

# MC102

## Algoritmos e Programação de Computadores

---

Aula de Laboratório 05  
Instituto de Computação  
Primeiro Semestre de 2012

---

2 de abril de 2012



# Conteúdo

1 Vetores

2 Strings

3 Matrizes



# Vetores

## Problema

Faça um programa que recebe 100 valores e encontre a média desses valores!



# Vetores

## Problema

Faça um programa que recebe 100 valores e encontre a média desses valores!

## Solução

```
1 int valor1, valor2, ..., valor100;  
2 printf("Digite o primeiro valor: ");  
3 scanf("%d",&valor1);  
4 printf("Digite o segundo valor: ");  
5 scanf("%d",&valor2);  
6 ...  
7 printf("Digite o centesimo valor: ");  
8 scanf("%d",&valor100);
```



# Vetores

- C fornece uma forma melhor para resolver esse problema;
- Usando **Vetor** (chamado também por **array**)
- Uma coleção de variáveis do mesmo tipo referenciada por um mesmo nome;
- Declaração:

*(Hebert Schildt)*

## Declaração

```
<tipo> identificador [<tamanho>;
```

## Exemplo

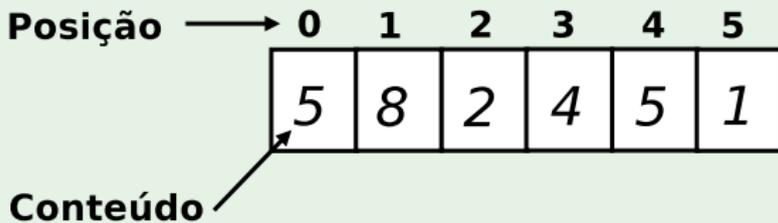
```
int valores[100];  
float notas[40];  
char nome[50];
```



# Vetores

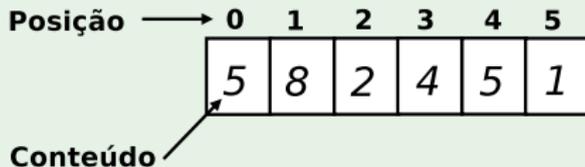
- Tamanho pré-definido;
- Acesso por meio de índice inteiro;
- O índice começa de **0** e vai até o **<tamanho> - 1**;
- Índice fora desse limite causa comportamento anômalo do código.

## Vetor



# Exemplo

## Exemplo



- Para acessar uma posição do vetor: **identificador** [**<indice>**]
- **vetor[0]** retornará valor **5**;
- **vetor[3]** retornará valor **4**;
- **vetor[5]** retornará valor **1**;
- **vetor[6]**, resultará em um comportamento estranho do código;



# Vetores

## Declaração

Para declarar um vetor é necessário incluir:  
`<tipo> <identificador> [<tamanho>].`



# Vetores

## Declaração

Para declarar um vetor é necessário incluir:

```
<tipo> <identificador> [<tamanho>].
```

## Exemplo

```
1 #include <stdio.h>
2
3 int main(){
4     float notas[50];
5     notas[0] = 4.5;
6     notas[1] = 5.5;
7     notas[2] = 9.3;
8     printf("%.2f\n", notas[0]);
9     printf("%.2f\n", notas[1]);
10    printf("%.2f\n", notas[2]);
11 }
```



# Vetores

## Problema

Faça um programa que receba 10 notas e retorne a média geral usando vetor.



# Vetores

## Problema

Faça um programa que receba 10 notas e retorne a média geral usando vetor.

## Solução

```
1 #include <stdio.h>
2
3 int main(){
4     float vetor[10], media, soma;
5     int i;
6     for(i = 0; i < 10; i++){
7         printf("Digite um valor: \n");
8         scanf("%f",&vetor[i]);
9         soma += vetor[i];
10    }
11    media = soma/10;
12    printf("Media igual a: %.2f\n",media);
13 }
```



# Vetores

## Problema

Faça um programa que receba 10 notas e retorne a média geral usando vetor.



# Vetores

## Problema

Faça um programa que receba 10 notas e retorne a média geral usando vetor.

