Strings

Todas as funções apresentadas neste documento estão no arquivo de cabeçalho string.h.

1- Determinando o tamanho de uma string

Para determinar o tamanho de uma string use a função **strlen**( ). Sua sintaxe é: strlen(string)

Exemplo:

```
/* Determinando o tamanho de uma string usando a função strlen() */

#include <stdio.h>
#include <string.h>

int main(void)
{
 char string[20];

 printf("\n");
 printf("Determinando o tamanho de uma string\n");
 printf("-----------------------------------\n");
 printf("\n");
 printf("Digite a string :");
 scanf("%s",string);
 printf("\n");
 printf("A string tem %d carateres.\n\n",strlen(string));
 return(0);
}
```

2 - Copiando uma string em outra

Para copiar uma string em outra use a função strcpy( ). Sua sintaxe é: strcpy(destino, origem)

Exemplo:

```
/* Copiando uma string em outra usando a função strcpy() */

#include <stdio.h>
#include <string.h>

int main(void)
{
   char string1[10], string2[10];

   printf("\n");
   printf("Copiando uma string em outra\n");
   printf("----------------------------\n");
```

```
   printf("\n");
   printf("Digite string1 :");
   scanf("%s",string1);
   printf("\n");
   printf("string1 = %s\n",string1);
   printf("string2 = %s\n",strcpy(string2,string1));
   return(0);
 }
```

Na prática, todo conteúdo de string2 é substituído por string1.

3 - Unindo duas strings

Para unir duas strings use a função strcat( ). Sua sintaxe é: strcat(destino, origem)

Exemplo:

```
/* Unindo duas strings usando a função strcat() */

#include <stdio.h>
#include <string.h>

int main(void)
{
   char string1[100], string2[10];

   printf("\n");
   printf("Unindo duas strings\n");
   printf("-------------------\n");
   printf("\n");
   printf("Digite string1 :");
   scanf("%s",string1);
   printf("\n");
   printf("Digite string2 :");
   scanf("\n%s",string2);
   printf("\n");
   printf("Unindo string1 a string2 : %s\n\n",strcat(string2,string1));
   return(0);
 }
```

4 - Anexando caracteres de uma string em outra

Para anexar caracteres de uma string em outra use a função strncat( ). Sua sintaxe é :

strncat(destino, origem, nr_caracteres)

Exemplo:

```
/* Anexando caracteres de uma string em outra usando a função strncat()*/

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

```
int main(void)
{
   char string1[20],string2[6]="aeiou";
   printf("\n");
   printf("Anexando caracteres de uma string em outra\n");
   printf("-----------------------------------------\n");
   printf("Entre com string1 :");
   scanf("%s",string1);
   printf("\n");
   printf("string2 = %s\n\n",string2);
   printf("string1 + 3 caracteres de string 2 = %s\n",strncat(string1,string2,3));
   printf("\n");
   return(0);
 }
```

5 - Comparando duas strings

Para comparar duas strings use a função strcmp( ). Sua sintaxe é: strcmp(string1,string2)

Se as strings forem iguais a função retorna zero, se string1 for maior a função retorna um valor menor que zero e se string2 for maior a função retorna um valor maior que zero.

Exemplo:

```
/* Comparando duas strings com a função strcmp() */

#include <stdio.h>
#include <string.h>

int main(void)
{
   char string1[20],string2[20];
   int retorno;

   printf("\n");
   printf("Comparando duas strings\n");
   printf("-----------------------\n");
   printf("\n");
   printf("Entre com a primeira string :");
   scanf("%s",string1);
   printf("\n");
   printf("Entre com a segunda string :");
   scanf("\n%s",string2);
   printf("\n");

   retorno = strcmp(string1,string2);

   if(retorno == 0)
     printf("As strings são iguais.\n");
   else if(retorno < 0)
     printf("A string1 , maior.\n");
```

```
   else
     printf("A string2 , maior.\n");

   return(0);
}
```

OBSERVAÇÕES:

1. A função strcmp( ) possui uma variante, a função strncmp( ) que compara os n primeiros caracteres de duas strings. Sua sintaxe é: strncmp(string1,string2,nr_caracteres)
2. Existem ainda as funções stricmp( ) e strncmpi( ) que comparam duas strings sem considerar a caixa das letras (maiúsculas ou minúsculas).

6 - Convertendo strings em números

Para converter strings em números utilize as funções abaixo:

FUNÇÃO	CONVERTE STRINGS EM
atof(string)	float
atoi(string)	int
atol(string)	long int
strtod(string)	double
strtol(string)	long

Exemplo:

```
/* Convertendo strings em números */

#include <stdio.h>
#include <stdlib.h>

int main(void)
{    char string1[20],string2[20];

   printf("\n");
   printf("Convertendo strings em números\n");
   printf("------------------------------\n");
   printf("\n");
   printf("Entre com a primeira string :");
   scanf("%s",string1);
   printf("\n");
   printf("Entre com a segunda string :");
   scanf("\n%s",string2);
   printf("\n");
   printf("string1 + string2 = %f",atof(string1) + atof(string2));
   return(0);
}
```

Observações:

1. Toda string termina com “\0”, esse caracter é equivalente ao inteiro 0 (zero). O programa coloca automaticamente esse caracter no fim da string, então não se esqueça de contar que vc precisa dessa casa no vetor!!!
2. Se você for ler mais de uma string (ou caracter), a partir da segunda leitura feita com scanf(), precisa colocar um caracter “\n” antes do formato da entrada. Se você não fizer isso, o valor da string que está lendo não será o esperado!!! Exemplo: scanf(“\n%s”, string);
3. Note que você não deve digitar o & antes do nome do vetor para ler uma string!!! Isso acontece porque o nome do vetor já é o próprio endereço de memória da primeira posição do vetor.
4. Como o nome do vetor é um endereço de memória, se você tiver dois vetores: cadeia1 e cadeia2 e fizer cadeia1=cadeia2, o resultado não é o esperado!!!! Ele vai alterar o endereõ de memória do vetor cadeia1 e vai perder o seu conteúdo!!!! Use as funções de manipulação de strings!
5. Normalmente, quando se usa scanf(), essa função assume que se você digitou espaço em branco “ “, é por que terminou de digitar a entrada. Por isso, toda string só será lida até o primeiro espaço em branco. Para ler a string toda, incluindo espaços em branco, faça:

   scanf(“%[A-Z a-z]s”,string)