

## Strings

1. Escreva um programa em C com os seguintes passos:
  - a) ler uma *string* a partir do teclado;
  - b) escrever no écran a *string* lida anteriormente.
2. Escreva um programa em C com os seguintes passos:
  - a) ler uma *string* a partir do teclado;
  - b) calcular o comprimento da *string* através duma função;
  - c) escrever no écran o comprimento da *string*.
3. Implementar uma função para calcular o comprimento de uma *string* *s*.
4. Implementar uma função que, dada uma *string*, teste se cada carácter é uma letra e, em caso afirmativo, converta a letra na sua máiscula. Por exemplo, "O mote e o poeta sao 2 faces da mesma moeda" é convertida em "O MOTE E O POETA SAO 2 FACES DA MESMA MOEDA".
5. Implementar uma função para verificar se uma *string* é *substring* doutra. Por exemplo, "ver" é uma *substring* de "universo".
6. Implementar uma função para determinar o número de ocorrências do carácter *c* numa *string* *s*.
7. Implementar uma função para remover todas as ocorrências do carácter *c* numa *string* *s*.
8. Implementar os subprogramas seguintes:
  - a) int lers (char s[], int lim) → lê uma *string* *s* de tamanho máximo **lim**
  - b) void copia (char s[], char t[]) → copia a *string* *s* para *t*
  - c) void concat (char s[], char t[]) → acrescenta (concatena) a *string* *t* no fim de *s*
  - d) int compara (char s[], char t[]) → devolve 0 se  $t = s$ ,  $< 0$  se  $s > t$  e  $> 0$  se  $t > s$
  - e) void minusculas (char s[]) → converte todas as letras de *s* para minúsculas
  - f) void remove (char c, char s[]) → remove o carácter *c* de *s*
  - g) void inverte (char s[]) → inverte a *string* *s*
  - h) int strcounta (char \*s) → devolve o número de caracteres alfabéticos em *s*
  - i) char \*strset (char \*s, char c, int n) → coloca o carácter *c* apenas nas *n* posições iniciais da *string* *s*
  - j) int strwords (char \*s) → devolve o número de palavras duma *string* *s*
  - k) int strpack (char \*s) → recebe uma *string* *s* e compacta num único carácter qualquer sequência de caracteres repetidos /ex: "Arremesso" → "Aremeso")
9. Implementar uma função para devolver a posição da primeira ocorrência da *string* *t* na *string* *s* (devolver -1 se *t* não existir).

**10.** Implementar uma função para verificar se uma dada string `s` é capicua (string cuja leitura é a mesma quando feita em qualquer sentido) — devolver 1 se `s` for capicua.