

# Algoritmos e Programação de Computadores II (APC II)

## Aula 2 – Revisão: Vetores, Matrizes e *Strings*

Profa. Carla Gonçalves Pelissoni



## Vetores

**Vetores =**

**Variáveis Compostas Unidimensionais  
Homogêneas**

- Correspondem a posições de memória, identificadas por um mesmo nome, individualizadas por índices e cujo conteúdo é do mesmo tipo.

# Vetores

- **Exemplo:**

- Conjunto de notas de 100 alunos de uma disciplina.
- Associa-se o identificador **Notas** que passará a identificar não uma única posição de memória, mas 100 posições.
- A referência ao *n-ésimo* elemento do conjunto será identificado pela notação **Notas[i]**, onde *i* é um número inteiro ou uma variável numérica contendo um valor inteiro que identifica a posição do elemento no vetor.

# Vetores

**Exemplo:**

Notas[0] = 9.0

Notas[2] = 5.5

Pode-se usar um índice (i) para se ter acesso a qualquer posição do vetor.

i = 5, Notas[i] = 4.0

Notas

9.0	0
8.0	1
5.5	2
6.5	3
2.0	4
4.0	5

## Vetores

- Outros exemplos:

Nome	
'J'	0
'O'	1
'A'	2
'O'	3

Idade	
18	0
54	1
40	2
68	3
20	4
4	5

Disciplina APC I - Profa. Carla G. Pelissoni

5

Grupo **ibmec**  
METROCAMP **IBTA**

## Vetores

- **Declaração de uma Variável do Tipo Vetor (em Linguagem C):**

```
tipo nome[TAMANHO];
```

- Onde:
  - **nome** é o nome que se dá ao conjunto
  - **tipo** é o tipo que se quer associar o conjunto (*int*, *float*, *char*, por exemplo)
  - **TAMANHO** é a quantidade máxima de elementos do vetor.

- **Exemplo:**

```
float Notas[10];           limite inferior = notas[0]
                           limite superior = notas[9]
```

Disciplina APC I - Profa. Carla G. Pelissoni

6

Grupo **ibmec**  
METROCAMP **IBTA**

# Vetores

- **Acessando** um elemento do vetor
  - É preciso utilizar um **índice sempre!**
  - Notas[2]: representa o terceiro elemento do vetor (primeiro elemento é o de posição 0).
  - Pode-se usar uma variável para representar o índice para se ter acesso a qualquer posição do vetor. Como por exemplo, variável *i*: **Notas[i]**.
  - Se *i* = 5, estará sendo feito o acesso à 6ª posição do vetor Notas (**pois o 1º índice é sempre 0**).

# Vetores

- **Armazenando** dados no vetor (Exemplo: armazena 10 elementos)

```
.....
for (i=0; i<10; i++)
{
    cout << "Digite a nota do aluno:";
    cin >> Notas[i];
}
```

## Vetores

- **Utilizando** dados do vetor de 10 elementos (soma dos elementos do vetor e cálculo da média aritmética):

```
.....
soma=0;
for (i=0; i<10; i++)
{
    soma = soma + Notas[i];
}
media = soma/10;
cout << "Media das notas: " << media;
.....
```

## Vetores

- **Mostrando** na tela os dados de um vetor (Exemplo: 10 elementos)

```
.....
for (i=0; i<10; i++)
{
    cout << "\nElemento " << i << ": ";
    cout << Notas[i];
}
```

## Exemplo 1

- Escrever um programa em C que leia um conjunto de 10 notas, armazene-as num vetor Notas e calcule a média aritmética dessas notas.

## Exemplo 1

```
#include<iostream.h>
main()
{
    int i;
    float Notas[10], media, soma=0;

    /* Faz a leitura de cada nota e a somatória com a
       variável soma. */
    for (i=0; i < 10; i++)
    {
        cout << "\nDigite uma nota: ";
        cin >> Notas[i];
        soma = soma + Notas[i];
    }
    media = soma/10;
    cout << "Media da turma = " << media;
}
```

## Exemplo 2

- Fazer um programa em C que leia um conjunto de 50 notas e armazene-as num vetor. Calcule e escreva a quantidade de alunos com nota superior à 7.0.

## Exemplo 2

```
#include<iostream.h>
main()
{
    int i;
    float Notas[50], cont=0;
    for (i=0; i < 50; i++)
    {
        cout << "\nDigite uma nota: ";
        cin >> Notas[i];
        if(Notas[i] > 7)
            cont++;
    }
    cout << "Quantidade de notas maior que 7 = " <<
    cont;
}
```

## Exemplo 3

- Faça um programa em C para ler 5 números inteiros quaisquer e armazená-los em um vetor. Em seguida, imprimir o vetor original e o mesmo em ordem inversa.
- Por exemplo:  
 vetor original:     5 -3 6 67 -10  
 ordem inversa:   -10 67 6 -3 5

