INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Algoritmos e Programação II

Strings em C

Profa. Dra. Eloize Seno

INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Strings

- String (ou cadeia de caracteres) em C é um **vetor de caracteres**.
  - Na linguagem C **não existe** o tipo de dados string!

- Exemplo: criação de um vetor para armazenar um único nome:

```
char nome[40]; /* reserva um espaço
  para armazenar até 40 caracteres. */
```

INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Strings (cont.)

- **Terminador de strings**: toda string termina com o caracter null ('\0'), que é automaticamente inserido pelo compilador.
- Deve-se declarar sempre o vetor com uma posição a mais para armazenar o caractere nulo.
- Por exemplo, para armazenar a palavra **CADEIA** deve-se declarar um vetor do tipo char com sete posições: `char string[7];`

INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Strings (cont.)

- Inicialização de strings no momento da declaração:

```
char nome[]={'S','t','r','i','n','g','\0'};
char nome[] = "String";
```

- Obs: Para armazenar em uma cadeia aspas, apóstrofos ou barra invertida, é preciso usar barra invertida antes.
  - Ex: `char string[ ] = "caixa d\' agua";`

INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Strings (cont.)

- **Leitura** e **impressão**: pode-se informar apenas o identificador da variável do tipo vetor de caracteres, sem informar o índice do vetor.
  - Ex:

```
char nome[30];
printf("Digite seu nome:\n");
scanf("%s", &nome);
printf("Olá %s!", nome);
```

INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Lendo e imprimindo strings com scanf e printf

```
#include  <stdio.h>
#include<stdlib.h>
#define tam 40 // define o tamanho máximo do
  vetor nome
int main ( )
{  char nome[tam];

    /* Entrada de dados do vetor */
    printf("Por favor, qual o seu nome?");
    scanf(“%s”, &nome);
    printf("Eu sou um computador PC, em que
      posso ajuda-lo %s?\n",nome);
    system(“Pause”);
}
```

INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Lendo e imprimindo strings com scanf e printf (cont.)

- Atenção: Caso o nome digitado seja Antonio da Silva, o programa imprimirá somente:
  - Eu sou um computador PC, em que posso ajuda-lo Antonio.

- O comando **scanf** não lê o nome todo e encerra a leitura quando encontra um caractere em branco.

INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Lendo e imprimindo strings com gets e puts

- **gets**: lê toda a string até que a tecla ENTER seja digitada (inclui espaços em branco).
- Exemplo:

```
gets(nome);
```

- **puts:** imprime toda a string até encontrar o o caracter null ('\0');
- Exemplos:

```
puts("O nome eh:\n");
puts(nome);
```

Obs: para usar os comandos **gets** e **puts**, use a biblioteca **stdio.h**

INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Lendo e imprimindo strings com gets e puts (cont.)

```
#include  <stdio.h>
#include <stdlib.h>
#define tam 40 // define o tamanho máximo do
  vetor nome
int main ( )
{    char nome[tam];
     /* Entrada de dados do vetor */
     puts("Por favor, qual o seu nome?");
     gets(nome);
     puts("Eu sou um computador PC, em que posso
       ajuda-lo?\n");
     puts(nome);
     system(“Pause”);
}
```

INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Lendo e imprimindo strings com gets e puts (cont.)

- Atenção: Caso o nome digitado seja Antonio da Silva, o programa imprimirá:
  - Eu sou um computador PC, em que posso ajuda-lo Antonio da Silva.

INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Vetor de Strings

- Um **vetor de strings** em C é uma **matriz bidimensional.**
  - 1ª dimensão: número máximo de strings a serem armazenadas;
  - 2ª dimensão: número máximo de caracteres a serem armazenados em cada string.

- Cada linha da matriz será tratada como um vetor de caracteres.

INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Vetor de Strings (cont.)

- Exemplo:
  - `char nomes[5][40]; /* reserva um espaço para armazenar até 5 nomes contendo até 40 caracteres cada.*/`

	0	1	2	...								39
0	J	O	Ã	O	\0							
1	M	A	R	I	A	N	A	\0				
2	C	A	R	L	O	S	\0					
3	A	N	A		M	A	R	I	A	\0		
4	J	O	S	É	\0							

INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Vetor de Strings

- **Leitura** e **Impressão**: é preciso informar apenas o identificador da variável do tipo vetor de caracteres e o índice da linha na matriz que contém ou conterá a string.

INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Programa exemplo: cria e imprime o vetor de nomes

```
#include<stdlib.h>
#include<stdio.h>
int main()
{
      int i;
      char nomes[5][40];

      for (i=0; i<5; i++)
      {
          printf("Entre com o nome da linha
            %d", i);
          gets(nomes[i]);
      }
```

continua

INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Programa exemplo: cria e imprime o vetor de nomes

```
for (i=0; i<5; i++)
      {
          printf("\nO nome %d e\n", i);
          puts(nomes[i]);
      }
      system(“Pause”);
}
```

INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Matriz de Strings

- Uma **matriz de strings** em C é uma **matriz tridimensional.**
  - 1ª dimensão: número de linhas da matriz;
  - 2ª dimensão: número de colunas da matriz;
  - 3ª dimensão: tamanho de cada string (número máximo de caracteres permitido).

INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Matriz de Strings (cont.)

- Exemplo:
  - char nomes[5][3][40]; /* reserva um espaço para armazenar até 15 nomes contendo até 40 caracteres cada.*/

INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Matriz de Strings (cont.)

