



## Lista de exercícios 2

1. Faça uma função que receba:
  - a. Dois valores e retorne o maior elemento, sem usar a função `max(x, y)`;
  - b. Um número  $n$  e retorne o valor absoluto ( $|n|$ ), sem usar a função `abs(n)`;
  - c. Um número e retorne `True` se o número for divisível por 5 e `False`, caso contrário;
  - d. Três números e os imprima em ordem crescente;
  - e. Três números e os retorne em ordem crescente;
  - f. O valor do raio de uma esfera, calcule, imprima e retorne o valor da área e do volume da esfera.
  - g. Três números como parâmetros e retorne a soma dos quadrados dos dois maiores números;
  - h. Um valor em segundos e escreva o horário no formato hh:mm:ss;
  - i. Cinco valores e retorne a soma e a média dos pares e a soma e a média dos ímpares;
  - j. Um valor e imprima o tipo desse valor (“Inteiro”, “Ponto flutuante”, “String”, “Booleano” ou de outro tipo);
  - k. Um número inteiro  $n$  e retorne `True` se  $n$  for um *número ascendente* e `False`, caso contrário. Caso  $n$  não seja um número inteiro ou não esteja entre  $[100, 999]$ , sua função deve exibir uma mensagem de erro e retornar `False`.
  - l. Um número inteiro  $n$  e retorne `True` se  $n$  for um *número palíndromo* e `False`, caso contrário. Caso  $n$  não seja um número inteiro ou não esteja entre  $[1000, 9999]$ , sua função deve exibir uma mensagem de erro e retornar `False`.
2. Em matemática, nomeadamente em teoria dos números, um *terno pitagórico* (ou *tripla pitagórica*) é formado por três números naturais  $a$ ,  $b$  e  $c$  tais que  $a^2 + b^2 = c^2$ . Faça uma função que receba três números naturais e imprima se os mesmos formam ou não uma *tripla pitagórica*.
3. Faça uma função que receba o valor de  $x$ , calcule e retorne o valor de  $f(x)$ :

$$f(x) = \begin{cases} \ln x, & \text{se } x \leq 1 \\ \log_{10} x + \sqrt{x}, & \text{se } 1 < x \leq 2 \\ x^2 + e^x, & \text{se } 2 < x \leq 5 \\ x^{x/2} + \log_2 x, & \text{se } x > 5 \end{cases}$$

4. Faça uma função que receba o tempo (horas, minutos e segundos) e a distância (em km) percorrida por um corredor e imprima o ritmo (min/km) do corredor. Faça uma função `main` que faz leitura dos dados (tempo e distância) e chama a função anterior. Veja dois exemplos:

Tempo do corredor (horas, minutos e segundos):

0

40

0

Distância percorrida (em km): 10

--> Ritmo: 04:00min/km

Tempo do corredor (horas, minutos e segundos):

2

1

39

Distância percorrida (em km): 42.195

--> Ritmo: 02:53 min/km

5. Faça um programa que contenha duas funções: `delta`, `raizes`. Ambas as funções devem receber os coeficientes de uma equação de segundo grau. A função `delta` deve retornar o valor do discriminante da equação ( $\Delta = b^2 - 4ac$ ). Já a função `raizes` deve utilizar a função anterior e calcular e imprimir as raízes da equação. Caso o discriminante seja negativo, a função `raizes` deve informar que a equação não possui raízes reais. Se o discriminante for zero, a função deve informar que as duas raízes são iguais e exibí-la. Por fim, se o  $\Delta > 0$ , a função deve exibir as duas raízes. Após a definição das duas funções, defina uma função `main` que faça a leitura dos coeficientes e use as funções para exibir as raízes.
6. O número 3025 possui uma propriedade interessante: ao se dividir o número em duas partes, somar essas partes, e elevar a soma ao quadrado, o resultado é o próprio número.  

$$30 + 25 = 55 \Rightarrow 55^2 = 3025$$
  - a. Faça uma função para verificar se um dado número tem essa propriedade. A função deve receber um número inteiro e retornar `True` ou `False`, indicando, respectivamente, se o número tem essa propriedade ou não. Se o número não tiver 4 dígitos, ela também retorna `False`.
  - b. Faça uma função solicite ao usuário que digite um número de quatro dígitos e, em seguida, imprime se o número possui ou não essa propriedade. Essa função não recebe nem retorna valor algum, apenas escreve. Ela deve utilizar a função anterior.
7. Faça um programa que contenha duas funções: `ehTriangulo` e `tipoTriangulo`. A função `ehTriangulo` deve receber as medidas dos lados de um triângulo e retornar `True` se essas medidas podem formar um triângulo e `False`, caso contrário (o exercício 13 da Lista 1 pode te ajudar). Já a função `tipoTriangulo` deve receber as medidas dos lados de um triângulo e exibir o tipo do triângulo: *equilátero* ou *isósceles* ou *escaleno*. Essa função deve, inicialmente, verificar se as medidas formam um triângulo (usando a função `ehTriangulo`), caso não forme, exiba uma mensagem. Após a definição das duas funções, faça uma função `main` que faz a leitura dos três lados de um triângulo e use a função `tipoTriangulo` para exibir o tipo de triângulo.

8. Faça o rastreio (teste de mesa) do programa a baixo. O que a função `mystery` faz?

```
def min(a, b):  
    if a < b:  
        return a  
    return b  
  
def mystery(a, b, c, d):  
    x = min(a, b)  
    y = min(c, d)  
    return min(x, y)  
  
def main():  
    a = 10  
    b = 15  
    c = 20  
    d = 5  
    print(mystery(a, b, c, d))  
  
main()
```

Boa diversão!!!