

7.1 Escreva uma função `void capitalizar(char str[])` que transforma todas as letras duma cadeia em maiúsculas; outros caracteres devem ficar inalterados. O argumento é uma cadeia de caracteres (não necessariamente letras) terminada por `\0`.

Sugestão: use a função `toupper` da biblioteca padrão para transformar cada carater em maiúscula.

7.2 Escreva uma função `int palindromo(char str[])` que testa se uma cadeia de caracteres é um palíndromo, isto é, se tem a mesma sequência de caracteres da esquerda para a direita e vice-versa.

Sugestão: modifique a função apresentada na aula teórica 14 para inverter uma cadeia.

7.3 Escreva uma função `int todos_letras(char str[])` que testa se uma cadeia contém apenas caracteres letras (maiúsculas ou minúsculas). O resultado deve ser 1 em caso afirmativo e 0 em caso negativo.

Sugestão: utilize a função `isalpha` da biblioteca-padrão.

7.4 Escreva uma função `int algum_digito(char str[])` que testa se uma cadeia contém algum carater de dígito decimal (`'0'` e `'9'`). O resultado deve ser 1 em caso afirmativo e 0 em caso negativo.

Sugestão: utilize a função `isdigit` da biblioteca-padrão.

▷ **7.5** Escreva uma função `int forte(char str[])` que verifica se uma cadeia de caracteres é uma *palavra passe forte* usando o seguinte critério:

1. deve ter pelo menos 6 caracteres;
2. deve conter pelo menos uma letra maiúscula, uma letra minúscula e um algarismo.

O resultado da função deve ser 1 se ambos os critérios se verificam e 0 caso contrário. Por exemplo: `"Abr4cadabra"` e `"Apric0t"` são palavras-passe fortes, mas `"Ub40"` não é (porque o comprimento é inferior a 6) e `"POLICE"` também não (porque só tem letras maiúsculas).

Sugestão: pode usar a função `strlen` da biblioteca-padrão para calcular o comprimento da cadeia e as funções `islower`, `isupper`, `isdigit` para testar os três tipos de caracteres.

7.6 Escreva uma função `int decimal(char str[])` que converte uma cadeia de caracteres com algarismos de 0 a 9 no valor inteiro decimal correspondente. Por exemplo: `decimal("1234")` deve dar retornar o inteiro 1234.

7.7 Escreva uma função `int calc(char str[])` que implementa uma mini-calculadora: a cadeia de caracteres dada tem sempre 3 caracteres correspondentes a dois algarismos decimais (`'0'` até `'9'`) e um sinal de operação no meio (`'+'`, `'-'`, `'*'`). A função deve calcular o valor da expressão e retornar o inteiro correspondente.

Exemplos: `calc("5-3")` dá 2; `calc("2*3")` dá 6.

7.8 Defina uma função `int scrabble(char str[])` que calcula a pontuação de uma palavra dada como cadeia de caracteres; pode assumir que a cadeia é constituída apenas por letras maiúsculas sem acentos.

Recorde que a pontuação de palavras em inglês no jogo *SCRABBLE*: A,E,I,L,N,O,R,T,S,U: 1 ponto; D,G: 2 pontos; B,C,M,P: 3 pontos; F,H,V,W,Y: 4 pontos; K: 5 pontos; J,X: 8 pontos; Q,Z: 10 pontos. A pontuação duma palavra é a soma dos pontos de letras individuais.¹ Exemplo: “PITFALL” vale $3 + 1 + 1 + 4 + 1 + 1 + 1 = 12$ pontos, logo `scrabble("PITFALL")` dá resultado 12.

7.9 Escreva uma função `int contar_maiores(int vec[], int size, int val)` cujos argumento são uma variável indexada `vec` com tamanho `size` e um valor `val` e cujo resultado deve ser a contagem do número de elementos de `vec` que são estritamente maiores do que `val`.

▷ **7.10** Escreva uma função `int filtrar_positivos(int vec[], int size)` que remove os valores não positivos (isto é, negativos ou zero) de um vector `vec` com tamanho `size`.

A função deve modificar a variável indexada dada de forma a que os valores positivos fiquem num segmento inicial do vetor. O resultado deve ser o número de valores positivos (i.e. o comprimento do segmento final).

Sugestão: pode utilizar uma técnica análoga à usada na função para eliminar repetidos apresentada na aula teórica 16.

Apêndice: funções da biblioteca-padrão

```
#include <ctype.h>

int isupper(int ch); // testes
int islower(int ch);
int isdigit(int ch);

int toupper(int ch); // converter em maiúsculas
int tolower(int ch); // converter em minúsculas

#include <string.h>

size_t strlen(char str[]); /* comprimento */
```

¹ Há outras condições que afetam a pontuação mas que vamos ignorar neste exercício.