


Introdução à Programação

Aula 09 – Vetores

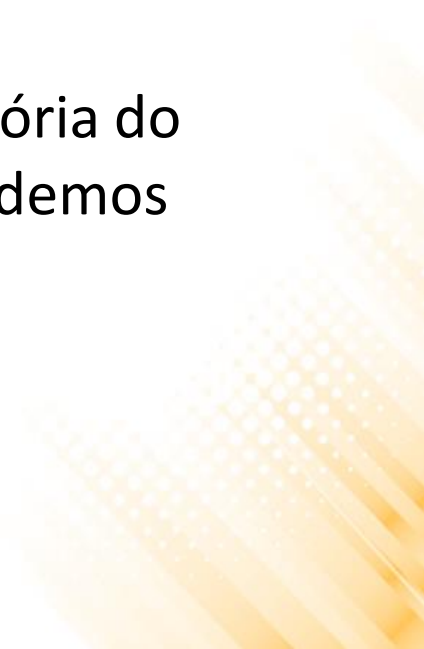
Edirlei Soares de Lima
<edirlei@iprj.uerj.br>



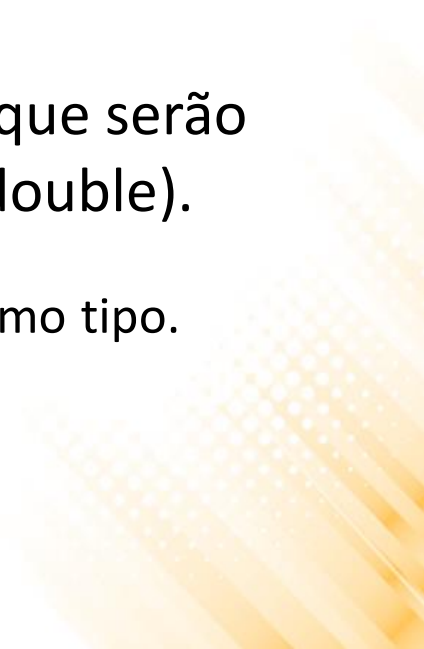
Introdução

- Até agora nós temos usado variáveis simples para armazenar valores usados por nossos programas.
- Em várias situações, precisamos armazenar um **conjunto de valores**.
- A partir de agora vamos aprender a usar um mecanismo que nos permite armazenar um conjunto de valores na memória do computador.
 - Posteriormente, estes valores podem ser livremente processados de forma eficiente, pois já estariam na memória do computador.

Vetores

- Podemos armazenar um conjunto de valores na memória do computador através do uso de **vetores (arrays)**
 - O vetor é a forma mais simples de **organizarmos dados** na memória do computador.
 - Com vetores, os valores são armazenados na memória do computador **em sequência**, um após o outro, e podemos livremente acessar qualquer valor do conjunto.
- 

Vetores

- Ao declarar um vetor (conceito análogo ao de declaração de uma variável simples) é necessário informar **dimensão do vetor**.
 - O tamanho define o **número máximo de elementos** que poderá ser armazenado no espaço de memória que é reservado para o vetor.
 - Também é necessário informar o **tipo dos valores** que serão armazenados no vetor (por exemplo, int, float ou double).
 - Em um vetor, só podemos armazenar valores de um mesmo tipo.
- 

Declaração e Inicialização

- **Declaração de um vetor:**

```
int meu_vetor[10];
```

- Reserva um espaço de memória para armazenar 10 valores inteiros no vetor chamado meu_vetor.

- **Inicialização de algumas posições do vetor meu_vetor:**

```
meu_vetor[0] = 5;  
meu_vetor[1] = 11;  
meu_vetor[4] = 0;  
meu_vetor[9] = 3;
```

5	11	?	?	0	?	?	?	?	3
0	1	2	3	4	5	6	7	8	9

Declaração e Inicialização

- É possível **acessar os valores do vetor** através de seu **índice**.

```
int meuvetor[5];
```

0	1	2	3	4
5	?	?	8	1

```
meuvetor[0] = 5;
```

```
meuvetor[3] = 8;
```

```
meuvetor[4] = 1;
```

Declaração e Inicialização

- **Exemplos de Declaração:**

```
int a, b[20];  
  
float c[10];  
  
double d[30], e, f[5];
```

