



# Algoritmos e Programação II

Prof. Joilson dos Reis Brito  
Prof.<sup>a</sup> Noeli A. Pimentel Vaz

# Agenda da Aula

- Carga Inicial de Vetores
- Utilização de Constantes
- Exemplos

# CARGA INICIAL AUTOMÁTICA DE VETORES

preco					
9.1	4.5	75.3	84	19.2	1.88
0	1	2	3	4	5

# Carga Inicial Automática de Vetores

Quando utilizamos vetores para resolver problemas é preciso estar atento, pois quando declaramos um vetor ele contém inicialmente valores aleatórios (LIXO) em cada uma de suas posições.



# Carga Inicial Automática de Vetores

Em todos os exercícios que foram desenvolvidos, os vetores foram inicializados por meio de leitura (scanf).

É possível também, iniciar automaticamente todos os elementos do vetor, usando a seguinte sintaxe:  
tipo vetor[n] = {valor1, valor2, ... , valorn};

# Carga Inicial Automática de Vetores

## Exemplo:

Declare e inicie um vetor com todas as vogais do alfabeto.

```
char Vogal[5] = {'a', 'e', 'i', 'o', 'u'};
```

Evita-se, assim, escrever o seguinte conjunto de código:

```
char Vogal[5];
```

```
Vogal[0] = 'a';
```

```
Vogal[1] = 'e';
```

```
Vogal[2] = 'i';
```

```
Vogal[3] = 'o';
```

```
Vogal[4] = 'u';
```

# Carga Inicial Automática de Vetores



Se um vetor numérico(int ou float) for declarado com  $N$  elementos e forem colocados apenas  $K$  valores ( $K < N$ ) na carga inicial do vetor, então os primeiros  $K$  elementos do vetor serão iniciados com os respectivos valores e os restantes serão iniciados com o valor ZERO.

# Carga Inicial Automática de Vetores

## Exemplo:

```
int Valor[10] = {5, 12, 30};
```

Neste exemplo os três primeiros elementos do vetor (índices 0, 1 e 2) ficam iniciados com os valores 5, 12 e 30, respectivamente e todos os outros ficam iniciados com o valor 0.

## Declarações equivalentes:

```
int Valor[10] = {5, 12, 30};
```

```
int Valor[10] = {5, 12, 30, 0, 0, 0, 0, 0, 0, 0};
```



# Carga Inicial Automática de Vetores

## Exemplos de declaração:

```
int V[10];
```

```
int V[3] = {5, 10, 15};
```

```
int V[10] = {5, 10, 15};
```

```
int V[] = {5, 10, 15};
```

```
int V[]; //declaração incorreta, erro de compilação!
```

[Error] storage size of 'V' isn't known

# Constantes

O desenvolvimento dos programas deve permitir que pequenas modificações, como por exemplo o tamanho do vetor, não provoque grandes transformações no código.

## **Exemplo:**

Programa que lê 10 notas em um vetor, calcula média e mostra as notas acima da média.

```
int main()
{
    float Notas[10], Soma = 0, Media;
    int I;
    for(I = 0; I < 10; I++)
    {
        printf("Digite a nota do aluno %d: ", I+1);
        scanf("%f",&Notas[I]);
        Soma = Soma + Notas[I];
    }
    Media = Soma/10;
    printf("\nMostrando as notas acima da Media:");
    for(I = 0; I < 10; I++)
    {
        if(Notas[I] >= Media)
        {
            printf("\n%.1f",Notas[I]);
        }
    }
    ...
}
```

# Constantes

## Exemplo:

Observe que neste exemplo, foi necessário fazer uso do tamanho do vetor (valor 10) em três locais no código.

Para facilitar as alterações podemos fazer uso de **Constantes!**

# Constantes

Uma constante é um nome correspondente a um valor fixo (não se pode alterar ao longo de uma execução).

As constantes devem ser definidas, no início do código, abaixo das bibliotecas, de modo a serem visíveis ao longo de todo o programa.

Constantes podem ser definidas de duas maneiras:

Através da palavra reservada `const`:

**`const tipo simbolo = valor;`**

Através da diretiva `#define`

**`#define simbolo valor`**

**Na disciplina será utilizada a diretiva `#define`.**

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

```
#define NUM 10
```

# Exemplo Constante - define

```
int main()
{
    float Notas[NUM], Soma = 0, Media;
    int I;
    for(I = 0; I < NUM; I++)
    {
        printf("Digite a nota do aluno %d: ", I+1);
        scanf("%f",&Notas[I]);
        Soma = Soma + Notas[I];
    }
    Media = Soma/NUM;
    printf("\nMostrando as notas acima da Media:");
    for(I = 0; I < NUM; I++)
    {
        if(Notas[I] >= Media)
        {
            printf("\n%.1f",Notas[I]);
        }
    }
    ...
}
```

# #define

- As constantes definidas com o símbolo **#define** chamam-se **Constantes Simbólicas**.
- Embora não seja obrigatório, habitualmente os programadores colocam as constantes simbólicas maiúsculas.