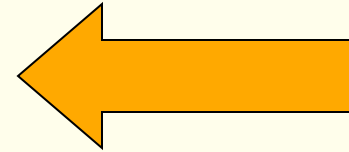


Arranjos

- *1 dimensão*
- *várias dimensões*



Ex: Ler e armazenar as 5 notas de um aluno.

```
//Notas de um aluno
#include <stdio.h>
#define LIMITE 5
int main ( )
{
    float notas[LIMITE]; // arranjo para as notas
    int indice;          //variável de controle da repeticao
    ...
    //obtem as 5 notas do aluno
    for (indice = 0; indice < LIMITE ; indice++)
        scanf ("%f", &notas [indice]);
    ...
    return 0;
}
```

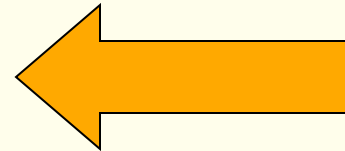
	0	1	2	3	4
Notas	7.5	8.2	9.1	6.5	8.7

Ex: Dez alunos de uma turma
Ler e armazenar as 5 notas de cada aluno.



Arranjos

- *1 dimensão*
- *várias dimensões*



Arranjo 2 dimensões (Matriz)

4

Ex: Dez alunos de uma turma
Ler e armazenar as 5 notas de cada aluno.

alunos →

```
float notas [10] [5];
```

Notas

0 1 2 3 4 5 6 7 8 9

notas ↓

0

1

2

3

4

nota [7] [3]

8.5

Arranjo 2 dimensões

Linguagem C

Declaração

< tipo >

<identificador> [numero elementos 1a dimensão] [num elementos 2a dimensão]

Ex:

```
float notas [5] [10];  
int    x [10] [25];
```

Arranjo 2 dimensões

Linguagem C

Utilização

< nome do arranjo > [< índice1^adim>] [<índice2^adim >]

Ex:

```
int main ( )
{
    float nota [5] [10];
    scanf( "%f", &nota [2] [3]);

    nota [1] [2] = 7.5 ;

    nota [2] [1] = nota [1] [1] + 2 ;

    if (Nota [0] [1] > 6.0)
        printf ("Aprovado");

    ...
}
```

Arranjo 2 dimensões Linguagem C

Inicialização automática de um arranjo na
declaração: vetores de vetores

Exemplos:

```
int x[2][3] = { {1,2,-1} , {2,0,3} };
```

```
char y[2][2] = { { 'a' , 'h' } , { 'o' , '9' } };
```

```
char nomes[2][20]={ "Jose" , "Maria"};
```

Ex: Ler e armazenar as 5 notas de cada um dos alunos de uma turma (10 alunos).

Calcular e informar a média da 1ª nota

Quantos alunos tem a 1ª nota superior a esta média ?

//Processa notas de uma turma

#include <stdio.h> #define NALUNOS 10 #define NNOTAS 5

int main ()

{

float notas [NALUNOS] [NNOTAS] ; // matriz para as notas

int a, n; // indices para as 2 dimensoes

int cont; // contador de alunos

float soma, media; // soma e media de notas

for (a = 0; a< NALUNOS; a++) // para cada aluno

for (n = 0; n < NNOTAS; n++) // obtem as 5 notas do aluno

scanf ("%f", ¬as[a] [n]);

// Calculo da media da primeira nota

soma = 0;

for (a = 0; a< NALUNOS ; a++)

soma = soma + notas [a] [0];

media = soma / NALUNOS;

printf("Media da primeira nota: %6.2f", media);

// Conta quantos alunos tem 1a nota > media da primeira nota

cont = 0;

for (a = 0; a < NALUNOS; a++)

if (notas [a] [0] > media)

cont++;

printf("Numero de alunos ...: %d", cont)

return 0;

}

Ex: Matriz inteira **m** (10, 3), já preenchida.
Imprimir em forma de matriz!

```
...  
for (linha = 0; linha < 10; linha++)  
{  
    for (coluna = 0; coluna < 3; coluna++)  
        printf( "%10d", m[linha] [coluna]);  
    printf("\n\n");  
}  
...
```

Imprimir terceira coluna

```
...  
for (linha = 0; linha < 10; linha++)  
    printf( "%10d", m[linha] [2]);  
printf("\n\n");  
...
```

