


Arranjos

- *1 dimensão* 
- *várias dimensões*

Ex: Ler as notas de 30 alunos.
Calcular e informar a média da turma.

Algoritmo MédiaTurma

{ Obtém as notas de 30 alunos e informa a média }

Entradas: 30 notas

Saída: média das 30 notas lidas

Processamento

- 1. Inicializar somatório das notas**
- 2. Repetir 30 vezes (para cada aluno)**
 - 2.1 obter a nota de um aluno**
 - 2.2 acrescentar esta nota ao somatório de notas**
- 3. Calcular a média**
- 4. Informar a média**
- 5. Terminar**

Ex: Ler as notas de 30 alunos.
Calcular e informar a média da turma.

```
//calcula media de uma turma
#include <stdio.h>
int main ( )
{
    float nota; // nota de um aluno
    float media, soma;
    int aluno; //variável de controle da repeticao
    soma = 0; // inicializa a soma das notas
    for (aluno = 1; aluno <= 30; aluno++) // para cada aluno da
        turma
    {
        scanf("%f", &nota); //obtem a nota do aluno
        soma = soma + nota;
    }
    media = soma / 30;
    printf( 'Media da turma: %7.2f", media);
    return 0;
}
```

Ex: Ler as notas de 30 alunos. Calcular e informar a média da turma. Informar, ainda, as notas que são superiores à média calculada.

Algoritmo MédianotasSup

{ Obtém as notas de 30 alunos.

Informa a média e as notas superiores à media }

Entradas: 30 notas

Saídas: média das 30 notas lidas
notas superiores à média

Processamento

1. Inicializar somatório das notas
2. Repetir 30 vezes (uma vez para cada aluno)
 - 2.1 obter a nota de um aluno
 - 2.2 acrescentar esta nota ao somatório de notas
3. Calcular e informar a média
4. Informar as notas superiores à média
5. Terminar



Ex: Ler as notas de 30 alunos. Calcular e informar a média da turma. Informar, ainda, as notas que são superiores à média calculada.

//calcula media de uma turma e notas superiores a media

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

```
int main ( )
```

```
{
```

```
float n1, n2, n3, n4, n5, { ...}, n30;// uma variavel para cada nota
```

```
float media, soma;
```

```
// obtem as 30 notas
```

```
scanf("%f %f (...) %f", &n1, &n2, ( ... ) &n30);
```

```
soma = n1+n2+n3+n4+n5+n6+ (...) n30;
```

```
media = soma / 30;
```

```
printf("Media da turma: %7.2f", media);
```

```
if (n1 > media) // testa cada nota !!!
```

```
printf("%f", n1);
```

```
if (n2 > media)
```

```
printf("%f", n2);
```

```
{ ... }
```

```
return 0;
```

É a única solução ?

