Strings

Linguagem C

Autora: Gabrielle Ribeiro

O que são Strings?

- Strings são sequências de caracteres;
- Em C, as strings não são um tipo nativo, sendo assim representadas por vetores de caracteres, delimitados pelo caractere nulo ‘\0’.

- Geralmente, as strings são usadas para representar textos

O caractere nulo '\0'

- Indica o fim de uma string
- Toda string deve ter o '\0' como último caractere
- É acrescentado automaticamente na string

Ao declarar uma string precisamos reservar um espaço a mais para armazenar o caractere nulo!!!

EXEMPLO

- Se queremos armazenar um texto de 10 caracteres em uma string, devemos declarar essa string com 11 espaços: 10 espaços para os caracteres do texto + 1 para o caractere nulo ‘\0’

char texto[11] = “estruturas”;

texto =

‘e’	**‘s’**	**‘t’**	**‘r’**	**‘u’**	**‘t’**	**‘u’**	**‘r’**	**‘a’**	**‘s’**	**‘\0’**

Representação de Strings

- As strings são representadas entre aspas duplas
- Exemplo:

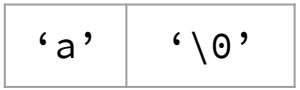
```
string1 = “Linguagem C”;
```

Representação - String x caractere

Strings

- Representadas entre aspas duplas

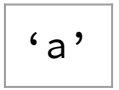
  var = “a”;

  var = 

Caracteres

- Representados entre aspas simples

  var = ‘a’;

  var = 

Declaração de Strings

- Sintaxe:

```
// Forma padrão
char nome_da_string[tamanho];
// Outras formas
char nome_da_string[tamanho] = “Algum texto qualquer”;
char nome_da_string[tamanho] = {‘t’, ‘e’, ‘x’, ‘t’, ‘o’};
char nome_da_string[] = “Outro texto qualquer”;
```

Exemplos

- Somente declaração:

  char string01[10];

- Declaração com inicialização da string:

  char string02[21] = “Algum texto qualquer”;

  char string03[21] = {‘t’, ‘e’, ‘x’, ‘t’, ‘o’};

Exemplos

- Declaração sem definir o tamanho da string.

  char string04[] = “Outro texto qualquer”;

  Neste caso, o tamanho da string é calculado automaticamente, de modo a ter a dimensão exata para armazenar o texto atribuído a ela, incluindo também o caractere nulo.

  No exemplo acima, a string terá 21 espaços reservados.

Exemplos

- Declaração com atribuição de uma string nula (sem conteúdo)

  char string05[10] = “”;

  char string06[10] = {};

- Observação: Essa atribuição de um texto sem conteúdo só funciona no momento da declaração da string

Leitura de Strings

Com o especificador de tipo %s

scanf(“%s”, text);

- Lê a string até achar o primeiro espaço ou quebra de linha;

Com o delimitador de leitura

scanf(“%[^\n]”, text);

- Lê a string até a quebra de linha
- Ideal para leitura de strings que contenham espaços no seu conteúdo

ATENÇÃO!!

NA LEITURA DE STRINGS NÃO USAMOS O &

Printando uma string

- Para printar uma string na tela usamos o especificador de tipo %s

  char string_exemplo[ ] = “Bom dia!”;

  printf(“%s\n”, string_exemplo);

  A saída será:

  Bom dia!

Percorrendo uma string

- Uma forma prática de percorrer uma string usando um laço for de repetição é definir como condição de parada encontrar o caractere nulo ‘\0’ da string

```
for(i = 0; text[i] != ‘\0’; i++)
{
    …
}
```

Atenção!

operações de concatenação (soma) e atribuição NÃO são possíveis em strings Na linguagem C.

Como resolver esse problema?

COM A BIBLIOTECA STRING.H !!

