

# Introdução à Programação - ISCTE-IUL

## Exercícios — Aulas 1 a 6 — Parte 1

O objetivo desta aula é experimentar a definição de funções e procedimentos que envolvam os conteúdos leccionados nas aulas 1 a 6.

### Trabalho prévio

Resolver os exercícios propostos nas aulas práticas 1 a 6.

### Exercícios

A representação de caracteres em Java baseia-se no código [Unicode](#). Historicamente, o Unicode pode ser visto como uma extensão do [ASCII](#) (American Standard Code for Information Interchange), um dos primeiros sistemas digitais de codificação de caracteres. No ASCII as letras do alfabeto inglês são representadas por códigos em sequência: 'A' - 'Z' e 'a' - 'z'. Para além do alfabeto inglês, o ASCII definia também códigos para os dígitos, alguns sinais de pontuação e símbolos de controlo. O Unicode usa os mesmos códigos, mas estende a representação base para permitir a representação dos alfabetos de todas as línguas do mundo, para além de muitos outros símbolos.

Considere a função

```
static int diff(char c) {  
    return c - 'a';  
}
```

que calcula a diferença entre o carácter recebido no parâmetro `c` e a letra minúscula 'a'. Por exemplo,

```
diff('c') devolve 2  
diff('f') devolve 5  
diff('z') + 1 corresponde ao número de letras do alfabeto ('z' - 'a' + 1)
```

- A. Defina uma função `static boolean isLower(char c)` que indica se o carácter `c` é ou não uma letra minúscula.
- B. Defina uma função que dado um vetor de caracteres, devolva quantas letras minúsculas o vetor contém.
- C. Defina uma função que dado um vetor de caracteres, devolva um novo vetor apenas com as letras minúsculas.
- D. Defina uma função que dada uma matriz de caracteres, devolva quantas letras minúsculas a matriz contém. De preferência use a função da alínea B.
- E. Defina uma função que dados um carácter e um vetor de caracteres, devolva quantas vezes o carácter aparece no vector.
- F. Defina uma função que dado um vetor de caracteres, devolva se existe algum carácter repetido. Faça de duas maneiras diferentes, uma delas usando a função da alínea E.
- G. Defina uma função `static char uppercase(char c)` que dada uma letra minúscula, devolva a correspondente maiúscula. Se a função receber um carácter que não é uma letra minúscula, deve devolver o carácter recebido.  
`uppercase('a') → 'A'`  
`uppercase('%') → '%'`
- H. Defina um procedimento que receba um vetor de caracteres e converta cada letra minúscula nesse vetor na maiúscula correspondente.
- I. Defina um procedimento que receba uma matriz de caracteres e converta cada letra minúscula nessa matriz na correspondente maiúscula. Faça de duas maneiras diferentes, uma delas usando o procedimento da alínea H.
- J. Defina um procedimento que dado um vetor de caracteres e dois índices,  $i$  e  $j$ , coloque, caso ambas as posições contenham letras minúsculas, as duas letras existentes por ordem alfabética.  
`sortPositions({'b', 'a', 'c', '+', '-'}, 0, 1) → {'a', 'b', 'c', '+', '-'}`  
`sortPositions({'b', 'a', 'c', '+', '-'}, 1, 2) → {'b', 'a', 'c', '+', '-'}`
- K. Defina um procedimento que dado um vetor de caracteres coloque na primeira posição a menor letra minúscula (por ordem alfabética). Faça pelos menos de duas maneiras diferentes, uma delas usando a função da alínea J.
- L. Defina um procedimento que dado um vector de caracteres ordene o seu conteúdo.