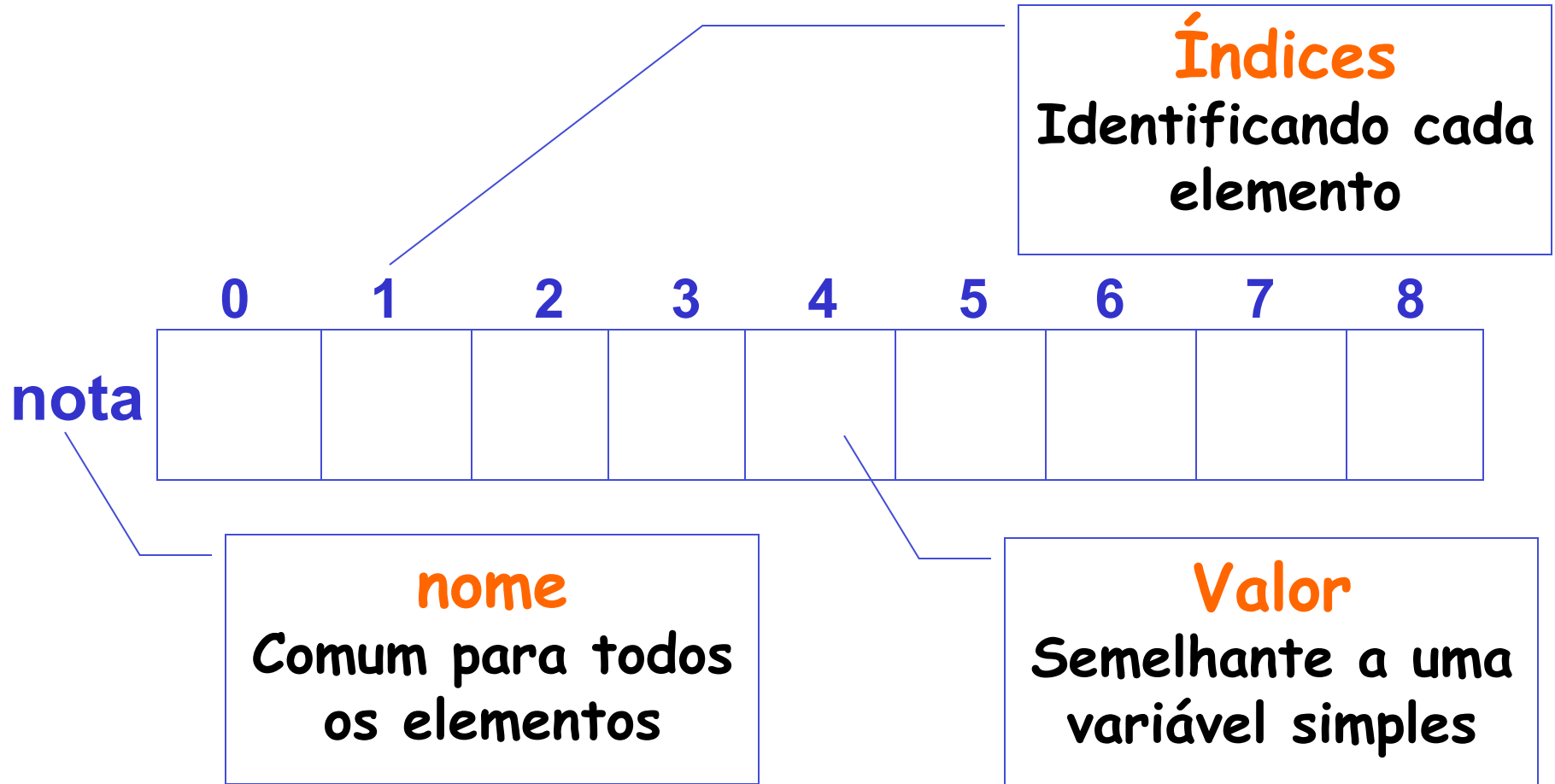
Tipos de dados na Linguagem C

- **básicos**
 - **definidos pelo usuário**
- 
- ```
graph LR; A["• básicos
• definidos pelo usuário"] --> B["• char
• int
• float
• double
• void"]; A --> C["• estrutura
• campo de bit
• união
• enumeração
• typedef"]
```

- char
- int
- float
- double
- void

- estrutura
- campo de bit
- união
- enumeração
- typedef

# Arranjo ( Vetor )



# Arranjo

- conjunto ordenado de informações de mesma natureza;
- elementos todos do mesmo tipo;
- acesso randômico.

|      |   |   |     |   |   |   |   |   |   |
|------|---|---|-----|---|---|---|---|---|---|
|      | 0 | 1 | 2   | 3 | 4 | 5 | 6 | 7 | 8 |
| nota |   |   | 8,5 |   |   |   |   |   |   |

nota [2] vale 8,5



# Arranjo - Linguagem C

## Declaração

### Tipo arranjo



`<tipo> <nome do arranjo> [ < número de elementos da dimensão > ]`

- número de elementos da dimensão deve ser um inteiro ou representar um valor inteiro;
- tipo pode ser, entre outros, int, float, double.

**Ex:**

```
float nota[30];
int x[15];
```

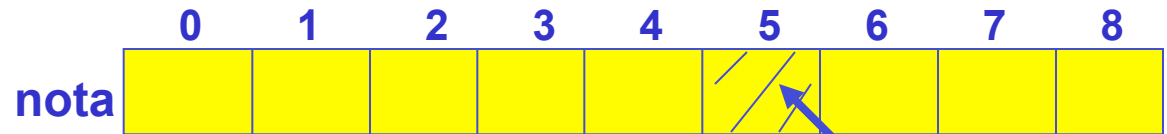
**ATENÇÃO:** em C, strings (= cadeias de caracteres) são arranjos de caracteres, que serão vistos em detalhes mais adiante.