Somar 2 em cada elemento da linha 7

```
...  
for (coluna = 0; coluna < 3; coluna++)  
    m [7] [coluna] = m [7] [coluna] + 2;  
...
```

m

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

Arranjo 3 dimensões - Linguagem C

Declaração

< tipo > <identificador> [<num. elem. da 1ª dim. >] [< num. elem. da 2ª dim. >] [< num. elem. da 3ª dim. >]

Ex:

```
float nt [2] [10] [5];
```

1ª dimensão

2ª dimensão

3ª dimensão

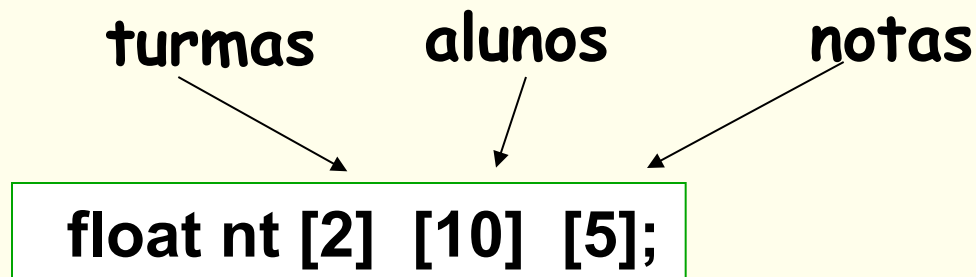
Arranjo 3 dimensões - Linguagem C

Utilização

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

Ex:

turmas alunos notas



```
float nt [2] [10] [5];
```

referência



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

Ex: 10 lojas
5 setores por loja
30 produtos por setor
12 meses

O arranjo armazena quantidade de itens de cada produto vendido em cada mês, por setor e por loja.

```
int quant_vend [11] [6] [31] [13];
```

loja

setor

produto

mês

```
#define LOJA 11
#define SETOR 6
#define PRODUTO 31
# define MÊS 13
/*listar quantidade vendida do produto 17
 em janeiro, em todos os setores da loja 6*/
for (set = 1; set<SETOR; set++)
    printf( "%4d",quant_vend [6] [set] [17] [1] );
...
```


Arranjo de “n” dimensões

Utilização

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

Ex:

notas alunos turmas escolas



```
float nt [ 5] [10] [3] [2 ];
```

referência



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

Arranjo Bidimensional – Operações mais comuns

Dada uma matriz inteira m (TLINHA, TCOLUNA), preenchê-la por leitura e imprimir:

- a) o maior elemento de cada linha da matriz;
- b) a média dos elementos de cada coluna;
- c) o produto de todos os elementos diferentes de zero;
- d) quantos elementos são negativos;
- e) posição ocupada (linha-coluna) por um elemento cujo valor será lido pelo programa.

a) o maior elemento de cada coluna da matriz:

```
for (coluna = 0; coluna < TCOLUNA; coluna++)  
{  
    maior = matriz[0] [coluna];  
    for (linha = 1; linha < TLINHA; linha++)  
        if (maior < matriz [linha] [coluna])  
            maior = matriz [linha] [coluna];  
    printf ("Maior da coluna %d = %d", coluna,  
maior);  
}
```


b) a média dos elementos de cada linha:

17

```
for (linha = 0 ; linha < TLINHA ; linha++)
{
    somacoluna = 0;
    for (coluna = 0; coluna < TCOLUNA; coluna++)
        somacoluna += matriz [linha] [coluna];
    media = (float) somacoluna / TCOLUNA; *
    printf("Media linha  %d = %f", linha, media);
}
```

* somacoluna e tcoluna sendo inteiros, para obter-se media como float fez-se um *casting* e somacoluna foi considerada *float* na expressão, como consequência o resultado da divisão também resultou *float*.

c) produto de todos os elementos diferentes de zero:

```
produto = 1;
for (linha = 0; linha < TLINHA ; linha++)
    for (coluna = 0; coluna < TCOLUNA; coluna++)
        if (matriz [linha] [coluna])
            produto = produto * matriz[linha][coluna] ;
printf ("Produto = %d", produto );
```

d) quantos elementos são negativos:

```
negativos = 0;
for (linha = 0; linha < TLINHA; linha++)
    for (coluna = 0; coluna < TCOLUNA; coluna++)
        if (matriz [linha] [coluna] < 0)
            negativos++;
printf ("Numero de negativos = %d", negativos);
```

e) posição ocupada (linha-coluna) por um elemento cujo valor será lido:

```
scanf ("%d", &valor);  
for (linha = 0; linha < TLINHA; linha++)  
    for (coluna = 0; coluna < TCOLUNA; coluna++)  
        if (matriz [linha] [coluna] == valor)  
            printf  
            ("Valor na linha: %d e coluna: %d", linha,  
            coluna);
```

