

Arranjos

- 1 dimensão
- várias dimensões



Slide 1

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Arranjo 2 dimensões (Matriz)

Ex: Cinco alunos de uma turma;
Ler e armazenar as 3 notas de cada aluno.
notas

Notas		1	2	3
alunos	1	7.5	8.2	9.1
	2			
	3			
	4			
	5			

Slide 2

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Arranjo 2 dimensões (Matriz)

C - declaração:

tipo nome_arranjo[nro_elem_1ª_dimo] [nro_elem_2ª_dim];

- Número de elementos da dimensão: inteiro ou representar um valor inteiro
- tipo: *int*, *float*, *double* (entre outros).

Ex: `float notas[5][3];` // matriz de 15 elementos reais
`int x[10][25];` // matriz de 250 elementos inteiros

Slide 3

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Arranjo 2 dimensões (Matriz)

C - utilização:

nome_arranjo [índice1ªdim] [índice2ªdim]

Ex:

```
int main () // refere todas as dimensões
{
    float nota [5] [3];
    scanf( "%f", &nota [0] [2]); // 1º aluno, 3ª nota dele

    nota [1] [2] = 7.5 ;
    nota [2] [1] = nota [1] [1] + 2 ;
    if (nota [0] [1] > 6.0)
        printf ("Aprovado");
    ...
}
```

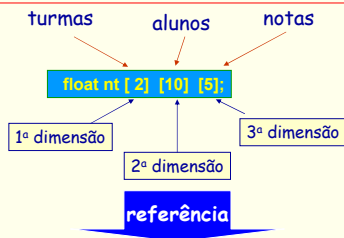
Slide 4

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Arranjo 3 dimensões

Declaração:



Utilização:

`nt [turma] [aluno] [nota]`

- um índice para cada dimensão;
- respeitando a ORDEM da declaração.

Slide 5

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Ex: Suponha que uma rede de lojas seja composta de:

- 10 lojas
- 5 setores por loja
- 30 produtos por setor
- 12 meses

Definir um arranjo para armazenar as quantidades de itens de cada produto vendidos em cada mês, por setor e por loja.

`int quant_vend [10] [5] [30] [12];`

loja setor produto mês

```
...
/*listar quantidade vendida do produto 17
em janeiro, em todos os setores da loja 6 */
for (setor = 0; setor < 5; setor++)
    printf( "%4d",quant_vend [6-1] [setor] [17-1] [1-1] );
...
```

Slide 6

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Arranjo de reais e de strings

EX: Fazer um programa que lê e armazena o nome e as 3 notas de cada um dos NALUNOS alunos de uma turma e calcula e informa a média de cada aluno.

Dica:

- Criar uma matriz de notas com 4 colunas - 3 para as notas e a última para a média do aluno;
- Criar uma matriz para nomes;
- Usar o mesmo índice aluno nas duas matrizes.

```
//Programa que lê nomes e notas, calcula e informa a média de cada aluno
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
#define NRALUNOS 5 // número de alunos da turma
#define NRNOTAS 3 // número de provas
int main()
{
    float nota[NRALUNOS][NRNOTAS + 1]; // armazena média na última coluna
    char nome[NRALUNOS][31]; // aceita nomes de até 30 caracteres
    int a, n; // índices para percorrer as duas dimensões - aluno e nota
    float media_provas[NRNOTAS + 1]; // médias das provas e das médias
    float maior_media;
    system("color f1");
    // leitura de nome e notas do aluno - mesmo índice a:
    for (a = 0; a < NRALUNOS; a++) // para cada aluno
    {
        printf ( "\nDados do aluno %d:\n", a + 1);
        fflush(stdin);
        gets(nome[a]);
        // laço de leitura das notas:
        for (n = 0; n < NRNOTAS; n++) // para cada nota
        {
            printf ( "Nota %d: ", n + 1);
            scanf ( "%f", &nota [ a][ n ] ); // nota do aluno
        }
        printf("\n");
    }
    // continua...
}
```

