetor

```
#include <stdio.h>
int main(){
    int M[20][20], i,j,n;
        /* Valida a dimensão da matriz*/
    do{
        printf("Informe o numero de linhas e colunas da matriz quadrada
(>0 e <=20)");
        scanf ("%d",&n);
    }while (n>20 || n<=0);

    /*Constroi o triangulo de Pascal de ordem n*/

    for (i=0; i<n; i++){
        M[i][0]=M[i][i]=1;
        for (j=1; j<i; j++)
            M[i][j]=M[i-1][j]+M[i-1][j-1];
    }
```

```
/* Preenche a matriz triangular superior com zeros (exceto a diagonal principal*/  
for (i=0; i<n; i++){  
    for (j=i+1; j<n; j++){  
        M[i][j]=0;  
    }  
    for (j=0; j<n; j++){  
        printf ("%d\t",M[i][j]);  
    }  
    printf ("\n");  
}  
/*Constroi o triangulo de Pascal de ordem n usando apenas um vetor*/  
for (j=0; j<n; j++){  
    v[j]=1;  
    for (i=j-1; i>0; i--){  
        v[i]=v[i-1]+v[i];  
    }  
    for (i=j+1; i<n; i++){  
        v[i]=0;  
    }  
    for (i=0; i<n; i++){  
        printf ("%d\t",v[i]);  
    }  
    printf ("\n");  
}  
}
```