Biblioteca String.h

- Nela existem várias funções para auxiliar a manipulação de strings
- Inclusão da biblioteca:

```
#include <string.h>
```

Biblioteca string.h

- Suas principais funções são:
  - strlen
  - srtcpy
  - strcmp
  - strcat

FUNÇÃO STRLEN

- Retorna a quantidade de caracteres de uma string, excluindo o caractere ‘\0’
- Cabeçalho da função:

```
int strlen(char * string)
```

Exemplo de uso - strlen

```
char text[25] = “Casa amarela”;

int tam = strlen(text);

printf(“%d\n”, tam);
```

Será printado na tela o valor:

12

Função strcpy

- Copia o conteúdo de uma string para outra, inclusive o caractere ‘\0’
- Cabeçalho da função:

```
int strcpy(char *destino, char *origem);
```

Exemplo de uso - strcpy

```
char origem[10] = “Apple”;

char destino[10];

strcpy(destino, origem);

printf(“%s\n”, destino);
```

Será printado na tela:

Apple

Função strcmp

- Compara o conteúdo de duas strings e retorna:
  - Um valor igual a 0 caso as duas strings sejam iguais;
  - Um valor maior que 0 caso a string1 seja maior que a string2 lexicograficamente;
  - Um valor menor que 0 caso a string1 seja menor que a string2 lexicograficamente.

- Cabeçalho da função:

```
int strcmp(char * string1, char * string2)
```

Exemplo de uso - strcmp

```
char string1[10] = “oi”;

char string2[10] = “oi”;

if(strcpm(string1, string2) == 0)

    printf(“São iguais\n”); // Saída que será printada

else

    printf(“Não são iguais\n”);
```

Função strcat

- Escreve a string2 ao final da string1 e retorna o endereço de memória da string1
- Cabeçalho da função:

```
char * strcat(char * string1, char * string2)
```

Exemplo de uso - strcat

```
char string1[30] = “oi,”;

char string2[30] = “ tudo bem?”;

strcat(string1, string2);

printf(“%s\n”, string1); // string1 = “oi, tudo bem?”
```

A saída será

oi, tudo bem?

Outras funções

- A biblioteca string.h possui outras funções. Para consultá-las basta pesquisar em algum site de referência da linguagem ou digitar no terminal:

man string

Algumas dicas

- Caso você esteja tendo problemas com o buffer durante a leitura, acrescente um espaço antes do especificador de tipo ou delimitador de leitura

  scanf(“ %s”, text);

  scanf(“ %[^\n]”, text);

- Para limpar uma string copie para ela uma string vazia

  strcpy(text, “”);

Algumas dicas

- Caso esteja em dúvida sobre o que uma biblioteca da linguagem C contém, ou sobre o que uma função de C faz, basta digitar no terminal:

man nome_da_biblioteca

man nome_da_função

Tabela ascii

Tabela Ascii

- Código binário que codifica 128 sinais
- Temos nela a representação de cada caractere em decimal, graças a isso podemos fazer manipulações com caracteres somando ou subtraindo valores inteiros de um caractere.

Exemplo

```
char var = ‘a’;

var += 3; // Agora var armazena o caractere ‘d’

printf(“%c”, var);
```

Tabela ASCII

- Isso é muito útil na resolução de problemas de strings, como problemas envolvendo a Cifra de César.
- Para consultar a tabela ASCII basta digitar no terminal:

man ascii

Algumas sugestões de exercícios - URI Online Judge

- URI 1234 - Sentença Dançante
- URI 1235 - De dentro pra fora
- URI 1238 - Combinador
- URI 1239 - Atalhos Bloggo
- URI 1332 - Um-Dois-Três
- URI 1551 - Frase Completa
- URI 1581 - Conversa Internacional

Algumas sugestões de exercícios

1. Implementar suas próprias funções para:
   a. my_strlen
   b. my_strcpy
   c. my_strcmp
   d. my_strcat

- Observação: A implementação dessas funções tem somente fim didático. Caso precise utilizar essas funções na resolução de seus problemas, use as funções prontas da linguagem.

  NÃO REINVENTE A RODA!!!