- **Declaração e Inicialização:**

```
int teste[5] = {12, 5, 34, 32, 9};  
  
float vetor1[3] = {2.5, 5.8, 10.1};
```

Exemplo 1: Imprimindo os Valores Armazenados em um Vetor

```
#include <stdio.h>

int main(void)
{
    int i;
    float v[6] = {2.3, 5.4, 1.0, 7.6, 8.8, 3.9};

    for (i=0; i<6; i++)
    {
        printf("%f", v[i]);
    }

    return 0;
}
```


Exemplo 2: Somatório dos Valores Armazenados em um Vetor

```
#include <stdio.h>

int main(void)
{
    int i;
    float v[6] = {2.3, 5.4, 1.0, 7.6, 8.8, 3.9};
    float s = 0.0;

    for (i=0; i<6; i++)
    {
        s = s + v[i];
    }
    printf("%f", s);
    return 0;
}
```

Exemplo 3: Encontrar o Maior Valor

```
#include <stdio.h>

int main(void)
{
    int i;
    float v[6] = {2.3, 5.4, 1.0, 7.6, 8.8, 3.9};
    float maior_valor = v[0];
    for (i=0; i<6; i++)
    {
        if(v[i] > maior_valor)
            maior_valor = v[i];
    }
    printf("%f", maior_valor);
    return 0;
}
```

Exemplo 4: Calculo da Média

- Dada uma turma com n alunos (onde n é conhecido a priori), crie um programa para obter a notas dos alunos e calcula a média da turma.

7.5

8.4

9.1

4.0

5.7

4.3

```
#include <stdio.h>
#define NUM_ALUNOS 6

int main (void)
{
    float notas[NUM_ALUNOS];
    float media, soma = 0.0;
    int i;
    /* leitura dos dados via teclado para o vetor */
    for(i=0; i<NUM_ALUNOS; i++)
    {
        printf("Entre com a nota do aluno %d: ", i+1);
        scanf("%f", &notas[i]);
    }
    /* soma das medias dos alunos */
    for(i=0; i<NUM_ALUNOS; i++)
        soma = soma + notas[i];
    media = soma/NUM_ALUNOS;
    printf("Media da turma = %.2f\n.", media);
    return 0;
}
```

**Solução 1: com leitura
dos dados via teclado.**

```
#include <stdio.h>
#define NUM_ALUNOS 6

int main (void)
{
    float notas[NUM_ALUNOS];
    float media, soma = 0.0;
    FILE* arq_entrada;
    int i;
    arq_entrada = fopen("notas.txt", "r");
    if (arq_entrada == NULL)
    {
        printf("Erro no arquivo de entrada.");
        return 0;
    }
    /* leitura dos dados - arquivo */
    for(i=0;i<NUM_ALUNOS;i++)
    {
        fscanf(arq_entrada, "%f", &notas[i]);
    }
    fclose(arq_entrada);
}
```

Solução 2: com leitura
dos dado via arquivo.

(continuação)

```
/* soma das medias dos alunos */  
for(i=0;i<NUM_ALUNOS;i++)  
{  
    soma = soma + notas[i];  
}  
media = soma/NUM_ALUNOS;  
printf("Media da turma = %.2f\n", media);  
  
return 0;  
}
```

Exercícios

Lista de Exercícios 08 - Vetores

<http://www.inf.puc-rio.br/~elima/intro-prog/>



Vetores Passados para Funções

- Além de passarmos variáveis simples como **parâmetros**, podemos passar **vetores** também:
 - Quando passamos um vetor como parâmetro, a função chamada recebe uma **referência para o vetor**.
 - Quando a função chamada acessa os elementos do vetor, ela acessa as **mesmas posições de memória** que a função que declarou vetor.
 - Se atribuirmos um valor a um elemento do vetor passado como parâmetro, este elemento **também é alterado no vetor original**.
 - Portanto, podemos declarar um vetor numa função e chamar uma outra função auxiliar para acessar e/ou modificar seus elementos.

