

# CP63B-DPGR3A

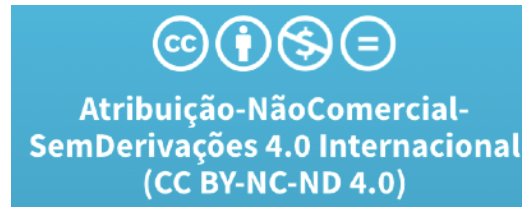
# COMPUTAÇÃO 2

ADNP 02 - Vetores

Prof. Rafael Gomes Mantovani

# Licença

Este trabalho está licenciado com uma Licença CC BY-NC-ND 4.0:



maiores informações:

[https://creativecommons.org/licenses/by-nc-nd/4.0/deed.pt\\_BR](https://creativecommons.org/licenses/by-nc-nd/4.0/deed.pt_BR)

# Roteiro



- 1** Objetivo
- 2** Vetores
- 3** Como usar
- 4** Exercícios
- 5** Referências

# Roteiro

- 1** Objetivo
- 2** Vetores
- 3** Como usar
- 4** Exercícios
- 5** Referências

# Objetivo



**vetor**

# Objetivo



vetor



o que é um **vetor**?  
onde usamos?  
como usamos?

# Roteiro



- 1 Objetivo
- 2 Vetores
- 3 Como usar
- 4 Exercícios
- 5 Referências

# Vetores

Suponhamos que queremos organizar uma ...



festa



# Vetores



festa

para facilitar o controle e  
acesso da festa, podemos  
automatizar o processo por  
meio de um programa



programa em C

# Vetores



festa

Nosso programa vai então armazenar informações dos convidados:

- nome
- documento
- se pagou ou não
- etc ...



programa em C



# Vetores



Quantos inteiros? Dezenas? Centenas?  
Quantos caracteres para armazenar os nomes?  
Que outras variáveis mais para armazenar o que precisa?

- Armazenar (convidado): nome, documento, contato, se pagou ou não, etc

programa em C

# Vetores



Quantos inteiros? Dezenas? Centenas?  
Quantos caracteres para armazenar os nomes?  
Que outras variáveis mais para armazenar o que precisa?

- Armazenar (convidado): nome, documento, contato, se pagou ou não, etc

**precisamos de VETORES**

programa em C

# Vetores

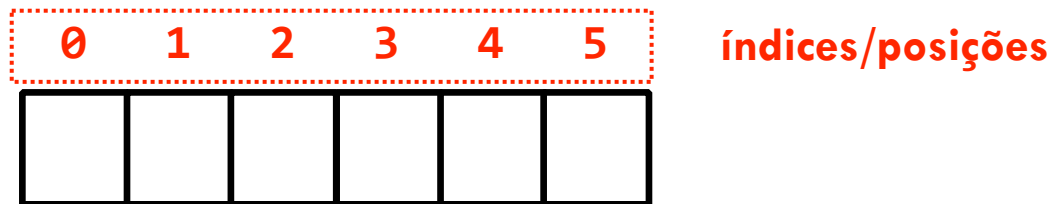
- Em C, um vetor é uma série de variáveis indexadas que podem ser acessadas por meio de um índice inteiro. Por ex:



`vetor[6]`

# Vetores

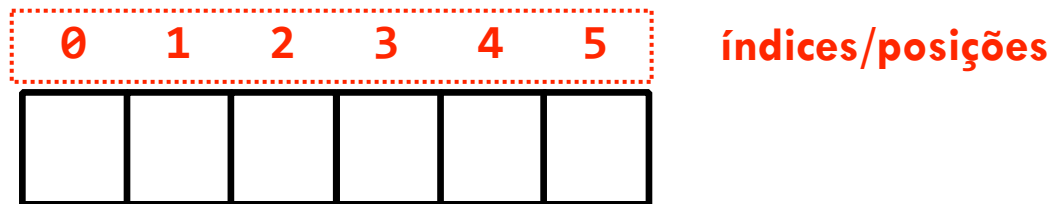
- Em C, um vetor é uma série de variáveis indexadas que podem ser acessadas por meio de um índice inteiro. Por ex:



