

# Dados Estruturados

## *Vetores e Matrizes*

SCC121 - Introdução à Programação

São Carlos  
Abril de 2009

# Array - Definição

- *Vetor* ou *Array* é a forma mais familiar de dados estruturados.
- Um *array* é um conjunto de componentes do mesmo tipo.

# Array - Problema

Dada uma relação de 5 estudantes, imprimir o nome de cada estudante, cuja nota é maior do que a média da classe.



# Array - Solução



Imagine fazer um  
algoritmo deste tipo  
para 100 números!!!

1º. Algoritmo

## **Início**

*Leia(nome1,nota1,nome2,nota2,nome3,nota3,nome4,  
nota4,nome5,nota5)*

*media ← (nota1+nota2+nota3+nota4+nota5) / 5,0*

**Se** *nota1 > media* **então** escreva (*nome1*)

**Se** *nota2 > media* **então** escreva (*nome2*)

**Se** *nota3 > media* **então** escreva (*nome3*)

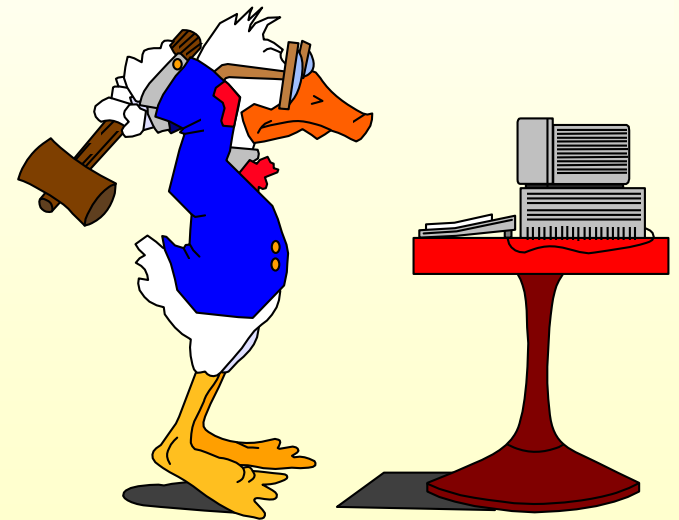
**Se** *nota4 > media* **então** escreva (*nome4*)

**Se** *nota5 > media* **então** escreva (*nome5*)

**Fim**

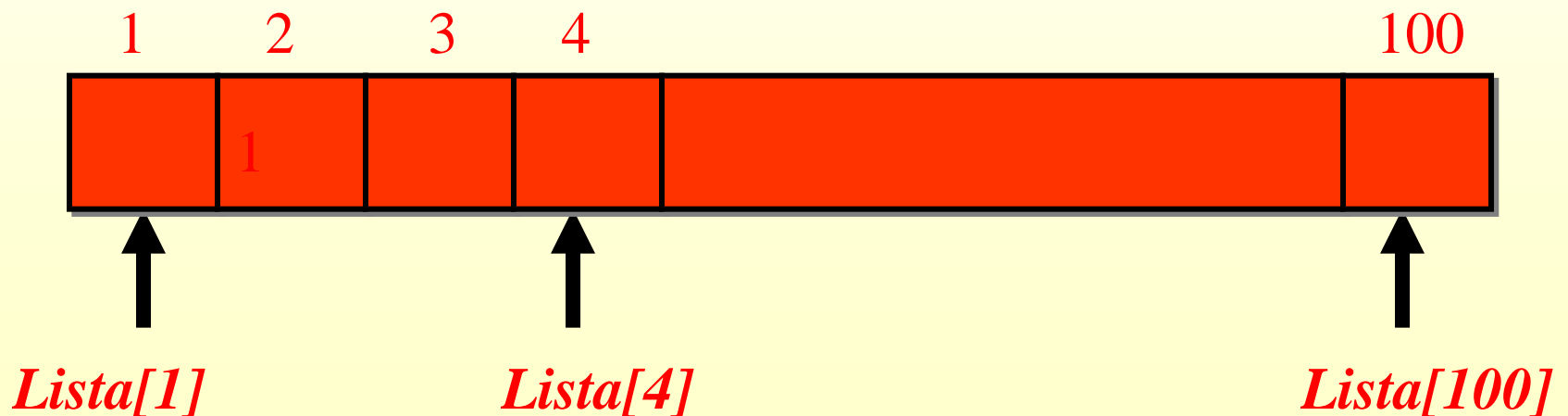
# Array - Solução 1

1. Uma variável para cada nome  $\rightarrow$  100 variáveis
2. Uma variável para cada nota  $\rightarrow$  + 100 variáveis
3. 100 testes



