

Folha prática 12 Introdução à Programação, DCC/FCUP, 2009/2010

Notas. Nos exercícios 1, 2 e 3 pode apenas utilizar a função `len` e os operadores como concatenação (“+”). Se começar por fazer o exercício 3, não necessita obviamente de fazer o exercício 2.

1. A representação de um inteiro

Escreva uma função `num(n)` que retorna a representação na base 10 do inteiro `n`. Assume-se que `n` não é negativo. Exemplos.

```
num(5)    -> "5"
num(0)    -> "0"
num(12345) -> "12345"
```

2. A representação de uma lista de inteiros

Generalize a função que implementou no exercício anterior, escrevendo uma função `repr(li)` que retorna a representação de `li` que pode ser um inteiro ou uma lista de 0 ou mais inteiros. Exemplos.

```
repr(5)    -> "5"
repr([0])  -> "[0]"
repr(0)    -> "0"
repr([2,12345]) -> "[2,12345]"
```

3. Ainda mais geral

Generalize a função que implementou no exercício anterior, assumindo agora um tipo T de `li` que é definido por

- Um inteiro (não negativo) é do tipo T .
- Uma lista de 0 ou mais elementos do tipo T tem o tipo T .
- Para além dos termos que resultam das 2 alíneas anteriores, mais nenhuma expressão é do tipo T .

Essencialmente está a afirmar-se que uma expressão pode ser um inteiro ou uma lista cujos membros são inteiros ou listas cujos membros etc...

Exemplos.

```
repr(5)    -> "5"
repr([0, [], [5], [2], 9, [[]]]) -> "[0, [], [5], [2], 9, [[]]]"
```

4. Aplicar a todos os elementos da lista

Escreva uma função `mapeia(j,li)` que retorna a lista que resulta de aplicar a função `f` a todos os elementos da lista `li`, conforme se ilustra

```
def f(n):
    return n*n+1

def mapeia(f,li):
    ??? função a escrever!

#-----
>>> print mapeia(f,range(4))
    [1, 2, 5, 10]

>>> from math import *
>>> print mapeia(sin,range(4))
    [0.0, 0.841, 0.909, 0.141]
```

Nota. Uma função como `mapeia(j,li)` em que um dos argumentos é uma função (neste caso `f`) chama-se uma função de ordem superior. Neste caso é de segunda ordem.

5. Copiar ficheiros

Escreva uma função `copia(f1,f2)`, onde `f1` e `f2` são nomes de ficheiros, cujo efeito é copiar todo o conteúdo de `f1` para `f2`. Método sugerido

```
copia(f1,f2):
    1) Abrir o ficheiro com nome f1 para leitura.
       Seja f a designação do ficheiro.
    2) Ler o conteúdo do ficheiro f para a string s.
    3) Fechar o ficheiro f.
    4) Abrir o ficheiro com nome f2 para escrita.
       Seja f a designação do ficheiro.
    5) Escrever a string s no ficheiro f.
    6) Fechar o ficheiro f.
```