`vetor[6]`

# Vetores

- Em C, um vetor é uma série de variáveis indexadas que podem ser acessadas por meio de um índice inteiro. Por ex:

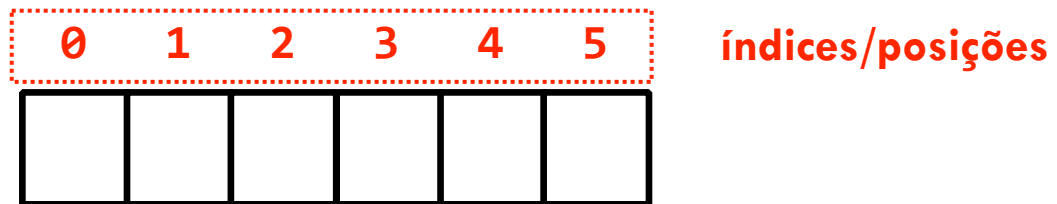


Vetor só guarda variáveis do **mesmo tipo!**

- vetor de inteiros
- vetor de caracteres
- vetor de floats ...

# Vetores

- Em C, um vetor é uma série de variáveis indexadas que podem ser acessadas por meio de um índice inteiro. Por ex:



vetor[6]

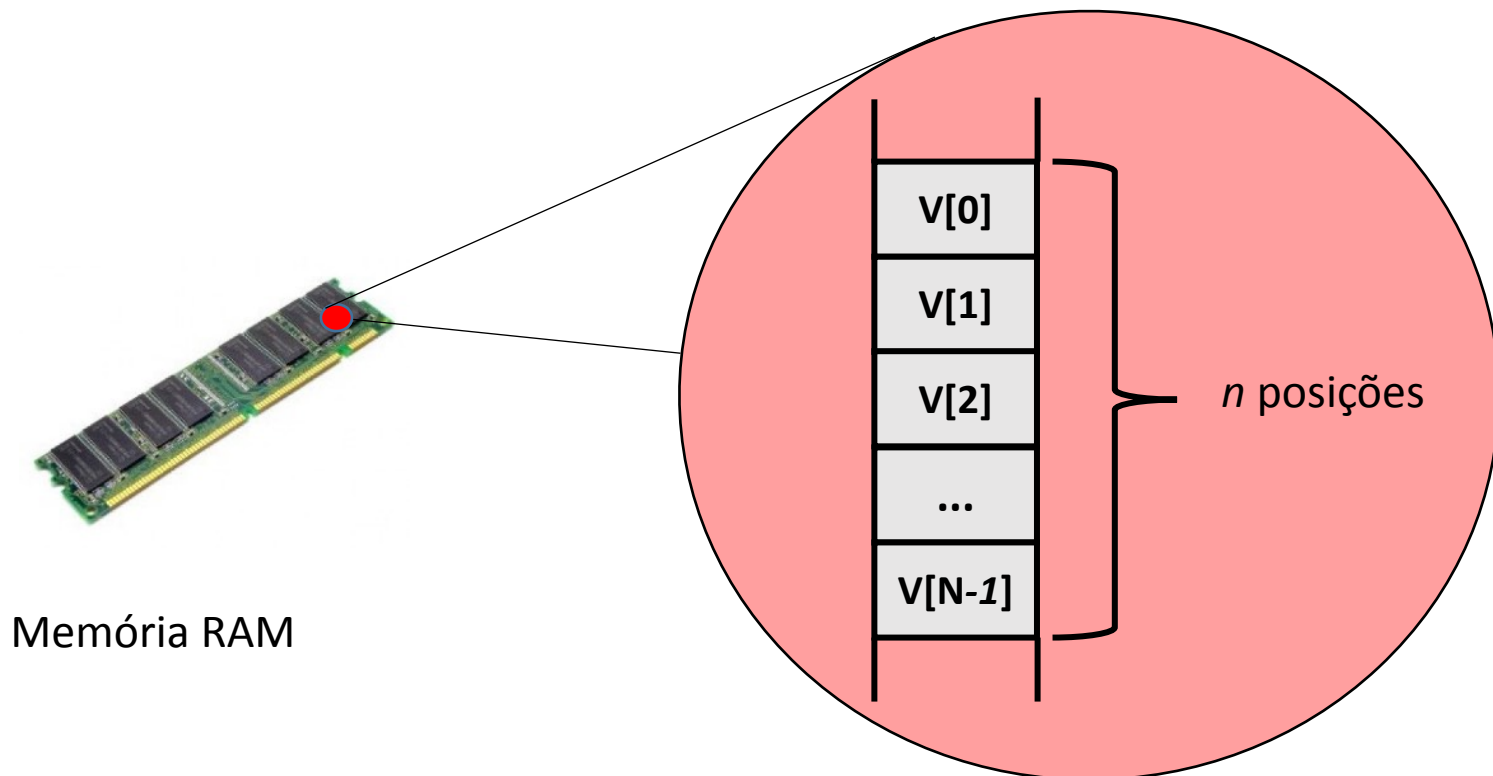
Vetor só guarda variáveis do **mesmo tipo!**