Slide 7

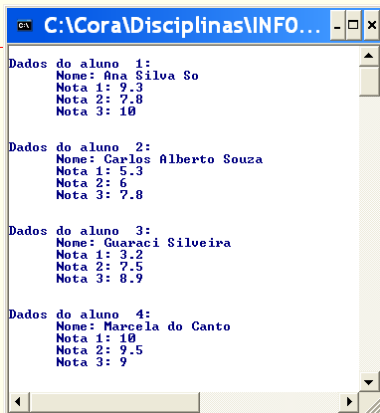
Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

Slide 8

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática



```
Dados do aluno 1:
Nome: Ana Silva So
Nota 1: 9.3
Nota 2: 7.8
Nota 3: 10

Dados do aluno 2:
Nome: Carlos Alberto Souza
Nota 1: 5.3
Nota 2: 6
Nota 3: 7.8

Dados do aluno 3:
Nome: Guaraci Silveira
Nota 1: 3.2
Nota 2: 7.5
Nota 3: 8.9

Dados do aluno 4:
Nome: Marcela do Canto
Nota 1: 10
Nota 2: 9.5
Nota 3: 9
```

Slide 9

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

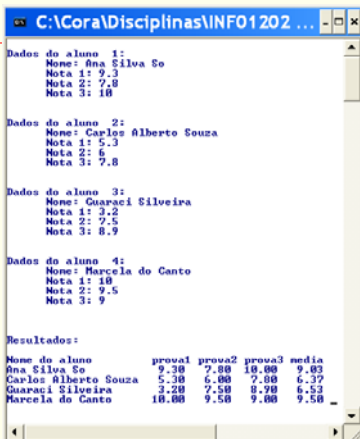
```
...
// calculo da média de cada aluno (usa coluna NRNOTAS):
for (a = 0; a < NRALUNOS; a++) // para cada aluno
{
    nota [ a][ NRNOTAS] = 0;
    for (n = 0; n < NRNOTAS; n++) // soma notas obtidas
        nota [ a][ NRNOTAS] = nota [ a][ NRNOTAS] + nota [ a][ n];
    // calcula média:
    nota [ a][ NRNOTAS] = nota [ a][ NRNOTAS] / NRNOTAS; }

//Imprime resultados:
printf("\n\nResultados:\n");
printf("\nNome do aluno          prova1 prova2 prova3 media");
for (a = 0; a < NRALUNOS; a++) // para cada aluno
{
    printf ( "\n%-20s", nome[a]);
    // laço de impressão das notas:
    for (n = 0; n < NRNOTAS; n++) // para cada nota
        printf ( "%6.2f ", nota [ a][ n ] ); // nota do aluno
    }
    system("pause>>null");
    return;
}
```

Slide 10

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática



```
Dados do aluno 1:
Nome: Ana Silva So
Nota 1: 9.3
Nota 2: 7.8
Nota 3: 10

Dados do aluno 2:
Nome: Carlos Alberto Souza
Nota 1: 5.3
Nota 2: 6
Nota 3: 7.8

Dados do aluno 3:
Nome: Guaraci Silveira
Nota 1: 3.2
Nota 2: 7.5
Nota 3: 8.9

Dados do aluno 4:
Nome: Marcela do Canto
Nota 1: 10
Nota 2: 9.5
Nota 3: 9

Resultados:
Nome do aluno          prova1 prova2 prova3 media
Ana Silva So           9.30   7.80   10.00   9.03
Carlos Alberto Souza   5.30   6.00   7.80   6.37
Guaraci Silveira       3.20   7.50   8.90   6.52
Marcela do Canto       10.00   9.50   9.00   9.50
```

Slide 11

UFRGS Informática

Matrizes – operações mais comuns

Dada uma matriz inteira **matriz** (linha, coluna):

- a) preenche-la por leitura, imprimindo a seguir a matriz lida;
- b) o maior elemento de cada coluna da matriz;
- c) a média dos elementos de cada linha;
- d) o produto de todos os elementos diferentes de zero;
- e) o número de elementos negativos;
- f) posição ocupada (linha-coluna) por um elemento cujo valor será lido pelo programa.