## Exemplo 3

```
#include <iostream.h>
main(){
    int vetor[5], i;
    // Leitura dos elementos do vetor
    for(i=0; i<5; i++){
        cout << "\nDigite um numero inteiro: ";
        cin >> vetor[i];
    }
    // Escrita do vetor original
    cout << "O vetor original é: ";
    for(i=0; i<5; i++)
        cout << vetor[i] << " ";
    // Escrita do vetor em ordem inversa
    cout << "O vetor em ordem inversa é: ";
    for(i=4; i>=0; i--)
        cout << vetor[i] << " ";
}
```



## Exercícios

- 1) Fazer um programa C para ler 50 valores numéricos e armazená-los num vetor. Em seguida, encontre o maior valor do vetor.
- 2) Fazer um programa em C para ler 10 números inteiros positivos e armazená-los num vetor. Em seguida, calcule e mostre a média aritmética apenas dos números ímpares.
- 3) Faça um programa em C que leia 50 números inteiros positivos e armazene-os num vetor **A**. A partir dos dados do vetor **A**, gere dois outros vetores:
  - Vetor **B**: só números pares;
  - Vetor **C**: somente os números ímpares.

Escreva os dois vetores resultantes.



Disciplina APC I - Profa. Carla G. Pelissoni

17

Grupo **ibmec**  
METROCAMP **IBTA**

## Matrizes

- **Matrizes = Variáveis Compostas Bidimensionais Homogêneas**
- Conjunto de dados referenciado por um mesmo nome e que necessita de **dois índices** para ter seus elementos individualizados.
- MATRIZES:

$$\begin{array}{cccccc}
 \mathbf{A} = & a_{00} & a_{01} & a_{02} & \dots & a_{0n-1} \\
 & a_{10} & a_{11} & a_{12} & \dots & a_{1n-1} \\
 & a_{20} & a_{21} & a_{22} & \dots & a_{2n-1} \\
 & \dots & \dots & \dots & \dots & \dots
 \end{array}
 \Rightarrow
 \begin{array}{l}
 A_{m \times n} \\
 m = \text{linha} \\
 n = \text{coluna}
 \end{array}$$

APC I - Profa. Carla G. Pelissoni

Grupo **ibmec**  
METROCAMP **IBTA**

# Matrizes

- Usa-se sempre **2 índices**: um para percorrer as linhas e outro para as colunas (geralmente, são chamados de ***i*** e ***j***).
- **Exemplo**: matriz de números para cálculos matemáticos.

Matriz<sub>3x3</sub> =

199	35	57
29	10	6
30	0	27

Matriz[0][1] = 35

Matriz[1][2] = 6

Matriz[i][j] = 30, supondo i=2 e j=0

APC I - Profa. Carla G. Pelissoni

Grupo **ibmec**  
METROCAMP 19 **IBTA**

# Matrizes

- Se o número de linhas é igual a ***m***, o índice que percorre as linhas varia de **0 a *m*-1**.
- Se o número de colunas é igual a ***n***, o índice que percorre as colunas varia de **0 a *n*-1**.
- Por exemplo Mat<sub>3x4</sub>: o índice para linha varia de 0 a 2 e o índice para coluna de 0 a 3.

APC I - Profa. Carla G. Pelissoni

Grupo **ibmec**  
METROCAMP 20 **IBTA**

# Matrizes

- **Declaração:**

Tipo NomeMatriz[MAX\_LINHAS][MAX\_COLUNAS];

- **Exemplo:**

float Matriz[3][8];

primeira posição: Matriz[0][0];

última posição: Matriz[2][7];

APC I - Profa. Carla G. Pelissoni

# Matrizes

- **Armazenando** elementos em uma matriz 3x4.

```
//Percorre as linhas
for (i=0; i < 3; i++)
{
    //Percorre as colunas
    for (j=0; j < 4; j++)
    {
        cout << "\nDigite um valor: ";
        cin >> Matriz[i][j];
    }
}
```

APC I - Profa. Carla G. Pelissoni

# Matrizes

- **Mostrando na tela** os elementos de uma matriz 3x4.

```
cout << "\nOs elementos da matriz são: ";  
//Percorre as linhas  
for (i=0; i < 3; i++)  
{  
    //Percorre as colunas  
    for (j=0; j < 4; j++)  
    {  
        cout << Matriz[i][j] << " ";  
    }  
    cout << "\n";  
}
```

APC I - Profa. Carla G. Pelissoni

# Exemplo

- O programa a seguir lê um conjunto de números inteiros e armazena em uma matriz  $Mat_{3 \times 4}$ . O programa calcula a média aritmética dos elementos dessa matriz e escreve o resultado.

APC I - Profa. Carla G. Pelissoni

## Exemplo

```
#include <iostream.h>
main()
{
    int Matriz[3][4], i, j;
    float soma=0;
    cout << "Digite os valores da matriz:\n";
    for (i=0; i < 3; i++)
    {
        for (j=0; j < 4; j++)
        {
            cout << "\nLinha " << i << ", Coluna " << j << ": ";
            cin >> Matriz[i][j];
            soma = soma + Matriz[i][j];
        }
    }
    cout << "\nOs elementos digitados para a matriz foram:\n";
    for (i=0; i < 3; i++)
    {
        for (j=0; j < 4; j++)
        {
            cout << Matriz[i][j] << " ";
        }
        cout << "\n";
    }
    cout << "\nMedia dos elementos da matriz: " << soma/(3*4);
    system("pause");
}
```

APC I - Profa. Carla G. Pelissoni

## Strings

- **Strings** são vetores do tipo **char** que armazenam um texto formado de caracteres e sempre terminado pelo caractere '\0'.
- O caractere '\0' representa final de *string*.
- Cada caractere é um elemento independente do vetor e pode ser acessado por meio de um índice.

