

Strings

Uma *string* é uma sequência de caracteres que permite representar nomes, endereços e outras informações textuais.

Declaração

Em C, *strings* são implementadas como vetores de caracteres terminados pelo caractere especial `"` (leia-se *barra zero*). Este caractere deve ser considerado ao medir tamanho do vetor. As aspas duplas ("`...`") são usadas para declarar *strings* constantes.

Exemplos:

```
char nome[100] ; // variável com até 99 caracteres mais o '\0'
char *endereco ; // endereço que aponta para uma string, mas sem alocar espaço
endereco = "Rua da Batata, 1000" ; // string constante, com '\0' no fim
```

Deve-se observar que uma *string* é um vetor de caracteres, então as duas declarações abaixo são equivalentes:

```
char codigo_1[] = "XT47A" ;
char codigo_2[] = { 'X', 'T', '4', '7', 'A', '\0' } ;
```

Além disso, a visão de *string* como vetor de caracteres permite o acesso aos caracteres individuais que a compõem. Na última linha, o `"(int)"` antes das constantes direciona o compilador a efetuar uma subtração dos códigos dos caracteres, que são inteiros. Os códigos de todos os caracteres podem ser vistos com **man ascii**.

```
for (i=0; i < strlen (str); i++) // varre todos os caracteres
    if (str[i] >= 'a' && str[i] <= 'z') // se for letra minúscula
        str[i] -= ((int)'A' - (int)'a') ; // converte em letra maiúscula
```

Leitura e escrita

A escrita de *strings* pode ser feita com `"printf"` (usando o formato `"%s"`), `"puts"` ou `"putchar"` (caractere a caractere):

```
#include <stdio.h>

char nome[] = "Machado de Assis" ; // constante de tipo string

printf ("Nome: %s\n", nome) ; // imprime o nome

puts (nome) ; // imprime o nome

for (i = 0; i < strlen(nome); i++) // imprime o nome, um char por vez
    putchar (nome[i]) ;
```

Similarmente, a leitura de *strings* pode ser feita por várias funções, como `"scanf"`, `"gets"` ou `"fgets"`:

```
#include <stdio.h>

#define SIZE 100

char nome[SIZE] ;

printf ("Digite seu nome: ") ;
```

```
// leitura usando scanf
scanf ("%s", nome) ;           // sem & (por que?)
scanf ("%[^\\n]", nome);       // lê string com espaços (até achar '\\n')

// leitura usando gets / fgets
gets (nome) ;                  // função perigosa, não usar!
fgets (nome, SIZE-1, stdin) ;  // lê até '\\n' ou SIZE-1 caracteres
```

Para mais informações sobre as funções acima, consulte as respectivas páginas de manual.

Observe que a leitura de uma *string* deve ser feita para uma variável com espaço suficiente para recebê-la, para não gerar um estouro de buffer (*buffer overflow*).

Manipulação

A manipulação de *strings* é geralmente efetuada através das funções disponíveis na biblioteca padrão C, que estão declaradas no arquivo de cabeçalho "string.h". Algumas das funções mais usadas são:

```
int strlen (str)
```

Informa o número de caracteres da *string* (sem considerar o " final).

```
char * strcpy (dest, src)
```

Copia a *string* de origem ("src") para o local indicado como destino ("dest"). Observe que a área de memória de destino deve ter sido previamente alocada, e ser grande o suficiente para conter toda a *string*.

```
char * strncpy (dest, src, n)
```

Copia "n" caracteres da *string* de origem ("src") para o local indicado como destino ("dest").

```
int strcmp (st1, st2)
```

Compara as duas strings indicadas, retornando 0 se forem iguais, <0 se st1<st2 e >0 se st1>st2.

```
int strncmp (str1, str2 ,n)
```

Idem, mas só considera os "n" primeiros caracteres.

```
char * strcat (dest, src)
```

Concatena a *string* "src" ao final da *string* "dest", para espaço previamente alocado, e com espaço suficiente.

```
char * strncat (strto, strfrom, n)
```

Idem, mas só concatena os "n" primeiros caracteres.

```
char * strchr (str, c)
```

Retorna o endereço (um ponteiro) para a primeira ocorrência do caractere "c" na *string* "str", ou NULL se não encontrar.

```
char * strrchr (str, c)
```

Idem, mas retorna o endereço (um ponteiro) para a **última** ocorrência do caractere.

Outras funções para manipulação de strings estão disponíveis na página de manual "string" (comando "man string") ;

Exercícios

Escreva programas em C para atender ao especificado abaixo. Cada programa deve ler uma ou mais *strings* da entrada padrão e escrever o resultado na saída padrão.

Diga

```
wget http://www.inf.ufpr.br/roberto/ci067/a05_main.c
```

para baixar o arquivo com o esqueleto das soluções para os problemas.

1. Calcular o tamanho de uma *string* sem usar "strlen".
2. Concatenar duas *strings* sem usar "strcat".
3. Copiar uma string para outra string sem usar "strcpy".
4. Compare o desempenho de sua implementação com aquele das funções da LibC. Sugestão: meça o tempo necessário para ativar cada função um milhão de vezes, e use o programa **time** para medir o tempo de execução:

time ./meu_strlen
5. Converter as letras de uma *string* para maiúsculas.
6. Ler uma *string* da entrada padrão e escrevê-la na saída padrão ao contrário (do final para o início).
7. Remover de uma *string* os caracteres que não sejam letras, números ou espaço, sem usar *string* auxiliar.
8. Remover de uma *string* caracteres repetidos em sequência (rr, ss, ee, etc), sem usar *string* auxiliar.
9. Colocar entre colchetes ([]) os caracteres de uma *string* que não sejam letras, números ou espaço; as alterações devem ser feitas na própria string, sem usar *string* auxiliar.