# Array - Definição

- Como estes dados têm uma relação entre si, podemos declará-los com um nome ÚNICO para todos os 100 elementos.
- Conjunto de 100 números = **LISTA**



# Array - Definição

- O elemento do vetor tem todas as características de uma variável e pode aparecer em expressões e atribuições.

*$Lista[2] \leftarrow Lista[3] + Lista[20]$*

- Para somar todos os elementos da Lista:

*$soma \leftarrow 0$*

*para  $I \leftarrow 1$  até 100 faça*

*$soma \leftarrow soma + Lista[i]$*

# Array - Características

- As características básicas são:
  - é uma estrutura homogênea, isto é, formada de elementos do mesmo tipo
  - todos os elementos da estrutura são igualmente acessíveis, isto é, o tempo e o tipo de procedimento para acessar qualquer um dos elementos do *Array* são iguais
  - cada elemento componente desta estrutura tem um nome próprio segundo sua posição no conjunto



# Array - Solução 2

## 2º. Algoritmo

### Início

**Para**  $i \leftarrow 1$  **até** 5 **faça**

    Leia(nome[i],nota[i])

soma  $\leftarrow$  0,0

**Para**  $i \leftarrow 1$  **até** 5 **faça**

    soma  $\leftarrow$  soma + nota[i]

media  $\leftarrow$  soma/5

**Para**  $i \leftarrow 1$  **até** 5 **faça**

**Se** nota[i] > media **então** escrever (nome[i])

### Fim



# Array - Declaração de Tipos

- arrays são agrupamentos de dados adjacentes na memória
- declaração:

`tipo_dado nome_array[<tamanho>];`

define um arranjo de *<tamanho>* elementos adjacentes na memória do tipo *tipo\_dado*

# Array - Problema 2



Para um vetor  $A$  com  $N$  números, formular um algoritmo que determine o maior e o menor elemento deste vetor. Imprimir o vetor, o maior e o menor elemento.

# Array - Solução

```
#include<stdlib.h>
#include<stdio.h>
#include <conio.h>
int A[10]; int i,Maior, Menor, N;
int main(){
    printf("Digite valor de N: ");
    scanf("%d",&N);
    for(i=0;i<N;i++){
        printf("%d numero:",i);
        scanf("%d",&A[i]);
    }
    Maior=A[0];
    Menor=A[0];
    for(i=1;i<N;i++){
        if (Maior < A[i])  Maior=A[i];
        if (Menor > A[i])  Menor=A[i];
    }
    printf("%d %d", Maior,Menor);
    getch( );
}
```

# Matrizes - Definição

- Também chamadas conjuntos bidimensionais, contém:
  - um número fixo de elementos;
  - todos são do mesmo tipo;
  - arranjos na forma de tabela de 2 dimensões;

# Matrizes - Definição

- Ex.: Uma matriz chamada **MAT** que tenha *m* elementos (horizontal) e *n* elementos (vertical)

	1	2	3	4	5	...	n
1							
2							
3				*			
4							
5							
...							
m							

An arrow points from the asterisk in the cell at row 3, column 4 to the text **Mat[3,4]**.

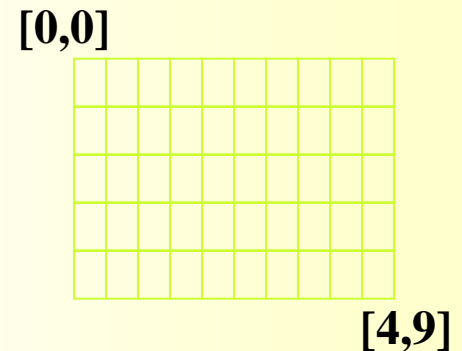
# Arrays Multidimensionais

- *Arrays* podem ter diversas dimensões, cada uma identificada por um par de colchetes na declaração

