



Seqüências de Caracteres

“Strings”

Material da Prof. Ana Eliza



Tipo de Dados STRING

- **Definição**

- Uma STRING é uma seqüência de caracteres.

- **Exemplos**

“Maria José”

“Número 10”

“5% de R\$ 20,00”



Tipo de Dados **STRING**

- **Definição** (cont.)

- Uma **STRING** pode ser composta por quaisquer combinações de caracteres sejam eles letras ('a',..., 'Z'), dígitos ('0',..., '9') ou símbolos ('&', '!', '+', ...).
- Uma **STRING** pode ter qualquer quantidade de caracteres, incluindo nenhum.
- Uma string sem caracteres é chamada de **STRING VAZIA**.



Tipo de Dados STRING

- **Implementação**

Na linguagem C, uma STRING é implementada como sendo um vetor (*array unidimensional*) de caracteres.

- **Declaração de Variáveis do tipo String**

char nomeDaVariável [tamanho];

- **Exemplos**

char nome [20];

char endereco [50];

char profissao [30];



Tipo de Dados **STRING**

- **Implementação** (cont.)

- Na linguagem C, uma **STRING** definida como sendo de tamanho N poderá armazenar, no máximo, $N-1$ caracteres.
- Internamente, as strings possuem um caractere adicional, o `'\0'` (caractere NULO), que delimita o final da string.

OBS: Não confunda o caractere NULO (de código ASCII zero) com o caractere zero (que possui código ASCII 48).

Tipo de Dados **STRING**

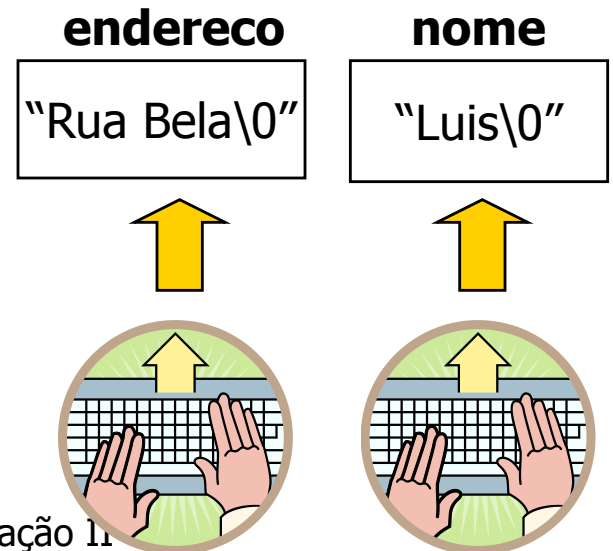
Operações com Strings

- Entrada de dados => **gets** (nomeVariavel)
- Armazena a seqüência digitada na variável.
- Coloca o '\0' no final da seqüência digitada.

Exemplos

gets (endereco);

gets (nome);



Tipo de Dados STRING

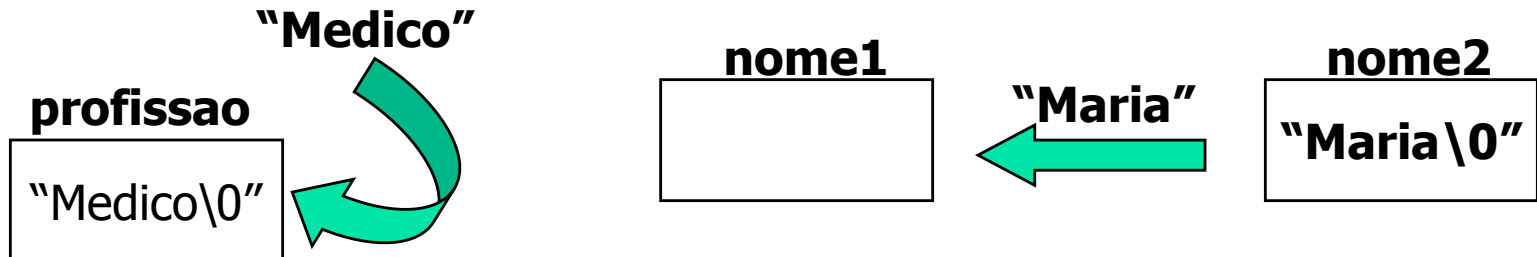
■ Operações com Strings

- Atribuição de valores => **strcpy** (*variavel, sequencia*)
- Armazena a **sequencia** na **variável**.
- Coloca o '\0' no final da seqüência.

■ Exemplos

strcpy (profissao, "Medico");

strcpy(nome1,nome2);



Tipo de Dados STRING

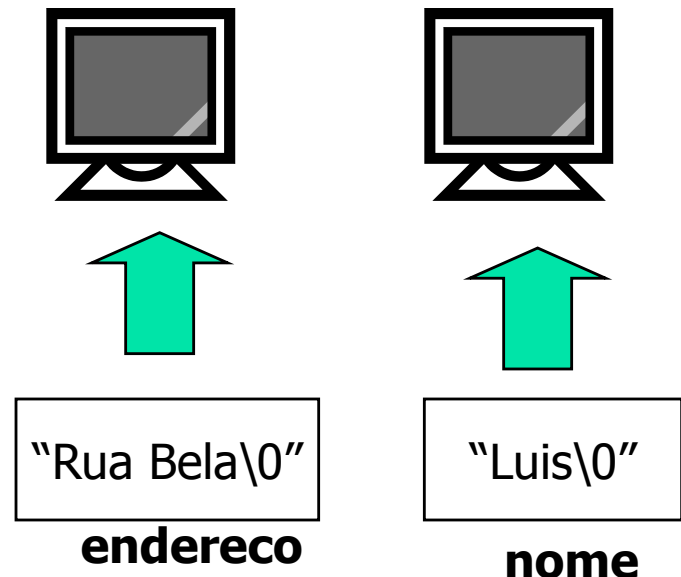
- **Operações com Strings**

- Saída de dados => **puts** (nomeVariavel)
- Exibe o conteúdo da variável.