# Arranjo - Linguagem C

## Utilização

Variável indexada

< nome do arranjo > [ < índice > ]



Ex:

```
float nota[9];
scanf("%f", ¬a[5]);

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

nota[ 5 ]

# Arranjo - Linguagem C

< nome do arranjo > [ < índice > ]

## Índice

- constante, nome de variável, ou expressão que seja um valor inteiro (ou resulte em) um valor inteiro;
- sempre inicia em 0.

## Variável indexada



**Ex:**

```
int main ()
{
 float nota [9];
 int indice;
 indice = 5;
 scanf ("%f", ¬a[indice]);
 nota[indice + 1] = 7.5 ;
 ...
}
```

**Ex:** nota [ 6 ]

|      |   |   |   |   |   |   |   |   |   |
|------|---|---|---|---|---|---|---|---|---|
|      | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| nota |   |   |   |   |   |   |   |   |   |

# Arranjo - Linguagem C

## Inicialização

Arranjos, como as demais variáveis, ao serem criados contêm lixo.

Arranjos podem ser inicializados:

- automaticamente ao serem criados;
- por atribuição, em algum momento da execução;
- por leitura.

# Inicialização automática de um arranjo na declaração:

**Forma geral:**

**tipo var[n] = {valor<sub>0</sub>, valor<sub>1</sub>, valor<sub>n-1</sub>};**

**int v[3] = {10, 20, 30};**

|          |           |           |           |
|----------|-----------|-----------|-----------|
|          | <b>0</b>  | <b>1</b>  | <b>2</b>  |
| <b>v</b> | <b>10</b> | <b>20</b> | <b>30</b> |

**Posições não inicializadas são preenchidas com zero:**

**int x[5] = {10, 20, 30};**

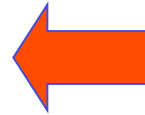
|          |           |           |           |          |          |
|----------|-----------|-----------|-----------|----------|----------|
|          | <b>0</b>  | <b>1</b>  | <b>2</b>  | <b>3</b> | <b>4</b> |
| <b>x</b> | <b>10</b> | <b>20</b> | <b>30</b> | <b>0</b> | <b>0</b> |

# Inicialização automática de um arranjo na declaração (cont.):

```
float z[5] = {2.7, 3.5, 4.2, 9.23 , 4.7};
```

o mesmo que escrever:

```
float z[5];
z[0] = 2.7;
z[1] = 3.5;
z[2] = 4.2;
z[3] = 9.23;
z[4] = 4.7;
```

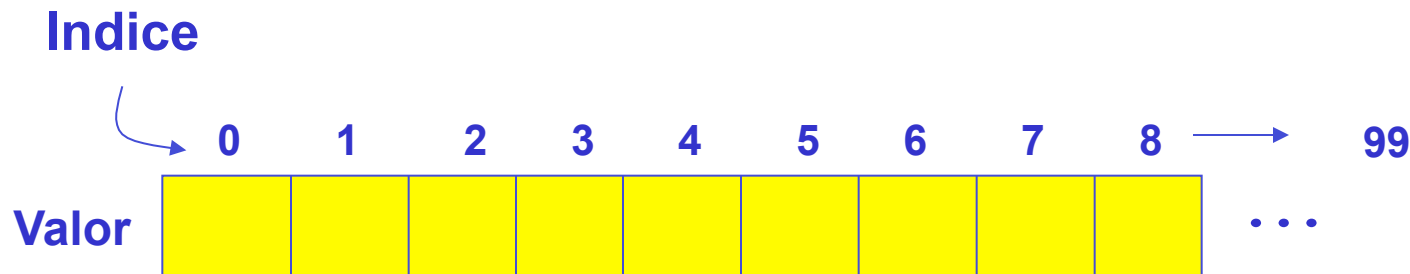


**Inicialização por  
atribuição, durante  
a execução.**

**Ex:** Preencher por leitura um arranjo de 100 elementos inteiros.