Considere as declarações:

```
int matriz[TLINHA][TCOLUNA]; // matriz e limites
int linha, coluna; // índices
int maior, somalinha, cont_neg;
float media;
long produto;
```

Slide 12

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

a) leitura e impressão da matriz:

```
// Manipulação de matrizes:
#include <stdio.h>
#include <stdlib.h>
#define TLINHA 3
#define TCOLUNA 4
int main()
{
    int matriz[TLINHA][TCOLUNA]; // matriz e limites
    int linha, coluna; // índices
    int maior, somalinha, cont_neg;
    int valor, achou;
    float media;
    long produto;

    // lendo a matriz
    for (linha = 0; linha < TLINHA; linha++)
        for (coluna = 0; coluna < TCOLUNA; coluna++)
        {
            printf("Informe elemento[%d,%d]: ", linha+1, coluna+1);
            scanf("%d", &matriz[linha][coluna]);
        }

    // imprimindo a matriz
    for (linha = 0; linha < TLINHA; linha++)
    {
        for (coluna = 0; coluna < TCOLUNA; coluna++)
            printf("%4d", matriz[linha][coluna]);
        printf("\n");
    }
    .....
}
```

```
Informe elemento[1,1]: 5
Informe elemento[1,2]: -3
Informe elemento[1,3]: 0
Informe elemento[1,4]: 12
Informe elemento[2,1]: -32
Informe elemento[2,2]: 44
Informe elemento[2,3]: 9
Informe elemento[2,4]: 0
Informe elemento[3,1]: -1
Informe elemento[3,2]: 1
Informe elemento[3,3]: 22
Informe elemento[3,4]: 15
5 -3 0 12
-32 44 9 0
-1 1 22 15
```

b) identificação do maior elemento de cada coluna:

```
// percorre coluna à coluna:
for (coluna = 0; coluna < TCOLUNA; coluna++)
{
    // para cada coluna, percorre todas as linhas:
    maior = matriz[0][coluna];
    for (linha = 1; linha < TLINHA; linha++)
        if (matriz[linha][coluna] > maior)
            maior = matriz[linha][coluna];
    printf("\nMaior da coluna %d = %d", coluna, maior);
}
```

```
5 -3 0 12
-32 44 9 0
-1 1 22 15
Maior da coluna 0 = 5
Maior da coluna 1 = 44
Maior da coluna 2 = 22
Maior da coluna 3 = 15
```

c) a média dos elementos de cada linha:

```
for (linha = 0; linha < TLINHA; linha++)
{
    // fixa linha e varia as colunas:
    somalinha = 0;
    for (coluna = 0; coluna < TCOLUNA; coluna++)
        somalinha = somalinha + matriz[linha][coluna];
    media = (float) somalinha / TCOLUNA; *
    printf("\nMedia linha %d = %.2f", linha, media);
}
```

* usa *cast* para que media seja considerada *float*, já que somalinha e tcoluna são inteiros (para que o resultado da divisão também seja *float*).

d) produto de todos os elementos diferentes de zero:

```
produto = 1; // inicializa com 1 !!!!
// percorre toda a matriz, aqui linha à linha (tanto faz)
for (linha = 0; linha < TLINHA; linha++)
    for (coluna = 0; coluna < TCOLUNA; coluna++)
        if (matriz[linha][coluna] != 0) // != 0 é false
            produto = produto * matriz[linha][coluna];
printf("Produto = %d", produto);
```

e) conta elementos negativos:

```
cont_neg = 0; // contador inicia em zero !!!
for (linha = 0; linha < TLINHA; linha++)
    for (coluna = 0; coluna < TCOLUNA; coluna++)
        if (matriz[linha][coluna] < 0)
            cont_neg++;
printf("Numero de negativos = %d", cont_neg);
```

d) localiza posição ocupada (linha-coluna) por um valor lido:

```
// percorrendo toda a matriz - for:
printf("\nInforme valor a ser pesquisado:");
scanf("%d", &valor);
for (linha = 0; linha < TLINHA; linha++)
    for (coluna = 0; coluna < TCOLUNA; coluna++)
        if (matriz[linha][coluna] == valor)
            // impressão das posições iniciando em 1:
            printf("Valor na posicao [ %d , %d ]", linha + 1, coluna + 1);
```

Se encontra logo?