## Solução

```
1 #include <stdio.h>
2
3 int main(){
4     float vetor[10], media, soma;
5     int i;
6     for(i = 0; i < 10; i++){
7         printf("Digite um valor: \n");
8         scanf("%f",&vetor[i]);
9         soma += vetor[i];
10    }
11    media = soma/10;
12    printf("Media igual a: %.2f\n",media);
13 }
```



# Strings

- C não possui o tipo **string**, mas podemos considerar um vetor de caracteres uma **string**;
- Em C uma string é sempre terminada pelo caracter especial **'/0'**;
- **Portanto, sempre declare a string com um caracter a mais para armazenar a sua string;**

## Exemplo de Declaração

```
char nome[30];  
char email[40];
```



# Strings

- Para strings não é necessário utilizar o **&** no comando **scanf**;

## Exemplo

```
1 #include <stdio.h>
2
3 int main(){
4     char nome[20];
5     char msgNome[] = "Digite o nome: ";
6     char msgIdade[] = "Digite a sua idade: ";
7     int idade;
8
9     printf("%s \n", msgNome);
10    scanf("%s", nome);
11    printf("%s \n", msgIdade);
12    scanf("%d",&idade);
13    printf("Meu nome e %s e tenho %d.\n", nome, idade);
14 }
```

# Strings

- Para strings não é necessário utilizar o **&** no comando **scanf**;
- O **scanf** lê até o primeiro espaço em branco, como alternativa pode-se utilizar **scanf("%[A-Z a-z 0-9]", nome);**

## Exemplo

```
1 #include <stdio.h>
2
3 int main(){
4     char nome[20];
5     char msgNome[] = "Digite o nome: ";
6     char msgIdade[] = "Digite a sua idade: ";
7     int idade;
8
9     printf("%s \n", msgNome);
10    scanf("%s", nome);
11    printf("%s \n", msgIdade);
12    scanf("%d",&idade);
13    printf("Meu nome e %s e tenho %d.\n", nome, idade);
14 }
```

# Comando fgets

- Uma alternativa para ler strings é usar o comando **fgets()**;

## Declaração

```
fgets(<identificador>, TAM, stdin)
```

## Exemplo

```
1 #include <stdio.h>
2
3 int main(){
4     char nome[30];
5     int idade;
6     printf("Digite o seu nome Completo: ");
7     fgets(nome, 30, stdin);
8     printf("Digite a sua idade: ");
9     scanf("%d",&idade);
10    printf("Meu nome e %s e tenho %d\n", nome, idade);
11 }
```

# Matrizes

- Uma Matriz é um vetor que possui duas ou mais dimensões;

## Declaração

```
<tipo> <identificador> [<linhas>][<colunas>]
```

```
<tipo> <identificador> [<dim1>][<dim2>] ... [<dimN>]
```

```
Ex.: int a [4][4];  
      int b [4][5][3];
```

| Coluna → | 0  | 1 | 2 | 3  |
|----------|----|---|---|----|
| Linha ↓  |    |   |   |    |
| 0        | 5  | 8 | 2 | 4  |
| 1        | 15 | 9 | 1 | 0  |
| 2        | 3  | 6 | 2 | 7  |
| 3        | 12 | 9 | 6 | 10 |



# Exemplo

## Exemplo

| Coluna → | 0  | 1 | 2 | 3  |
|----------|----|---|---|----|
| Linha ↓  | 5  | 8 | 2 | 4  |
| 1        | 15 | 9 | 1 | 0  |
| 2        | 3  | 6 | 2 | 7  |
| 3        | 12 | 9 | 6 | 10 |

- Para acessar uma posição da matriz (2D):  
**identificador [`<linha>`] [`<coluna>`]**
- **matriz[0,0]** retornará valor **5**;
- **matriz[3,1]** retornará valor **9**;
- **vetor[2,2]** retornará valor **2**;



# Matrizes

## Problema

Programa que declare uma matriz 10x10 e coloque 0 em todas as posições. A Matriz deve ser impressa na tela (em forma de matriz).



# Matrizes

## Problema

Programa que declare uma matriz 10x10 e coloque 0 em todas as posições. A Matriz deve ser impressa na tela (em forma de matriz).

## Solução

```
1 #include <stdio.h>
2 int main(){
3     int matriz [10][10], a, b;
4     for(a = 0; a < 10; a++){
5         for(b = 0; b < 10; b++){
6             matriz[a][b] = 0;
7         }
8     }
9     for(a = 0; a < 10; a++){
10        for(b = 0; b < 10; b++){
11            printf("%d ", matriz[a][b]);
12        }
13        printf("\n");
14    }
15 }
```



# Questões?

Obrigado!

*Para informação:*

**Página dos Laboratórios (Tarefas):** <http://susy.ic.unicamp.br:9999/mc102ab>

**Página do Curso:** <http://www.lrc.ic.unicamp.br/~geraldoms/mc102>

**E-mail:**

*geraldoms[at]lrc.ic.unicamp.br*

*brhenrique.fischer[at]gmail.com*

