



Funções Básicas para a Manipulação de Strings:

Uma string é um *array* de caracteres terminado por um caracter nulo `'\0'` :

p	a	l	a	v	r	a	\0
---	---	---	---	---	---	---	----

As funções básicas para se manipular uma string estão prototipadas na biblioteca *string.h* e devem estar incluídas no programa através do código:

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

Protótipo das Funções Básicas	Funcionalidade
<code>char *strcpy (const char *dest,const char *src)</code>	Copia uma string em outra.
<code>int strcmp(const char *string1,const char *string2)</code>	Compara duas strings e determina a ordem (alfabetica) das duas
<code>int strlen(const char *string)</code>	Retorna o tamanho de uma string.
<code>char *strncat(const char *string1, char *string2, size_t n)</code>	Concatena n caracteres da string2 na string1.
<code>int strncmp(const char *string1, char *string2, size_t n)</code>	Compara os n primeiros caracteres de duas strings.
<code>char *strncpy(const char *string1,const char *string2, size_t n)</code>	Copia os n primeiros caracteres da string1 na string2
<code>int strcasecmp(const char *s1, const char *s2)</code>	Versão case insensitive de <code>strcmp()</code> .
<code>int strncasecmp(const char *s1, const char *s2, int n)</code>	versão case insensitive de <code>strncmp()</code> .

Exemplo de declaração de strings e uso das funções *strlen* e *strcpy*:

```
char *str1 = "HELLO";  
char *str2;  
int length;  
  
length = strlen("HELLO"); /* length = 5 */  
strcpy(str2,str1);
```

A biblioteca *string.h* contém ainda funções para busca em strings. A tabela a seguir mostra algumas delas:

Protótipo de Funções de Busca	Funcionalidade
<code>char *strchr(const char *string, int c)</code>	Encontra a primeira ocorrência do caracter <i>c</i> na string.
<code>char *strrchr(const char *string, int c)</code>	Encontra a última ocorrência do caracter <i>c</i> na string.
<code>char *strstr(const char *s1, const char *s2)</code>	Localiza a primeira ocorrência da string <i>s2</i> em <i>s1</i> .
<code>char *strpbrk(const char *s1, const char *s2)</code>	Retorna um ponteiro para a primeira ocorrência em <i>s1</i> de qualquer caracter de <i>s2</i> .
<code>char *strtok(char *s1, const char *s2)</code>	Quebra a string <i>s1</i> numa sequência de <i>tokens</i> delimitados por um ou mais caracteres de <i>s2</i> .

Por exemplo, temos o código a seguir utilizando a função *strtok()*:

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

int main()
{
    int operando, cont = 0;
    char *pt;
    char op;
    char string[30];
    printf("entre com uma expressao: ");
    scanf("%s", string);
    pt = strtok(string, " ");
    while(pt) //enquanto pt != NULL
    {
        cont++;
        printf("\n token %d: %s", cont, pt);
        pt = strtok(NULL, " ");
    }

    printf("\n cont = %d \n", cont);
    system("pause");
}
```

Exercícios:

1) Seja o código abaixo que recebe uma string e devolve uma cópia da string com as letras minúsculas transformadas em maiúsculas:

```
#include <stdio.h>
#include <malloc.h>
#include <ctype.h>

char *copia(char *s);
```

```

main()
{
    char str[81];
    char *str1;

    printf("digite uma linha \n");
    scanf("%80[^\n]",str);
    str1 = copia(str);
    printf("\nNova linha: %s",str1);
}

char *copia(char *s)
{
    int i;
    for(i=0;s[i]!='\0';i++)
        s[i]=toupper(s[i]);
    char *t=(char *)malloc(strlen(s)+1);
    strcpy(t,s);
    return t;
}

```

(a) Crie uma nova função que retorna o número de letras, o número de dígitos e o número de símbolos contido na string de entrada.

(b) Crie uma nova função que retorna apenas as letras transformadas para maiúsculo, descartando símbolos e dígitos.

(c) Crie uma nova função para substituir vários espaços em brancos por apenas um.

2) Faça um programa que lê uma string e retorna a string invertida.

3) Escreva um programa que lê uma string e retorna verdadeiro se a string de entrada é uma palíndrome.

4) Codifique um programa que recebe uma string e substitui cada letra pela letra seguinte circularmente (A é substituído por B, ... e Z é substituído por A). Escreva funções para codificar e decodificar strings.

5) Escreva um programa que funciona como o jogo da forca. O usuário tenta acertar uma palavra (contida numa variável) usando 5 tentativas. A palavra é mostrada ao usuário com as letras marcadas com o símbolo '*'. O usuário deve digitar uma letra (sem acentos, til, trema) e caso a letra esteja correta, o símbolo '*' é substituído pela letra. Se o usuário não montar a palavra até a quinta tentativa, o jogo acaba em derrota.