

# Algoritmos e Lógica de Programação

- Douglas Baptista de Godoy

# Ementa

- Projeto e representação de algoritmos.
- Estruturas de controle de fluxo de execução: sequência, seleção e repetição.
- Tipos de dados básicos e estruturados (vetores e registros).
- Rotinas. Arquivos.
- Implementação de algoritmos usando uma linguagem de programação.

# Objetivo

- Analisar problemas computacionais e projetar soluções por meio da construção de algoritmos.

# Vetores

- **Definição de Vetor**
- As variáveis compostas homogêneas unidimensionais (ou, simplesmente, vetores) são capazes de armazenar diversos valores. Cada um desses valores é identificado pelo mesmo nome (o nome dado ao vetor), sendo diferenciado entre si apenas por índice.
- Os índices utilizados na linguagem C/C++ para identificar as posições de um vetor começam sempre em 0 (zero) e vão até o tamanho do vetor menos uma unidade. O índice de um vetor em C/C++ deve sempre ser representado por um dos tipos inteiros disponíveis na linguagem.

Fonte: Fundamentos da Programação de Computadores, Pearson Editora, 3ª edição

# Vetores

- **Declaração de Vetor**
- Os vetores em C/C++ são identificados pela existência pela existência de colchetes logo após o nome da variável no momento da declaração. Dentro dos colchetes, deve-se colocar o numero de elementos que o vetor poderá armazenar.
- Em C/C++, a indicação do tamanho do vetor (ou seja, a quantidade de elementos que o vetor poderá armazenar) deve ser feita por um inteiro fixo(representado por um literal ou uma constante). Se houver necessidade de definir o tamanho do vetor em tempo de execução, deve-se fazê-lo através de ponteiros.

Fonte: Fundamentos da Programação de Computadores, Pearson Editora, 3ª edição

# Vetores

- Exemplo de Vetor

## Declaração de Vetor

Exemplo 1:  
`int vet[10];`

Exemplo 2:  
`char x[5];`

Exemplo 3:  
`#define tam 5;`  
`char z[tam];`

## Atribuindo valores ao Vetor

Exemplo 1:  
`vet[0] = 1;`

Exemplo 2:  
`x[3] = 'b' ;`

Exemplo 3:  
`for (i = 0; i < 10; i++)`  
`scanf("%d%c", &vet[i]);`

## Mostrando elementos do Vetor

Exemplo 1:  
`printf("%d", vet[5]);`

Exemplo 2:  
`printf("%c", x[3]);`

Exemplo 3:  
`for (i = 0; i < 10; i++)`  
`printf("%d", vet[i]);`

Fonte: Fundamentos da Programação de Computadores, Pearson Editora, 3ª edição

# Vetores

– Vetor é uma **variável composta homogênea unidimensional**

```
#include <stdio.h>

int main(){
    //Criando variáveis para armazenar 10 números flutuantes (não é o ideal)
    float n1, n2, n3, n4, n5, n6, n7, n8, n9, n10;

    //Criando um vetor para armazenar 10 números flutuantes
    float n[10];

    return 0;
}
```

</>  
vetor.c

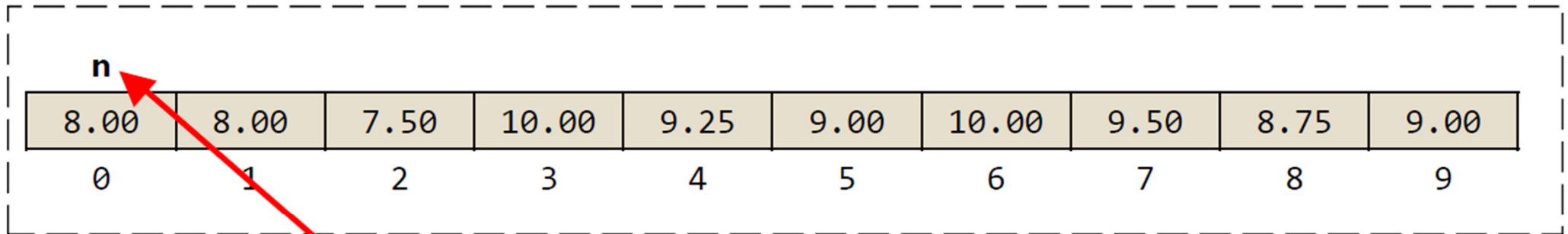
SLIDE: PROF. LUCAS MATHEUS GOMES DA SILVA

Fonte: Fundamentos da Programação de Computadores, Pearson Editora, 3ª edição

# Vetores

- Como poderíamos armazenar 10 números flutuantes na memória usando um vetor?