Exercício: Leia uma matriz quadrada (de inteiros) de ordem 10 e calcule a sua transposta. Imprima as duas matrizes.

```
/* Le uma matriz inteira 10 x 10 e calcula sua transposta
```

```
Entradas: matriz 10 x 10
```

```
Saida: matriz 10 x 10 (transposta da entrada) */
```

```
# define TAMMAT 10
```

```
int main()
```

```
{
```

```
    int m[TAMMAT][TAMMAT], t[TAMMAT][TAMMAT];
```

```
    int linha, coluna;
```

```
    // Leitura matriz
```

```
    for (linha = 0; linha < TAMMAT; linha++){
```

```
        printf("Entre os elementos da linha %d:\n", linha+1);
```

```
        for (coluna = 0; coluna < TAMMAT; coluna++){
```

```
            scanf("%d", &m[linha][coluna]);
```

```
        }
```

```
    // gera a transposta
```

```
    for (linha = 0; linha < TAMMAT; linha++){
```

```
        for (coluna = 0; coluna < TAMMAT; coluna++){
```

```
            t[linha][coluna] = m[coluna][linha];
```

```
    // Imprime matriz original
```

```
    // Imprime matriz transposta
```

```
    getchar();
```

```
    return 0;
```

```
}
```

Ex:

- Armazenar o número e as 3 notas de cada aluno de 1 turma.
- Calcular média de cada aluno - armazenar, e também a média da turma em cada prova.
- Informar o número do aluno de maior nota em cada prova.

//Programa notas e medias**#include <stdio.h>****#define NRALUNOS 30****#define NRNOTAS 3****int main ()****{****float nota[NRALUNOS] [NRNOTAS];****int n , a; // indices para percorrer as duas dimensoes****int numero[NRALUNOS]; // numero dos alunos****float media [NRALUNOS] ; // media de cada aluno****float media_provas[NRNOTAS]; // medias das provas****float pos_maior_media;****...**

//Programa notas e medias

```
#include <stdio.h>
#define NRALUNOS 3
#define NRNOTAS 3
int main ( )
{
    float nota[NRALUNOS] [NRNOTAS];
    int n , a; // indices para percorrer as duas dimensoes
    int numero[NRALUNOS]; // numero dos alunos
    float media [NRALUNOS] ; // media de cada aluno
    float media_provas[NRNOTAS]; // medias das provas
    int pos_maior_nota;

    ...
    for (a = 0; a< NRALUNOS; a++) //calculo da media de cada aluno
    {
        media [a] = 0;
        for (n = 0; n< NRNOTAS ; n++)
            media [a] = media [a] + nota [a] [n];
        media [a] = media [a] / NRNOTAS;
    }
    for (n= 0; n< NRNOTAS ; n++) // calculo da media da turma / prova
    {
        media_provas [ n ] = 0;
        for (a = 0; a < NRALUNOS; a++)
            media_provas [n] = media_provas[n] + nota[ a] [n];
        media_provas [n] = media_provas [n] / NRALUNOS;
    }
    ...
}
```

//Programa notas e medias

```
#include <stdio.h>
#define NRALUNOS 3
#define NRNOTAS 3
int main ( )
{
    float nota[NRALUNOS] [NRNOTAS];
    int n , a; // indices para percorrer as duas dimensoes
    int numero[NRALUNOS]; // numero dos alunos
    float media [NRALUNOS] ; // media de cada aluno
    float media_provas[NRNOTAS]; // medias das provas
    int pos_maior_nota;
    ...
    printf("\nNumero do aluno com a maior nota em cada prova:\n");
    for (n = 0; n < NRNOTAS; n++)
    {
        pos_maior_nota = 0;
        for (a = 1; a < NRALUNOS ; a++)
            if (nota [ a ] [n] > nota [ pos_maior_nota] [n])
            {
                printf("\n%d %d\n", pos_maior_nota, a);
                pos_maior_nota = a;
            }
        printf( "\nProva %d : aluno: %d: " , n+1 , numero[pos_maior_nota]);
    }
    printf ( "\nFinal de programa !\n");
    return 0;
}
```