- Ex:

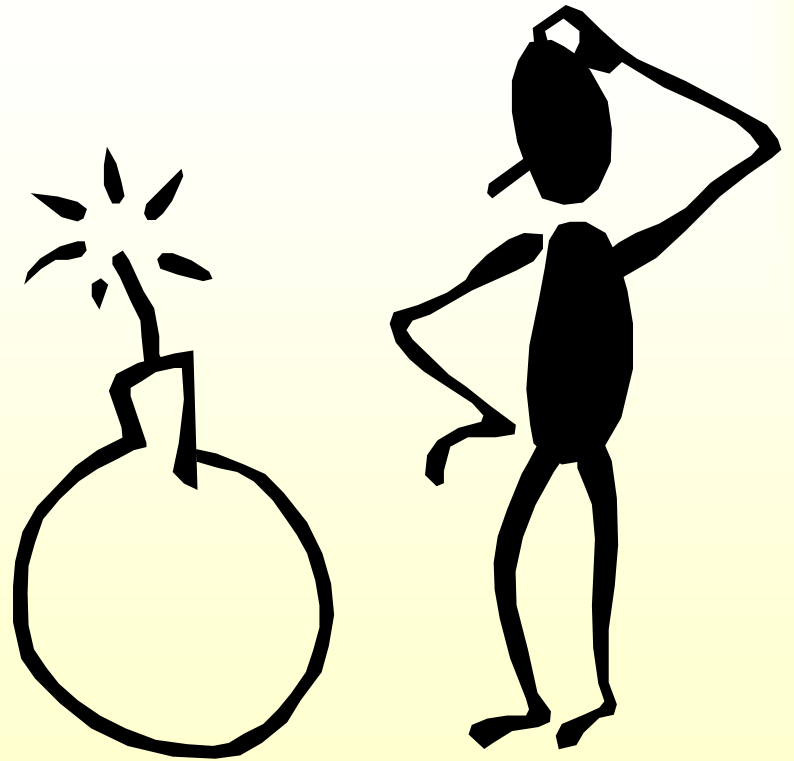
**char** matriz[5][10];

- declara uma matriz de 5 linhas e 10 colunas:
- na memória, entretanto, os caracteres são armazenados linearmente:



# Matrizes - Problema

Dada uma tabela de  
4x5 elementos, calcular  
a soma dos elementos e  
o maior elemento.





# Matrizes - Solução

```
Int main(){  
    int    A[4][5];  
    Int i, j, Maior, Soma;  
  
    {Leitura dos Dados}  
    for (i=0;i<4;i++)  
        for (j=0;j<5;j++)  
            scanf("%d",&A[i][j]);
```

*{continua no próximo slide...}*

*{continuação...}*

*{Inicialização de variáveis}*

Soma=0;

Maior=A[0][0];

*{Cálculo da Soma}*

**for** (i=0;i<4;i++)

**for** (j=0;j<5;j++){

        soma=soma + A[i][j];

**if** Maior < A[i][j]

            Maior:=A[i,j];

    }

*{Impressão dos Resultados}*

**printf**("%d %d",Soma,Maior);

}

*{Fim do programa}*

# Exercício 1

Dado um vetor VET, definido por:

