

# Aula Teórico-prática 2

## Programação Funcional

LEI/LCC 1º ano (2006/2007)

### Funções não recursivas

#### Exercícios

1. Defina uma função `max2` que calcula o maior de dois números inteiros. Comece por definir a assinatura da função.
2. Defina uma função que calcula o maior de três números inteiros. Para isso apresente duas definições alternativas: recorrendo ou não à função `max3` definida na alínea anterior.
3. Num triângulo verifica-se sempre que a soma dos comprimentos de dois dos lados é superior (ou igual) à do terceiro. A esta propriedade chama-se *desigualdade triangular*. Defina uma função que, dados três números, teste se esses números correspondem aos comprimentos dos lados de um triângulo.
4. Considere a seguinte definição:

```
opp :: (Int,(Int,Int)) -> Int
opp z = if ((fst z) == 1)
         then (fst (snd z)) + (snd (snd z))
         else if ((fst z) == 2)
              then (fst (snd z)) - (snd (snd z))
              else 0
```

Apresente uma definição alternativa que use concordância de padrões em vez dos *ifs*.

5. Defina uma função que recebe os (3) coeficientes de um polinómio de 2º grau e que calcula o número de raízes (reais) desse polinómio.
6. Usando a função anterior, defina uma função que, dados os coeficientes de um polinómio de 2º grau, calcula a lista das suas raízes reais.