

Programação I

Mais Funções (ficha 7)

1. Implemente a função `maximo()` que devolve o máximo de 3 valores.
2. Implemente a função `minimo()` que devolve o mínimo de 3 valores.
3. Implemente a função `triangulo()` que classifica um triângulo segundo os seus ângulos a partir do comprimento dos seus lados. Considerando L_{max} o comprimento do lado maior e L_1 e L_2 o comprimento dos outros dois lados, temos:

- $L_{max} \geq L_1 + L_2$: não existe triângulo
- $L_{max}^2 = L_1^2 + L_2^2$: triângulo rectângulo
- $L_{max}^2 < L_1^2 + L_2^2$: triângulo agudo
- $L_{max}^2 > L_1^2 + L_2^2$: triângulo obtuso

A função deve devolver: -1 quando não existe triângulo, 0 em presença de um triângulo rectângulo, 1 quando triângulo agudo e 2 quando é um triângulo obtuso.

4. Implemente as funções `raiz1()` e `raiz2()` que calcula as raízes reais de um polinómio de segundo grau ($y = ax^2 + bx + c$). As raízes são os valores de x para os quais o valor de $y = 0$ e podem ser calculadas através de

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Nota: Não se esqueça de verificar as condições em não existe raiz; defina uma função auxiliar `nRaizes()` que indica quantas raízes tem o polinómio.

5. Implemente a função `mdc()` que, dados dois números inteiros, calcula o máximo divisor comum entre eles.
6. Implemente a função `mmc()` que, dados dois números inteiros, calcula o mínimo múltiplo comum entre eles.
7. Implemente a função `perfeito()` que, dado um número inteiro, indica se é perfeito ou não. Um número é perfeito se for igual a soma dos seus divisores excluindo ele próprio; por exemplo, 6 é perfeito já que $6 = 1 + 2 + 3$.
8. Implemente a função `somacubos()` que, dado um número inteiro de três algarismos, indica se ele é igual à soma do cubo dos algarismos que o constitui. Um exemplo de tal número seria 153 já que $153 = 1^3 + 5^3 + 3^3$.
9. Implemente a função `serie()` que, dado um número x calcula o resultado dos n primeiros termos da série (n é também um parâmetro da função)

$$x - \frac{1}{3}x^3 + \frac{1}{5}x^5 - \frac{1}{7}x^7 + \dots$$

10. Implemente a função `fatorial()` que, dado um número, calcula o seu fatorial. O fatorial pode ser calculado da seguinte forma

$$n! = n \times (n - 1) \times (n - 2) \dots 3 \times 2 \times 1$$