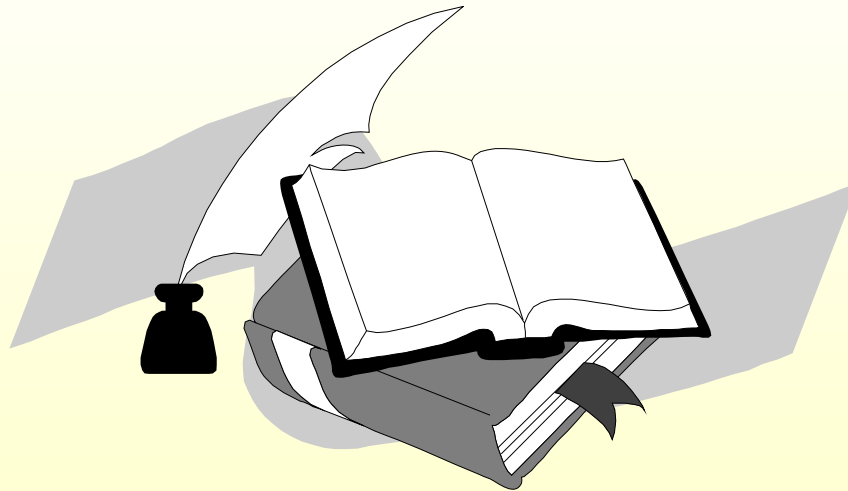
*tipo VET = vetor[1:100] - inteiros*

*v : VET*

- a) preenchê-lo com o valor inteiro 30;
- b) preenchê-lo com os números inteiros 1,2,3..100;
- c) preencher VET[i] com 1, se i é um quadrado perfeito, e com 0, nos demais casos.

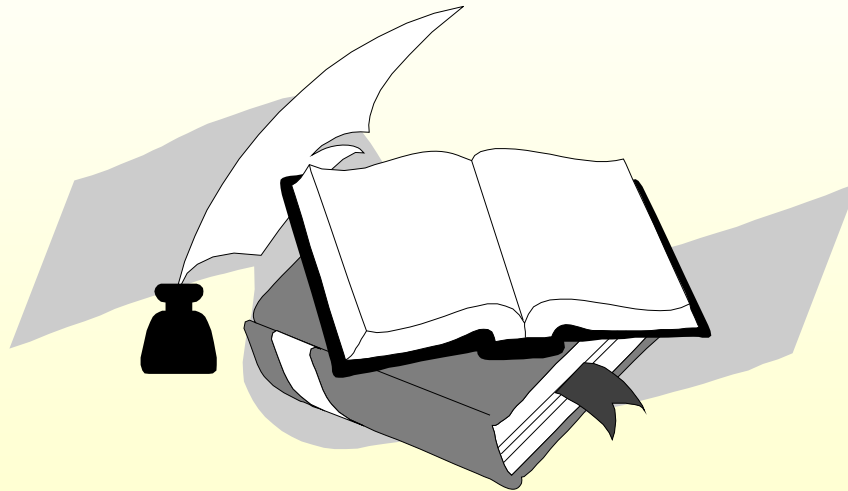
# Exercício 2

Fazer um programa em C para somar dois vetores de mesmo número de elementos.



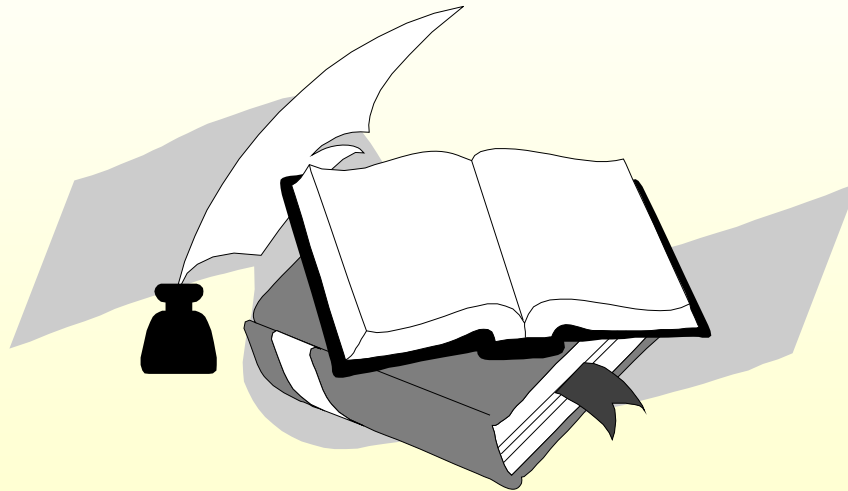
# Exercício 3

Fazer um programa para calcular a soma de duas matrizes reais de dimensão  $3 \times 5$ .



# Exercício 4

Fazer um programa para gerar a matriz transposta de uma matriz 3x3.



## Exercício 5

Dada uma matriz MAT de 4x5 elementos, fazer um programa para somar os elementos de cada linha gerando o vetor **SOMA**. Em seguida, somar os elementos do vetor **SOMA** na variável **TOTAL**, que deve ser impressa no final.

# **Dados Estruturados**

*Arrays: Vetores e Matrizes*

*Material Didático preparado por:  
profa. Roseli Romero*

**Fim**