- vetor de inteiros
- vetor de caracteres
- vetor de floats ...



# Vetores

- Essas variáveis são todas guardadas sequencialmente (sem buracos) na memória e, em um vetor de  $N$  elementos, são identificadas por índices de  $0$  a  $N - 1$



# Roteiro



- 1 Objetivo
- 2 Vetores
- 3 Como usar
- 4 Exercícios
- 5 Referências

# Como usar

- Para usar um vetor, primeiro é preciso declará-lo, como era feito para qualquer variável comum:

```
Tipo_de_dado nome_vetor[tamanho];
```

O compilador entende esse comandos como:

*Reserve na memória um espaço para **tamanho** variáveis do tipo **tipo\_de\_dado**, e chame esse espaço de **nome\_vetor***

```
int sequencia[5];
```

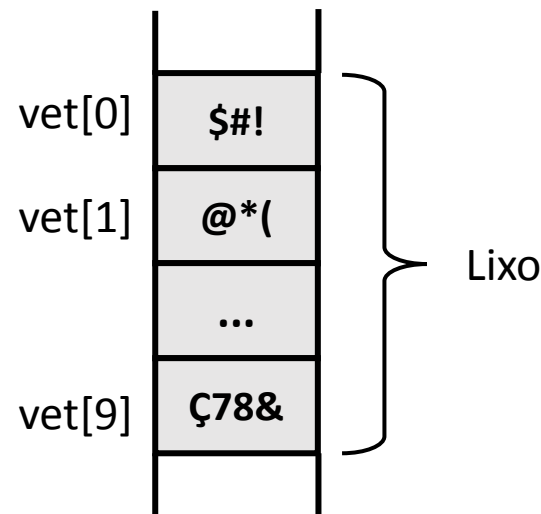
```
float nota[100];
```

# Como usar



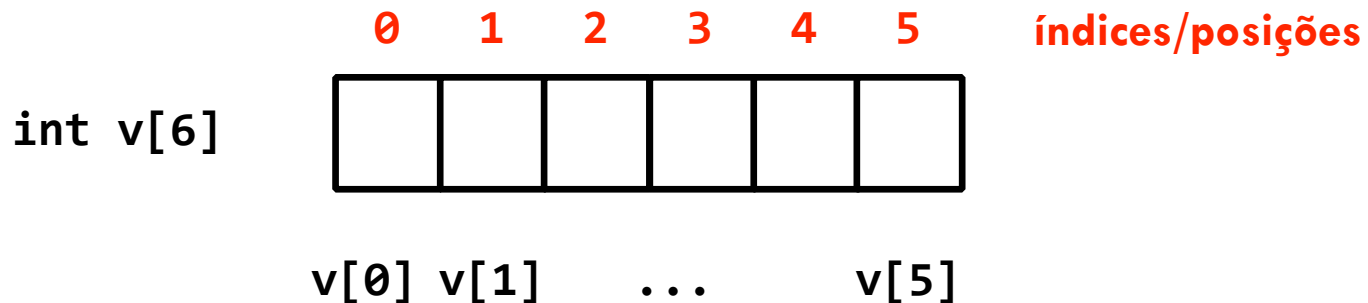
- O compilador apenas “*reserva*” o espaço de memória pedido, sem colocar nenhum valor especial nele;
- Isso significa que o vetor conterá inicialmente uma seleção “aleatória” de valores
  - Provavelmente que sobraram da execução de algum programa que usou aquele espaço, exatamente como ocorria para as variáveis comuns.

```
int vet[10];
```



