

Curso de C

Vetores

Vetores

Objetivos:

- Aprender a guardar mais de um valor de mesmo tipo usando um mesmo nome de variável
- Trabalhar com listas, vetores e matrizes

Vetores

Roteiro:

- Declaração de vetores
- Uso correto de vetores
- Vetores com tamanho variável
- Matrizes
- Vetores de caracteres (texto)

Introdução

Tipos de Dados:

- Da linguagem C:
 - Números inteiros: `int`, `long int`, `unsigned int`, ...
 - Números fracionários: `float`, `double`, ...
 - Caracteres: `char`
- Do programador:
 - Vetores, estruturas, enumerações, etc
 - Sinônimos

Vetores

Declaração e Uso

Vetores

Conceitos:

- Seqüência de valores
- Todos do mesmo tipo
- Nome único para a variável
- Acesso por índice
- Tamanho fixo
- Numeração de 0 até *tamanho-1*

int v[10]

v₀ (int)

v₁ (int)

v₂ (int)

v₃ (int)

v₄ (int)

v₅ (int)

v₆ (int)

v₇ (int)

v₈ (int)

v₉ (int)

Declaração

Declaração de vetor:

tipo *variavel* [tamanho]

Qualquer tipo
da linguagem C

Nome da
variável

Número de
elementos

Exemplo:

```
int vetor[10];
```

```
double medidas[100];
```

int v[10]

v₀ (int)

v₁ (int)

v₂ (int)

v₃ (int)

v₄ (int)

v₅ (int)

v₆ (int)

v₇ (int)

v₈ (int)

v₉ (int)

Acesso

Elementos com índice:

Elementos do vetor:

`vetor[0], vetor[1], vetor[2], ...`

Atribuição:

`vetor[indice] = valor;`

Exemplo:

```
int vetor[10];
```

```
vetor[5] = 3;
```

```
vetor[0] = vetor[1] + vetor[2];
```

`int v[10]`

`v0 (int)`

`v1 (int)`

`v2 (int)`

`v3 (int)`

`v4 (int)`

`v5 (int)`

`v6 (int)`

`v7 (int)`

`v8 (int)`

`v9 (int)`

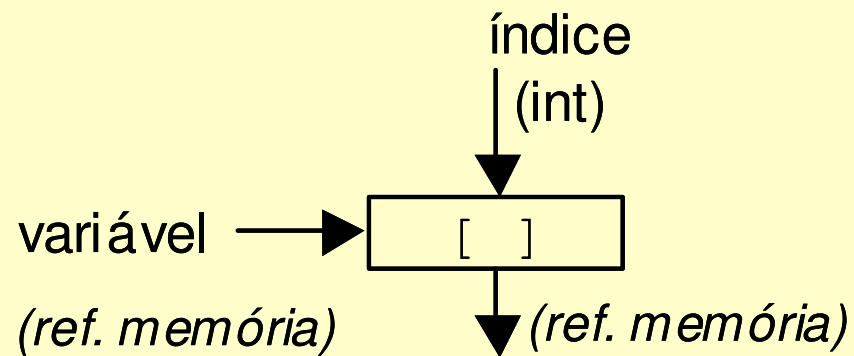
Acesso

Exemplo:

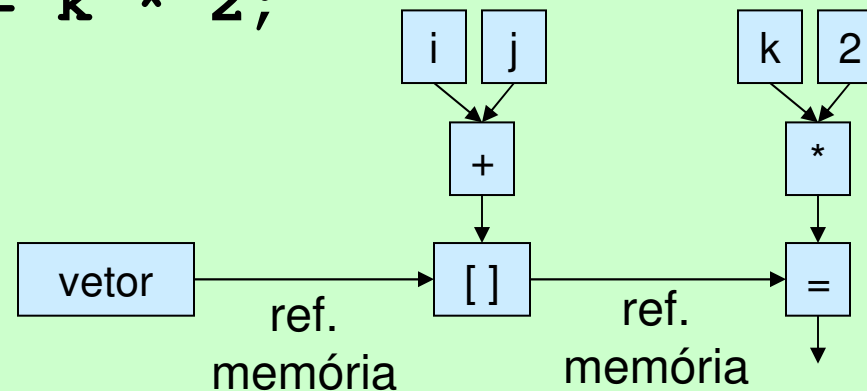
```
int vetor[100];  
...  
for (indice = 0;  
     indice < 100;  
     indice++) {  
  
    printf("%d, ", vetor[indice]);  
  
}
```