```
int valor[100];
int indice;
for (indice = 0; indice < 100; indice++)
 scanf ("%d", &valor[indice]);
...
```

**ATENÇÃO:**  
Como os índices de um arranjo sempre iniciam em 0, e o arranjo **valor** tem 100 elementos, os índices para acessá-lo vão de 0 a 99!



**Ex:** Somar os elementos de um arranjo  $x$  ( 200 )

```
...
soma = 0;
for (indice = 0; indice < 200; indice++)
 soma = soma + x[indice] ;
...
```

**Ex:** Ler as notas de 30 alunos. Calcular e informar a média da turma. Informar, ainda, as notas que são superiores à média calculada.



**Ex:** Ler as notas de 30 alunos. Calcular e informar a média da turma. Informar, ainda, as notas que são superiores à média calculada.

**//Calcula media de uma turma e informa notas superiores a media**

**#include <stdio.h>**

**int main ( )**

**{**

**float nota [30] ; //arranjo para as notas**

**float media, soma;**

**int aluno; // variavel de controle da repeticao**

**soma = 0; // inicializa soma**

**for (aluno = 0; aluno < 30 ; aluno++) //para cada aluno**

**{**

**scanf ("%f", &nota[aluno]); // obtem sua nota**

**soma = soma + nota[aluno]; // acumula a soma**

**}**

**media = soma / 30;**

**printf("Media da turma: %7.2f", media);**

**for (aluno = 0; aluno < 30; aluno++) // mostra notas > media**

**if (nota[aluno] > media)**

**printf ("%7.2f", nota[aluno]);**

**return 0;**

**}**

**Ex:** Os preços de mercadorias de uma loja são armazenados em um arranjo `Preco`. Os índices do arranjo correspondem aos códigos das mercadorias (10 a 99). Informe:

- o código da mercadoria cujo preço é R\$ 123,00 (supor todos os preços diferentes)

```
//codigo da mercadoria com preco determinado
#include <stdio.h>
int main ()
{
 int cod;
 float preco [100];
 // para usar indice 99, tem que declarar
 //vetor com 100 elementos
//obter preços
for (cod = 10; cod < 100; cod++)
 scanf("%f", &preco[cod]);
//procurar produto de preço R$ 123,00
for (cod = 10; cod <100; cod++)
 if (preco[cod] == 123)
 printf("%d ", cod);
return 0;
}
```

### Problemas desta solução:

- mesmo depois de encontrar, continua percorrendo o arranjo;
- não informa se não encontrou o valor procurado.

## Outra solução:

```
//procura codigo de mercadoria
//com preço determinado
#include <stdio.h>
int main ()
{
 float preco[100];
 int cod;
 int achou; // 0 = falso, 1 = verdadeiro
 //obter preços
 for (cod = 10; cod<=99;cod++)
 scanf("%f", &preco[cod]);
 // procurar produto de preço R$ 123,00
 achou = 0;
 cod = 9;
 do
 {
 cod = cod + 1;
 if (preco[cod] == 123)
 achou = 1;
 }
 while (cod < 99 && achou == 0);
 if (achou == 1)
 printf ("%d", cod);
 else
 printf("Não encontrou!");
 return 0;
}
```

**Ex:** Ler os números de alunos - arranjo numero(30).  
Ler nota de cada aluno - arranjo nota(30).  
Mesmo índice relaciona numero com sua nota.  
Informar:  
- os números dos alunos com nota menor que 6,0.

```
//alunos com nota menor que 6,0
#include <stdio.h>
int main ()
{
 int numero [30];
 float nota [30];
 int ind;
 for (ind = 0; ind <30; ind++)
 {
 scanf("%d", &numero[ind]);
 scanf("%f", ¬a[ind]);
 if (nota[ind] < 6.0)
 printf("%d", numero[ind]);
 }
 ...
 return 0;
}
```

## Exercício difícil:

1. Preencher um arranjo  $X$  ( 200 ) por leitura.
2. Calcular a média dos 10 maiores elementos deste arranjo.
3. Informar o índice do elemento mais próximo da média.