# Como usar

- Os elementos de um vetor são numerados a partir de zero:



- O tamanho do vetor deve ser um valor **constante**
  - O tamanho do vetor é imutável;
  - Deve-se estabelecer um tamanho que conseguirá ser suficiente para a aplicação, caso contrário deve-se usar *alocação dinâmica*;

# Vetores

- Inicialização de vetores

Tipo\_de\_dados nome\_vetor[tamanho] = {lista de valores};

```
int lista[5] = { 9, 35, -17, 8, 15};
```

- Se forem especificados todos os elementos do vetor, não é necessário indicar o tamanho

```
int lista[] = { 9, 35, -17, 8, 15};
```

```
int lista[]; —————> Errado!!!
```

- Inicializar qualquer elemento do vetor

```
int lista[5];  
  
lista[0] = 9;  
lista[2] = -17;  
lista[1] = 35;
```

# Roteiro



- 1** Objetivo
- 2** Vetores
- 3** Como usar
- 4** Exercícios
- 5** Referências

# Exercícios

1. Crie um vetor de 5 posições e imprima sem modificar os valores iniciais contidos no vetor.
2. Crie e inicialize um vetor de inteiros e faça a soma dos seus elementos. Apresente o resultado.
3. Leia 5 números e imprima em ordem inversa de leitura.
4. Leia um vetor A com 10 elementos inteiros correspondentes as idades de um grupo de pessoas. Escreva um programa que determine e escreva a quantidade de pessoas que possuem idade superior a 35 anos.
5. Dados dois vetores, A (4 elementos) e B (5 elementos), faça um programa em C que imprima todos os elementos comuns aos dois vetores.



# Resposta: ex 1

**Obs.:** %i ou %d identificam inteiros

```
1. #include<stdio.h>
2.
3. int main(){
4.     // declaração de variáveis
5.     int i, num[5];
6.
7.     // laço de repetição para acessar vetor
8.     for(i=0; i<=4; i++) {
9.         printf("%d ", num[i]);
10.    }
11.
12.    return 0;
13. }
```

# Resposta: ex 3

```
1.#include<stdio.h>
2.
3. int main(){
4.
5.     int i, num[5]; //declaracao de variaveis
6.
7.     for(i=0;i<=4;i++) { //leitura dos valores
8.         printf("Digite a nota %i: ", i);
9.         scanf("%d", &num[i]);
10.    }
11.
12.    for(i=4;i>=0;i--) { //impressao em ordem inversa
13.        printf("%d ", num[i]);
14.    }
15.
16.    return 0;
17. }
```

# Exercícios [Dificuldade = Médio]

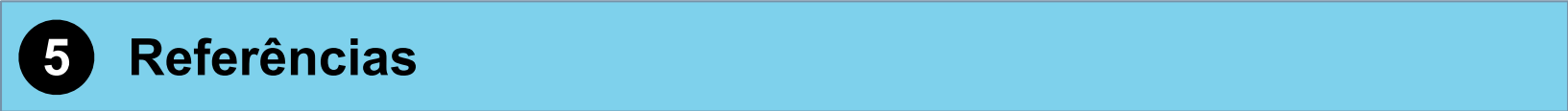
6. Leia um vetor de 10 posições e imprima o maior valor, menor valor e a média dos valores contidos no vetor.
7. Números *palíndromos* são aqueles que escritos da direita para a esquerda têm o mesmo valor quando escritos da esquerda para a direita. Exemplo: 545; 789987; 97379; 123454321; etc. Escreva um programa que verifique se um dado vetor A de 10 elementos inteiros é um palíndromo, ou seja, se o primeiro elemento do vetor é igual ao último, se o segundo elemento do vetor é igual ao penúltimo e assim por diante até verificar todos os elementos ou chegar a conclusão que o vetor não é um palíndromo.
8. Criar dois vetores A e B cada um com 10 elementos inteiros. Desenvolver um programa que crie um vetor C que é a diferença dos conjuntos formados pelos elementos dos vetores A e B. Diferença de conjuntos = todos os elementos do conjunto A que não existem no conjunto B.