## Memória RAM



- **Composta** porque através de uma **única variável** podemos armazenar vários valores

SLIDE: PROF. LUCAS MATHEUS GOMES DA SILVA

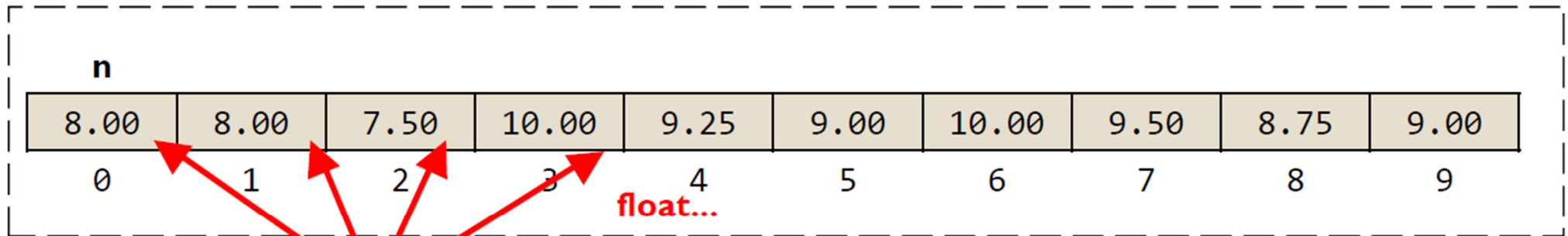
Fonte: Fundamentos da Programação de Computadores, Pearson Editora, 3ª edição



# Vetores

- Como poderíamos armazenar 10 números flutuantes na memória usando um vetor?

## Memória RAM



- **Homogênea** porque todos os dados armazenados em um vetor são do **mesmo tipo** (na linguagem C)

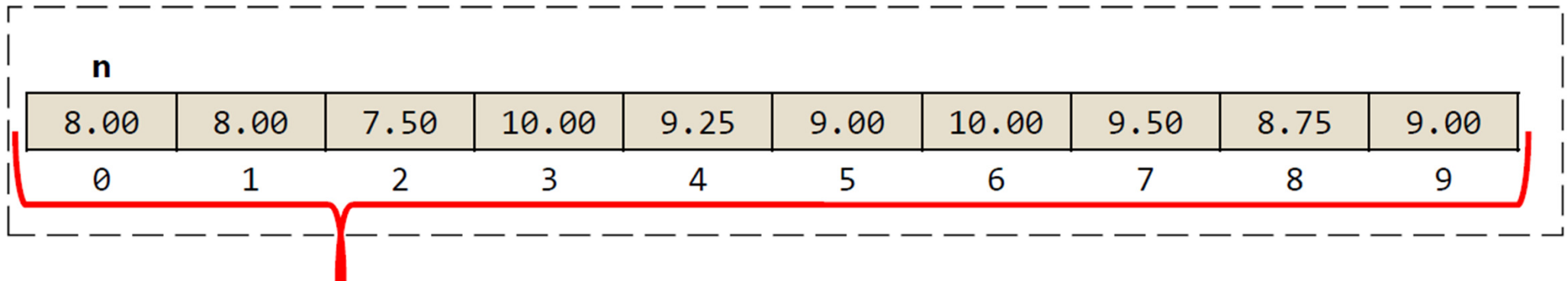
SLIDE: PROF. LUCAS MATHEUS GOMES DA SILVA

Fonte: Fundamentos da Programação de Computadores, Pearson Editora, 3ª edição

# Vetores

- Como poderíamos armazenar 10 números flutuantes na memória usando um vetor?

## Memória RAM



- **Unidimensional** porque são **alocadas sequencialmente na memória** do computador

SLIDE: PROF. LUCAS MATHEUS GOMES DA SILVA

Fonte: Fundamentos da Programação de Computadores, Pearson Editora, 3ª edição

# Vetores

- Como poderíamos armazenar 10 números flutuantes na memória usando um vetor?

## Memória RAM

n									
8.00	8.00	7.50	10.00	9.25	9.00	10.00	9.50	8.75	9.00
0	1	2	3	4	5	6	7	8	9

- Como a variável tem o mesmo nome, **podemos acessar cada posição do array através do índice** (que começa em 0)

SLIDE: PROF. LUCAS MATHEUS GOMES DA SILVA

Fonte: Fundamentos da Programação de Computadores, Pearson Editora, 3ª edição

# Referencias Bibliográficas

- ASCENCIO, Ana Fernanda Gomes; CAMPOS, Edilene Aparecida Veneruchi de, **Fundamentos da Programação de Computadores**, Pearson Editora, 3ª edição.