## Strings - Exemplo

Palavra

'O'	0
'L'	1
'A'	2
'\0'	3
	4

Declaração:

```
char nome[TAMANHO];
```

Exemplo:

```
char palavra[30];
ou
....
#define MAX 30
....
char palavra[MAX];
```

## Strings – Exemplo: Leitura e escrita

```
#include<iostream.h>
main()
{
    char Nome[30];

    //Lê a string inteira
    cout << "\nDigite o seu nome: ";
    cin >> Nome;

    //Escreve string inteira
    cout << "\nSeu nome inteiro eh " << Nome;

    //Escreve o caractere da primeira posição
    cout << "\nA primeira letra do seu nome é: " <<
    Nome[0];
}
```

## Strings

- Para manipular (copiar, comparar, etc.) cadeias de caracteres (*strings*) inteiras na Linguagem C é preciso utilizar funções específicas.
- Por exemplo:
  - strcpy
  - strcmp
  - strlen
  - Entre **muitas** outras ...

## Funções para Manipulação de *Strings*

- **strcpy (S1, S2):** copia o conteúdo de S2 em S1.

Exemplo:

```
#include<string.h>    //Para poder usar strcpy()
#include<iostream.h>
main()
{
    char Nome[10];
    //Atribui o nome "Ana" à variável Nome
    strcpy(Nome, "Ana");
    cout << Nome;
}
```

## Funções para Manipulação de *Strings*

- **strlen(S1):** função que retorna o tamanho de uma string.

**Exemplo:**

```
#include<iostream.h>
#include<string.h>
main()
{
    cout << "Tamanho do nome Maria: " <<
        strlen("Maria");
}
```

**Saída:** 5

**Obs.:** O caractere '\0' não é contado.

## Funções para Manipulação de *Strings*

- **strcmp(S1, S2):** compara duas strings e devolve um inteiro:

0	→	se S1 é igual a S2
Menor que 0	→	se S1 é menor que S2
Maior que 0	→	se S1 é maior que S2



## Funções para Manipulação de *Strings*

### Exemplo

```
#include<iostream.h>
#include<string.h>
main()
{
    char palavra[30];
    int aux;
    cout << "\nDigite uma palavra qualquer: ";
    cin >> palavra;
    aux = strcmp (palavra, "Teste");
    if (aux==0)
        cout << "O conteúdo da variável palavra é
        igual a Teste";
    else
        cout << "O conteúdo da variável palavra é
        diferente de Teste";
}
```

## Aquecimento ...

- Fazer um programa em linguagem C que leia uma string qualquer digitada pelo usuário e escreva na tela a quantidade de letras 'A' (maiúscula) e a quantidade de 'a' (minúscula) existentes nessa string.



## Exemplo de Solução

```
#include<iostream.h>
#include<string.h>
main() {
    char str[30];
    int cont_A=0, cont_a=0, i;
    cout << "\nDigite uma string: ";
    cin >> str;
    for(i=0; i < strlen(str); i++){
        if(str[i] == 'A')
            cont_A++;
        if(str[i] == 'a')
            cont_a++;
    }
    cout << "\nQuantidade de A: " << cont_A;
    cout << "\nQuantidade de a: " << cont_a;
}
```

35



## Exercícios de Revisão

- 1) Fazer um programa em C para ler 10 números inteiros positivos e armazená-los num vetor. Em seguida, calcule e mostre a média aritmética apenas dos números ímpares.
- 2) Faça um programa em C que leia 50 números inteiros positivos e armazene-os num vetor **A**. A partir dos dados do vetor **A**, gere dois outros vetores:
  - Vetor **B**: só números pares;
  - Vetor **C**: somente os números ímpares.

Escreva os dois vetores resultantes.



## Exercícios de Revisão

- 3) Faça um programa que preencha uma matriz 3x5 com valores numéricos, calcule e mostre a quantidade de elementos entre 15 e 20.
- 4) Fazer um programa em C que leia uma matriz 4x3 e imprima o maior valor e o menor valor da matriz.



## Exercícios de Revisão

- 5) Fazer um programa em Linguagem C que leia uma *string* qualquer e criptografe a string, substituindo todo caractere igual a 'a' pelo caracter 'x'. Como resultado, escrever a nova *string* e a quantidade de caracteres que foram substituídos.
- 6) Fazer um programa em C que leia uma string e gere o seu inverso. Por exemplo, se for introduzido o string ANTONIO, o programa deverá gerar uma outra string com o conteúdo OINOTNA. Exibir a string original e a string invertida.