- **Exemplos**

puts (endereco);

puts (nome);



Tipo de Dados STRING

- **Operações com Strings**

- Seleção de caracter \Rightarrow `variavel_string [posição]`
- Seleciona um caracter da string.

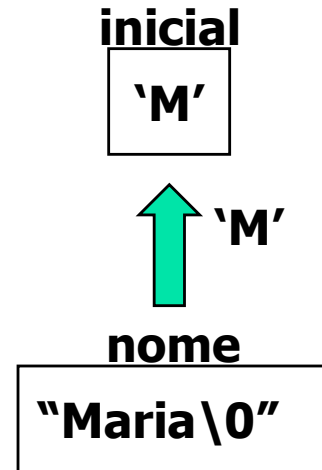
- **Exemplo**

char inicial;

char nome [10];

...

`inicial = nome[0];`





Tipo de Dados **STRING**

- **Função de Comparação de Strings**

strcmp (string1, string2)

- **Observação:**

- Retorna **0** (zero) se **string1** for igual a **string2**;
- Retorna **valor maior que zero** se **string1** for alfabeticamente maior que **string2**;
- Retorna **valor menor que zero** se **string1** for alfabeticamente menor que **string2**;

Tipo de Dados **STRING**

- **Função de Comparação de Strings**

- **Exemplo 1**

```
if (strcmp (profissao, "professor") == 0)
    printf ("É professor \n");
else
    printf ("Não é professor \n");
```

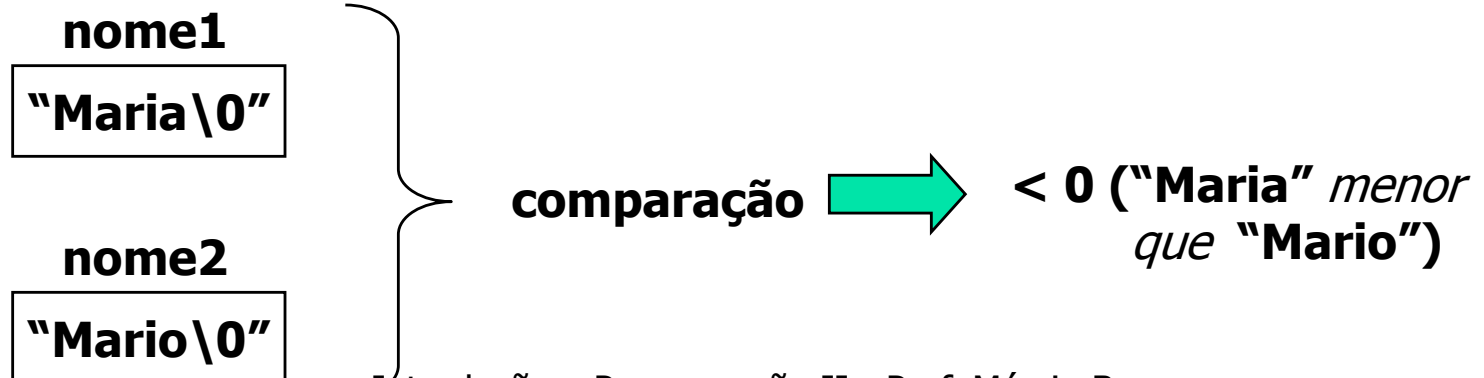


Tipo de Dados STRING

- **Função de Comparação de Strings**

- **Exemplo 2**

```
if (strcmp (nome1,nome2) < 0)
    printf ("%s vem primeiro que %s",nome1,nome2);
else
    printf ("%s vem primeiro que %s",nome2,nome1);
```



Tipo de Dados STRING

- **Operações com Strings**

- Quantidade de posições válidas ocupadas na string.
strlen (variável)
- Retorna o número de caracteres armazenados na string antes do primeiro '\0' encontrado.

- **Exemplo**

```
char nome [10];
```

```
int tam;
```

```
...
```

```
strcpy (nome, "Maria");
```

```
tam = strlen (nome);
```

```
printf("%s possui %i letras", nome, tam);
```

nome	tam
"Maria\0"	5

Tipo de Dados STRING

- Operações com Strings

- Concatenação de strings.

strcat (string1,string2)

- Concatena **string2** no final de **string1**.

- Exemplo

```
char nome [20];
```

```
char sobrenome[10];
```

```
...
```

```
strcat (nome,sobrenome);
```

nome

"Maria\0"

sobrenome

"Silva\0"

nome

"MariaSilva\0"