Vetores Passados para Funções: Voltando ao Exemplo Anterior

- Com o uso de funções auxiliares podemos **reescrever** o exemplo do **calculador da média** de uma forma mais organizada.
- **Podemos dividir o programa em funções:**
 - Uma função para ler os valores e armazená-los em um vetor
 - Uma função para calcular a média

```
#include<stdio.h>
#define NUM_ALUNOS 6

void ler_dados(float vet[], int num)
{
    int i;
    for(i=0;i<num;i++)
    {
        printf("Entre com o valor %d: ", i+1);
        scanf("%f", &vet[i]);
    }
}

float calcula_media(float vet[], int num)
{
    float soma = 0.0;
    int i;
    for(i=0;i<num;i++)
        soma = soma + vet[i];
    return soma/num;
}
```

(continuação)

```
int main (void)
{
    float notas[NUM_ALUNOS];
    ler_dados(notas, NUM_ALUNOS);
    printf("Media = %.2f\n.", calcula_media(notas, NUM_ALUNOS));
    return 0;
}
```

Exercícios

Lista de Exercícios 09 – Vetores e Funções

<http://www.inf.puc-rio.br/~elima/intro-prog/>



Busca em Vetores

- **Problema:** Implemente uma função que busca por um inteiro em um vetor de inteiros.

```
int busca(int n, int vet[], int elem)
{
    int i;
    for(i = 0 ; i < n ; i++)
    {
        if(vet[i] == elem)
        {
            return i;
        }
    }
    return -1;
}
```

Retorna o índice da primeira ocorrência do elemento procurado!

E se quiséssemos retornar a última ocorrência?

Busca em Vetores

- **Problema:** Implemente uma função que busca por um inteiro em um vetor de inteiros.

```
int busca(int n, int vet[], int elem)
{
    int i, id_elem = -1;
    for(i = 0 ; i < n ; i++)
    {
        if(vet[i] == elem)
        {
            id_elem = i;
        }
    }
    return id_elem;
}
```

Percorre o vetor até o final procurando o elemento desejado. Se achar, guarda seu índice, se não a variável `id_elem` já está com o valor -1 de retorno.

Busca em Vetores

- Podemos modificar o algoritmo anterior para que ele volte a retornar o índice da primeira ocorrência do elemento desejado:

```
int busca(int n, int vet[], int elem)
{
    int i, id_elem = -1;
    for(i = 0 ; i < n ; i++)
    {
        if(vet[i] == elem)
        {
            id_elem = i;
            break;
        }
    }
    return id_elem;
}
```

Este comando é usado para interromper o laço (for, while ou do-while) em que ele for chamado. O programa continua a execução no comando seguinte ao laço.

Busca em Vetores

- **Problema 2:** Criar uma função para encontrar o valor máximo em um vetor.

```
float maximo(int n, float v[])
{
    int i;
    float vmax = v[0];
    for (i = 1; i < n ; i++)
    {
        if (v[i] > vmax)
        {
            vmax = v[i];
        }
    }
    return vmax;
}
```


Busca em Vetores

- **Problema 3:** Criar uma função para encontrar o valor mínimo em um vetor.

```
float minimo(int n, float v[])
{
    int i;
    float vmin = v[0];
    for (i = 1; i < n ; i++)
    {
        if (v[i] < vmin)
        {
            vmin = v[i];
        }
    }
    return vmin;
}
```

Busca em Vetores

- **Problema 4:** Criar uma função para encontrar o índice e o valor máximo em um vetor?



```
#include <stdio.h>
#define CAPACIDADE_VETOR 10

void maximo (int n, float v[], float *vmax, int *imax)
{
    int i ;
    float maior_valor = v[0];
    int ind_maiorvalor = 0;
    for(i = 1; i < n ; i++)
    {
        if (v[i] > maior_valor)
        {
            maior_valor = v[i];
            ind_maiorvalor = i;
        }
    }
    *vmax = maior_valor;
    *imax = ind_maiorvalor;
}
```

(continuação)

```
void leitura_dados (int n, float v[])
{
    int i;
    for(i=0;i<n;i++)
    {
        printf("Digite o valor para armazenar no vetor: ");
        scanf("%f", &v[i]);
    }
}

int main(void)
{
    int vetor[CAPACIDADE_VETOR];
    float maior;
    int indice;
    leitura_dados(CAPACIDADE_VETOR, vetor);
    maximo(CAPACIDADE_VETOR, vetor, &maior, &indice);
    printf("Maior valor %f na posicao %d.\n", maior, indice);
    return 0;
}
```

Exercícios

Lista de Exercícios 10 – Busca em Vetores

<http://www.inf.puc-rio.br/~elima/intro-prog/>