Se não existe?

d) localiza posição ocupada (linha-coluna) por um valor lido:

```
// parando ao encontrar ou informando que não existe:
printf("\nInforme valor a ser pesquisado:");
scanf ("%d", &valor);
linha = 0;
achou = 1;
do
{
    coluna= 0;
    do
    {
        if (m [linha] [coluna]== valor)
        {
            achou=0;
            printf ("Valor na linha: %d e coluna: %d", linha + 1, coluna + 1);
        }
        coluna++;
    } while (coluna < TCOLUNA && achou==1);
    linha++;
} while (linha < TLINHA && achou == 1); /*para qdo acha ou qdo percorreu
                                     toda a matriz */
if (achou == 1)
    printf ("Valor não existe na matriz.");
```

Slide 19

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

```
Informe elemento1,1: 5
Informe elemento1,2: -3
Informe elemento1,3: 0
Informe elemento1,4: 12
Informe elemento2,1: -32
Informe elemento2,2: 44
Informe elemento2,3: 9
Informe elemento2,4: 0
Informe elemento3,1: -1
Informe elemento3,2: 1
Informe elemento3,3: 22
Informe elemento3,4: 15
5 -3 0 12
-32 44 9 0
-1 1 22 15
Maior da coluna 0 = 5
Maior da coluna 1 = 44
Maior da coluna 2 = 22
Maior da coluna 3 = 15
Media linha 0 = 3.50
Media linha 1 = 5.25
Media linha 2 = 9.25
Produto = -752716800
Numero de negativos = 3
Informe valor a ser pesquisado:0
Valor na linha: 1 e coluna: 3
```

Slide 20

UFRGS Informática

Exercício para a próxima aula: Um professor quer armazenar os dados referentes ao gabarito de provas aplicadas a seus alunos em uma matriz. O professor possui 2 turmas, sendo que para cada turma serão aplicadas 3 provas, de 5 questões de escolha simples (respostas a,b,c,d,e) cada uma. Para cada prova, são armazenados os resultados de cada questão correspondente ao gabarito desta prova. Pede-se:

- declarar um **arranjo tridimensional** para armazenar as respostas de todas as provas de cada turma, isto é, os gabaritos.
- preencher** por leitura este arranjo;
- imprimir**, para cada turma a matriz de respostas de cada prova (colocando na tela os cabeçalhos explicativos dos conteúdos apresentados).
- imprimir** a(s) resposta(s) mais comum(ns) para as provas da segunda turma (a,b,c,d ou e).

Saídas:

- GABARITOS formatados (qual turma, qual prova)
 - Resposta com maior incidência (a,b,c,d ou e)
- Vetor contador de resposta** Como declarar?

Slide 21

Profa. Cora H.F. Pinto Ribeiro

UFRGS Informática

```
//Programa gabarito
#include <stdlib.h>
#include <stdio.h>
#include <ctype.h>
#define TTURMA 1
#define TPROVA 2
#define TQUESTAO 5
int main( )
{
    char gabarito[TTURMA][TPROVA][TQUESTAO]; // armazena gabaritos
    int i,p,q; // indices para percorrer a matriz - turma prova questão
    char resp; // resposta
    char letra[5] = {'a','b','c','d','e'}; // para impressão da opção com mais uso
    int cont [5]={0}; // vetor contador de cada resposta
    int maior; // respostas com opção mais encontrada
    system("color f1");
```

```
Turma 1:
Prova 1:
  Questao 1: a
  Questao 2: B
  Questao 3: c
  Questao 4: b
  Questao 5: A

Prova 2:
  Questao 1: _
```

Slide 22

UFRGS Informática