# Exercícios Extras

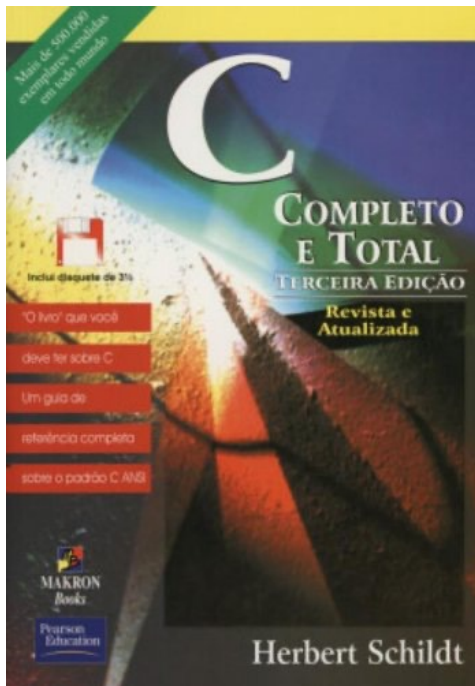
9. Crie um vetor que armazene 10 valores inteiros fornecidos pelo usuário. Mostre qual é o menor elemento e a posição em que ele se encontra no vetor.
10. Escreva um algoritmo que leia e mostre um vetor de 10 números inteiros. A seguir, conte quantos valores pares existem no vetor.
11. Faça um algoritmo que leia 10 valores inteiros e armazene em um vetor A. Inverta então os valores de A. Troque o primeiro pelo último, segundo pelo penúltimo e assim por diante. Mostre o vetor A após as alterações.

# Roteiro

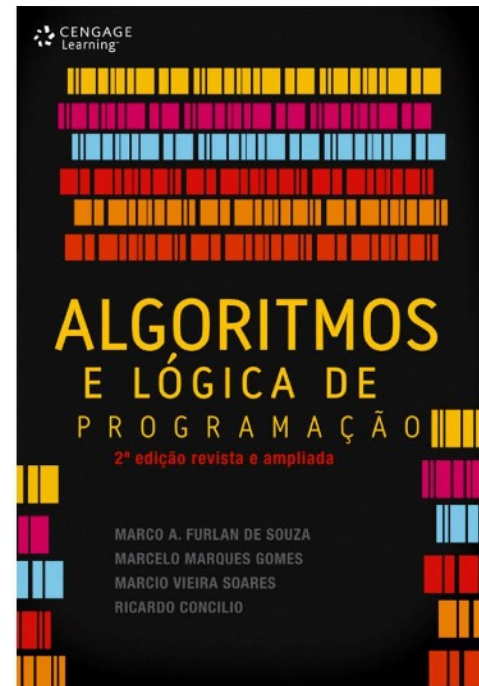


- 1 Objetivo
  - 2 Vetores
  - 3 Como usar
  - 4 Exercícios
  - 5 Referências
- 

# Referências



[Schildt, 1997]



[de Souza et al, 2011]

# Referências

- [Schildt, 1997] SCHILDT, H. **C Completo e Total**. 3. ed. São Paulo: Pearson, 1997.
- [de Souza et al, 2011] DE SOUZA, M. A. F. et al. **Algoritmos e lógica de programação**. 2. ed. São Paulo: Cengage Learning, 2011.

# Perguntas?

Prof. Rafael G. **Mantovani**

[rafaelmantovani@utfpr.edu.br](mailto:rafaelmantovani@utfpr.edu.br)