

Ficha 4

Programação Imperativa

Strings e Matrizes

1. Defina uma função `int minusculas (char s[])` que substitui na string `s` todas as letras maiúsculas pela correspondente letra minúscula.
A função deverá retornar quantas substituições foram efectuadas.
2. Defina uma função `int contalinhas (char s[])` que calcula quantas linhas tem uma string (assuma que o caracter que separa é duas linhas é `'\n'`).
3. Defina uma função `int contaPal (char s[])` que calcula quantas palavras tem uma string. Assuma que uma palavra é um conjunto de caracteres *não brancos* terminada por um *caracter branco*. Use para isso a função standard (`ctype.h`) `int isspace(int c)`.
4. Defina uma função `int procura (char *p, char *ps[], int N)` que procura uma string `p` num array de strings `ps`.
5. Redefina a função anterior assumindo que o array `ps` está ordenado por ordem crescente (ordem alfabética).
6. Uma forma muito mais eficiente de fazer a procura em arrays ordenados (e conhecida por **procura binária**) consiste em começar por comparar o elemento a procurar com o que está no meio do array. No caso de não se tratar do elemento em causa, podemos reduzir o problema à procura na metade inferior do array ou na metade superior.
Apresente duas versões, uma recursiva e outra iterativa para este método de procura.
7. Defina uma função `int zeros (int N, int M, int m [N] [M])` que conta quantos zeros tem uma matriz.
8. Defina uma função `void identidade (int N, int m[N] [N])` que preenche a matriz `m` com a matriz identidade de ordem `N`.
9. Defina uma função `void multV (float t[N] [N], float v[N], float r[N])` que multiplica a matriz `m` pelo vector `v`, colocando o resultado no vector `r`.