- **Leitura** e **Impressão**: é preciso informar o identificador da variável do tipo matriz de strings, o índice da linha e o índice da coluna na matriz que contém ou conterá a string.

INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Programa exemplo: cria e imprime a matriz de nomes

```
#include<stdlib.h>
#include<stdio.h>
int main()
{
      int i, j;
      char nomes[5][3][40];

      for (i=0; i<5; i++)
      {    for (j=0; j<3; j++)
           {  printf("Entre com o nome da linha
                    %d coluna %d", i, j);
              gets(nomes[i][j]);
           }
      }
```

continua →

INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Manipulando Cadeias de Caracteres

- Biblioteca **string.h:** oferece um conjunto de funções para manipulação de strings.

- Funções para atribuição (ou cópia) de valores, concatenação e comparação de strings, entre outras.

INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Atribuição (ou cópia) de Cadeias de Caracteres

- Função: **strcpy**

  strcpy(str1,str2); // copia a cadeia str2 para str1

  Ex:
```
char str1[10], str2[10];
strcpy(str1,"Programa");
strcpy(str2,str1);
```

- Função: **strncpy**

  strncpy(str1,str2,n); /* copia os n primeiros caracteres da cadeia str2 para str1 */

  Ex:
```
strncpy(str1,str2,3); /* copia os 3 primeiros caracteres de str2 para str1 */
```

INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Concatenação de Cadeias de Caracteres

- Função: **strcat**

  strcat(str1,str2); /* concatena a cadeia str2 com a cadeia str1 */

- Função: **strncat**

  strncat(str1,str2,n); /* concatena os n primeiros caracteres da cadeia str2 com a cadeia str1 */

- Exemplo:

```
char str1[ ] = "eu adoro";
char str2[ ] = "programar";
strcat(str1,str2); /* str1 contém a
string Eu adoro programar */
```

INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Comparação de Cadeias de Caracteres

- Função: **strcmp**

  resultado = strcmp(str1,str2);

- A função **strcmp** compara duas strings e retorna:
  - 0, se as *strings* forem iguais,
  - um nº menor que 0, caso str1 seja alfabeticamente menor que str2;
  - um nº maior que 0, se str1 for alfabeticamente maior que str2;

INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Comparação de Cadeias de Caracteres (cont.)

- Atenção: a função **strcmp** considera letras maiúsculas símbolos diferentes de letras minúsculas.

- As funções **stricmp** e **strcmpi** consideram letras maiúsculas e minúsculas iguais:

```
resultado = stricmp(str1,str2);
resultado = strcmpi(str1,str2);
```

INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Outras Funções Úteis

- Descobrindo o número de caracteres da cadeia (comprimento):

  resultado = **strlen(str1);** // retorna o tamanho de str1

- Verificando a posição de um **caractere** em uma cadeia:

  posicao = **strchr(str1, caractere);** /* verifica se **caractere** pertence a **str1** e retorna a posição em que ele se encontra na cadeia; Se o caractere não for encontrado na cadeia, o valor de posicao será **NULL** */

INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Outras Funções Úteis (cont.)

- Verificando a posição de início de uma **cadeia** dentro de outra **cadeia**:

  posicao = **strstr(str1, str2);** /* verifica se **str2** pertence a **str1** e retorna a posição em que ela se encontra na cadeia; Se str2 não fizer parte da cadeia str1, o valor de posicao será **NULL** */

INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Outras Funções Úteis (cont.)

- Conversão de maiúsculas em minúsculas:

  strlwr(str1);

- Conversão de minúsculas em maiúsculas:

  strupr(str1);

- Revertendo uma string:

  strrev(str1); // inverte a string armazenada em str1

INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Outras Funções Úteis (cont.)

- Conversão de cadeia de caracteres em valor numérico inteiro:

  numero = atoi(cadeia);

  Atenção: a variável numero deve ser do tipo **int.**

- Conversão de cadeia de caracteres em valor numérico real:

  numero = strtod(cadeia);

  Atenção: a variável numero deve ser do tipo **float.**

Obs: as funções **atoi** e **strtod** exigem a inclusão da biblioteca **stdlib.h**

INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Exercícios

1) Faça um programa que leia 10 nomes, ordene-os de forma crescente e mostre-os na tela.
2) Altere o programa 1) para imprimir todos os nomes que contenham uma letra qualquer fornecida pelo usuário.
3) Faça um programa que receba uma palavra e verifique se ela é um palíndromo, ou seja, se escrita do fim para o começo é igual à palavra escrita do começo para o fim. Exemplo: RENNER, ANA, MIRIM, OVO, etc.
4) Faça um programa que leia um nome e imprima as 4 primeiras letras.

INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Exercícios

5) Faça um programa que leia um nome e escreva o numero de letras que ele possui.

6) Faça um programa que leia o nome e o sobrenome de uma pessoa separadamente. O programa deve juntar as duas strings em uma só e escrever na tela: a nova string, o seu número de caracteres, a sua primeira e a última letra.

7) Faça um programa que leia um nome completo e o imprima de duas formas: com todas as letras em minúsculas e com todas as letras em maiúsculas.

INSTITUTO FEDERAL
São Paulo
Câmpus São Carlos

Programa exemplo: cria e imprime a matriz de nomes

```
        for (i=0; i<5; i++)
        {   for (j=0; j<3; j++)
            {  printf("\nNome da linha %d
                     coluna %d =", i, j);
               puts(nomes[i][j]);
            }
        }
        system("Pause");
}
```