Acesso

Operador de índice:



```
vetor[i+j] = k * 2;
```



Acesso

Imprimir uma lista de trás para frente:

```
int main(int argc, char *argv[]) {  
    int valores[10];  
    int indice;  
    printf("Escreva 10 números inteiros: ");  
    for (indice = 0; indice < 10; indice++) {  
        scanf("%d", &valores[indice] );  
    }  
    printf("Valores em ordem reversa:\n");  
    for (indice = 9; indice >= 0; indice--) {  
        printf("%d ", valores[indice]);  
    }  
    return 0;  
}
```

Vetores\Reverso01\Reverso01.vcproj

Conteúdo inicial

Declaração com conteúdo inicial:

```
tipo vetor [n] = {elem0, elem1, ..., elemn-1}
```



Tamanho do
vetor



Lista de n
valores

Exemplo:

```
int impares[5] = {1, 3, 5, 7, 9};
```

Conteúdo inicial

Declaração com conteúdo inicial :

```
tipo vetor [ ] = { elem0, elem1, ..., elemn-1 }
```



Tamanho do
vetor omitido



Lista de *n*
valores

Exemplo:

```
int impares[] = {1, 3, 5, 7, 9};
```

Vetores

Regras para uso correto



Vetores

Cuidado com índice:

Errado:

```
int vetor[10];
```

```
...
```

```
vetor[-2] = 3;
```

```
vetor[20] = 6;
```

Efeitos
imprevisíveis!

Correto:

```
vetor[4] = 3;
```

```
vetor[7] = 6;
```

Vetores

Atribuir todos os valores:

Errado:

```
int vetor[10];  
...  
vetor = 0;
```

Correto:

```
int indice;  
for (indice = 0; indice < 10; indice++) {  
    vetor[indice] = 0;  
}
```



Vetores

Copiar todos os valores:

Errado:

```
int vetorA[10], vetorB[10];  
...  
vetorA = vetorB;
```

Correto:

```
int indice;  
for (indice = 0; indice < 10; indice++) {  
    vetorA[indice] = vetorB[indice];  
}
```

Vetores

Vetor de tamanho variável

Vetor de tamanho variável

Solução Simples:

```
int valores[100];  
int numero_elementos;
```

Limite superior
para tamanho

Tamanho utilizado

```
int i;  
for (i = 0; i < numero_elementos; i++) {  
    printf("%d", valores[i]);  
}
```

Vetor de tamanho variável

Imprimir uma lista de números de trás para frente:

```
int main(int argc, char *argv[]) {
    int valores[100];
    int numero_valores;
    int i;

    printf("Quantos valores? (no máximo 100) ");
    scanf("%d", &numero_valores);
    if ( (numero_valores < 1) || (numero_valores > 100)) {
        printf("Quantidade invalida!\n");
        return 1;
    }

    ...
}
```



Vetores

Imprimir uma lista de números de trás para frente:

```
...  
    printf("Escreva os números: ");  
    for (i = 0; i < numero_valores; i++) {  
        scanf("%d", &valores[i] );  
    }  
    printf("Valores em ordem reversa:\n");  
    for (i = numero_valores-1; i >= 0; i--) {  
        printf("%d ", valores[i]);  
    }  
    printf("\n");  
    return 0;  
}
```

Vetores\Reverso02\Reverso02.vcproj

Vetores

Declaração de Matrizes

Matrizes

Conceitos:

- *Tabela* de valores
- Acesso por *dois* índices
- Linha e coluna
- Numeração: 0 até *linhas* – 1
0 até *colunas* – 1

Outras propriedades:

- Valores do mesmo tipo
- Acesso aos valores através de um único nome de variável
- Dimensões fixas

```
int a[4][10];
```

a_{00}	a_{01}	a_{02}	a_{03}	a_{04}	a_{05}	a_{06}	a_{07}	a_{08}	a_{09}
a_{10}	a_{11}	a_{12}	a_{13}	a_{14}	a_{15}	a_{16}	a_{17}	a_{18}	a_{19}
a_{20}	a_{21}	a_{22}	a_{23}	a_{24}	a_{25}	a_{26}	a_{27}	a_{28}	a_{29}
a_{30}	a_{31}	a_{32}	a_{33}	a_{34}	a_{35}	a_{36}	a_{37}	a_{38}	a_{39}

Matrizes

Declaração:

tipo	<i>variavel</i>	[linhas]	[colunas]
↓	↓	↓	↓
Qualquer tipo C	Nome da variável	Número de linhas	Número de colunas

