

Cadeias de Caracteres ou Strings

Thierson Couto Rosa
thierson@inf.ufg.br

Instituto de Informática - INF/UFG

Cadeias de Caracteres ou Strings

- ▶ Devido à necessidade recorrente de processar textos, as linguagens de programação possuem mecanismos para armazenar sequências de caracteres.
- ▶ Muitas linguagens de programação possuem o tipo “cadeia” de caracteres ou “String” (Ex. C++, Java).
- ▶ O tipo string não existe na linguagem C. Uma cadeia de caracteres é implementada com um vetor do tipo **char**, em que o último caractere da sequência é o caractere ‘\0’.
- ▶ Sendo assim, é necessário que o tamanho do vetor de caracteres declarado seja suficiente para incluir o caractere ‘\0’ (também denominado de delimitador da cadeia).

Constantes do Tipo Cadeias de Caracteres

- ▶ Embora a linguagem C não tenha o tipo cadeias de caracteres (ele é implementado por um vetor de char), a linguagem permite o uso de constantes. Elas são representadas envolvendo-se um texto com um par de aspas. Ex: "Introducao aos Sistemas de Informacao".
- ▶ Não confundir com uma constante do tipo **char**. Ex 'A', que usa apóstrofes como delimitadores.
- ▶ Não é necessário acrescentar o delimitador de cadeia de caracteres em uma constante string - - o compilador faz isso automaticamente.

Constantes do Tipo Cadeias de Caracteres

Inicialização de Cadeias

- ▶ A inicialização de uma cadeia uma cadeia de caracteres pode ser feita com no exemplo: **char** vet[30]="Ola Brasil !";
- ▶ Mas não é permitido atribuir uma constante cadeia de caracteres a um vetor fora da declaração da variável, ou seja, vet[30]="Bom Dia!!!" ; causa erro de compilação. Para obter o efeito da atribuição deve-se copiar caractere por caractere da constante para o vetor. Isso pode ser feito escrevendo um código em C ou usando a função de copia (strcpy () ;)

Funções de Manipulação de Strings

- ▶ A linguagem C disponibiliza um grande conjunto de funções que permitem realizar diversas operações com strings.
- ▶ Para utilizar essas funções no programa é necessário incluir o arquivo de cabeçalho `string .h`.
- ▶ Essas funções recebem como parâmetros um ou mais ponteiros para caracteres e processam as sequências de caracteres recebidas até encontrarem em cada uma um `'\0'`.

Funções de Manipulação de Strings

As principais

<code>strcpy (char *s1, char* s2)</code>	copia a string s2 na área de memória apontada por s1
<code>strcat (char *s1, char* s2)</code>	concatena a string s2 ao final da string s1
<code>strlen (char *s1)</code>	retorna o número de caracteres de s1 não incluindo '\0'
<code>strcmp(char *s1, char*s2)</code>	retorna 0 se as duas strings forem iguais, negativo se $s1 < s2$ e positivo, caso contrário

Leitura de uma Cadeia de caracteres

- ▶ A leitura de uma string com a função `scanf ("%s ", &cadeia)` lê até encontrar um espaço em branco.
- ▶ Uma cadeia de caracteres que termina com `\n`, pode ser lida com: `scanf ("%[^\\n]", str)`
- ▶ A expressão entre colchetes indica à função `scanf` para ler qualquer caractere, exceto o delimitador de cadeia.
- ▶ O caractere `'\\n'` (resultante do uso da tecla ENTER) não é lido, mas fica no buffer. Recomenda-se que seja consumido. Por exemplo, usando-se `getchar ()`, em seguida.

Exemplo1:

Considere uma entrada formada por um número de casos de testes, onde cada caso contém um inteiro, um espaço e uma string que ocupa o restante da linha. O código a seguir lê os dados de cada linha e os imprime.

```
1 #include <stdio.h>
2 int main() {
3     int quant,num;
4     char cadeia[1000];
5     scanf("%d", &quant);
6     getchar();
7     while(quant--){
8         scanf("%d", &num);
9         getchar(); // senao o espaco e armazenado em cadeia
10        scanf("%[^\\n]",cadeia);
11        getchar();
12        printf("%d %s\\n", num, cadeia);
13    }
14 }
```


Exemplo de Entrada

4
23 Rua Tamandare Bonaparte
32 Avenida Santos Dumondt
12 Praça Boaventura da Silva
30 Oi