Exemplo:

```
int matriz[4][10];
```

```
int a[4][10];
```

a ₀₀	a ₀₁	a ₀₂	a ₀₃	a ₀₄	a ₀₅	a ₀₆	a ₀₇	a ₀₈	a ₀₉
a ₁₀	a ₁₁	a ₁₂	a ₁₃	a ₁₄	a ₁₅	a ₁₆	a ₁₇	a ₁₈	a ₁₉
a ₂₀	a ₂₁	a ₂₂	a ₂₃	a ₂₄	a ₂₅	a ₂₆	a ₂₇	a ₂₈	a ₂₉
a ₃₀	a ₃₁	a ₃₂	a ₃₃	a ₃₄	a ₃₅	a ₃₆	a ₃₇	a ₃₈	a ₃₉



Matrizes

Acesso:

Elementos do vetor:

```
vetor[0][0], vetor[0][1], vetor[0][2], ...  
vetor[linhas-1][colunas-1]
```

Atribuição:

```
vetor[linha][coluna] = valor;
```

Exemplo:

```
int vetor[6][10];  
vetor[5][1] = 3;  
vetor[0][2] = vetor[1][0] + vetor[2][3];
```



Matrizes

Acesso:

Exemplo:

```
int lin, col;
int matriz[4][10];
...
for (lin = 0; lin < 4; lin++) {
    for (col = 0; col < 10; col++) {
        printf("%d ", matriz[lin][col]);
    }
    printf("\n");
}
```

Vetores multidimensionais

Vetor com n dimensões:

<code>tipo</code>	<code>variavel</code>	<code>[dim₁]</code>	<code>[dim₂]</code>	<code>...</code>	<code>[dim_N]</code>	<code>;</code>
↓	↓	└─┘ ↓	└─┘ ↓		└─┘ ↓	
Qualquer tipo C	Nome da variável	Dimensão 1	Dimensão 2		Dimensão N	

Vetores

*Vetor de caracteres
(Texto)*

Texto

Conceitos:

- Caracteres individuais:
 - 'a', 'A', 'f', '4', '.' ← **aspas simples!**
 - Representam apenas um símbolo, letra ou dígito
- Texto:
 - Seqüência de caracteres
 - "Algoritmos e Programação" ← **aspas duplas!**

Texto

Armazenamento:


- Vetor de caracteres
- Cada caractere do texto \Rightarrow um elemento do vetor
- Último caractere: nulo (`'\0'`)
- Tamanho mínimo do vetor: comprimento do texto + 1

char[20]	U	m		t	e	x	t	o		e	m		C	\0						
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Texto

Declaração:

```
char variavel [tamanho];
```



Tamanho máximo
do texto

Exemplo:

```
char texto[50];
```



Texto

Declaração:

```
char variavel [tamanho] = "texto";
```

Exemplo: `char texto[50] = "João da Silva";`

```
char variavel [] = "texto";
```

Exemplo: `char texto[] = "João da Silva";`



Texto

Ler e imprimir:

Impressão:

```
char texto[] = "Um texto em C";  
printf("A variável: %s", texto);
```

Leitura:

```
char texto[20];  
printf("Escreva uma palavra: ");  
scanf("%s", texto);
```

- OBS:
- Leitura por palavra
 - Não usa &
 - Cuidado com tamanho da variável



Texto

Atribuir à uma variável tipo texto:

Errado:

```
char texto[100];  
...  
texto = "João da Silva";
```

Correto:

```
#include <string.h>  
...  
strcpy(texto, "João da Silva");
```

Texto

Acessar caracteres:

```
char texto[20] = "Um texto em C";  
texto[12] = 'B';  
printf("%s", texto);
```

char[20]	U	m		t	e	x	t	o		e	m	C	\0							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Um texto em B

Vetores

The background of the slide features a faded image of a traditional Chinese abacus (suanpan). The abacus has multiple vertical rods with black beads. A person's hands are visible at the bottom, with fingers positioned to move the beads. The entire slide is framed by a light blue border.

Estatística [calcula média e desvio padrão]

Palind [testa palindromos]

Primos [pegar primos, método da peneira]

Strings1 [teste com cadeias e funções sobre cadeias]

Concat [concatena cadeias]

Anagrama [testa anagramas]

SomaMat [soma de matrizes]

QuadradMag [testa quadrado mágico]

Vetores

Fim do Capítulo