

Folha de apoio à aula prática 5 (exercícios para praticar antes da aula).

5.1 Considere um programa que começa com a atribuição de uma lista de valores de temperatura (em Celsius) à variável `tempC`:

```
tempC = [-5,0,5,10,15,20,25]
```

- (a) Escreva um ciclo `for` que imprime cada um dos valores da lista `tempC` numa linha separada.
- (b) Imprima os mesmos valores, gerando os índices com a função `range`.
- (c) Escreva um ciclo `while` que produza o mesmo resultado da alínea anterior.
- (d) Escreva um ciclo em que para os valores de temperatura acima, em cada linha imprime o valor em *Celsius* e o valor correspondente em *Fahrenheit*.

5.2 O fatorial de um número natural n , representado por $n!$, é dado pelo produto $1 \times 2 \times \dots \times n$. Por definição, $0! = 1$. Implemente uma função que calcule o fatorial de um número:

- (a) com um ciclo `for`
- (b) com um ciclo `while`
- (c) teste as suas funções com os seguintes argumentos: 0, 1, 2, 3, 10, 1000, 'zero' e 3.14159.
- (d) complete a definição de cada uma das funções para que, quando o argumento da função não é válido, imprima uma mensagem de erro. (Use a função `type`.)

5.3 Num triângulo cada lado tem um comprimento menor que a soma dos comprimentos dos outros dois e maior que a sua diferença absoluta. Escreva uma função `triangulo(a,b,c)` que verifica esta condição sobre os lados a, b, c ; o resultado deve ser `True` ou `False`.

5.4 Escreva uma função `classifica(p)` que, dada a pontuação p obtida num exame (de 0 a 100), retorna uma mensagem de classificação de acordo com a tabela seguinte. Utilize essa função para escrever um programa que imprime o número, nome e classificação segundo essa tabela da lista de alunos que se segue (cada tuplo contém o número de aluno, o seu nome, e a classificação de 0 a 100).

pontuação	mensagem	studs =
< 0 ou > 100	"inválido"	(("UP194187304", "José Fonseca", 97),
< 50	"insuficiente"	("UP194209183", "Manuel Ferreira", 87),
≥ 50, < 70	"suficiente"	("UP194294793", "Maria Ramos", 50)
≥ 70, < 80	"bom"	("UP194399128", "Antonio Fernandes", 45),
≥ 80, < 90	"muito bom"	("UP194739873", "Júlia Pinto", -1),
≥ 90	"excelente"	("UP194739889", "Manuela Faria", 50)]

5.5 Um ano é *bissexto* se for divisível por 4, exceto se for múltiplo de 100 e não for divisível por 400. Escreva a função `bissexto(n)` que resulta em `True` se n for um ano bissexto e `False` no caso contrário.

5.6 Teste a função do exercício anterior (5.5) fazendo um programa que escreve uma tabela dos anos bissextos entre 2000 e 2100 (inclusive). Verifique os resultados usando o calendário do computador.

5.7 Um número n diz-se um quadrado perfeito se, para algum natural k , pode ser escrito como a soma dos k primeiros números ímpares, isto é, $1 + 3 + \dots + k$. Os primeiros cinco quadrados perfeitos são

1, 4, 9, 16 e 25. Escreva uma função `quadrado_perfeito(n)` cujo resultado é `True`, se n é um quadrado perfeito. Caso contrário, o resultado deverá ser `False`.

5.8 Podemos contar algarismos decimais na representação de um número fazendo divisões inteiras por dez. Por exemplo: 9733 tem 4 algarismos porque podemos fazer quatro divisões sucessivas por 10 obtendo os quocientes 973, 97, 9 e 0 (paramos quando chegamos a zero). Escreva uma função `algarismos(n)` cujo resultado é o número de algarismos decimais de n . Sugestão: use um ciclo `while` para repetir as divisões sucessivas e contar o número de iterações.

5.9 A fórmula de Leibniz para aproximar π é:

$$\pi = 4 \times \left(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} + \dots \right) = 4 \times \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1}$$

Implemente a função `leibniz(k)` que resulta no somatório dos primeiros k termos desta série. Documente a sua função com uma `docstring`.

5.10 A constante de uma mola pode ser determinada por regressão linear: dada uma lista de n observações da extensão da mola e_i quando nela são suspensos pesos p_i , a constante é estimada por

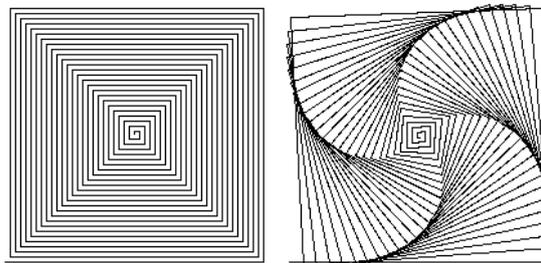
$$\frac{\sum_{i=1}^n e_i}{\sum_{i=1}^n p_i}.$$

Escreva a função `mola(e,p)` que, dadas listas com observações `e` e `p` retorna a constante da mola. (Nota: pode assumir que ambas as listas têm o mesmo número de observações.)

5.11 O *menor divisor próprio* de um inteiro n é o menor inteiro d tal que $d > 1$ e d divide n (ou seja: o resto da divisão de n por d é zero).

- Escreva a definição duma função `mindiv(n)` que calcula o menor divisor próprio.
- Um inteiro n é *primo* se o seu menor divisor próprio for n . Escreva a definição duma função `primo(n)` cujo resultado é `True` ou `False` conforme n é ou não primo.
- Note que se d é o menor divisor próprio de n e $d > \sqrt{n}$ então $d = n$ (porquê?). Modifique a definição da função da alínea (a) para usar esta propriedade para tornar mais rápida a pesquisa do menor divisor próprio.

5.12 As espirais da figura ao lado foram desenhadas usando o módulo `turtle` apenas mudando o ângulo de rotação entre cada lado. Escreva uma função `espiral(...)` para desenhar espirais deste tipo.



5.13 A área A de um triângulo cujos lados medem a , b e c pode ser calculada usando a fórmula (atribuída ao matemático grego Heron)

$$A = \sqrt{s(s-a)(s-b)(s-c)}$$

onde $s = (a + b + c)/2$ é o semi-perímetro do triângulo. Implemente a função `area_triangulo(a,b,c)` que calcula a área de um triângulo usando esta fórmula.

5.14 Implemente a função `sum_mult_k(j, k, n)` que calcula a soma dos inteiros de 1 a n (inclusive) que verificam as seguintes condições:

- são múltiplos de j mas não de k , ou
- são múltiplos de k mas não de j

Por exemplo, o resultado de `sum_mult_k(3,2,8)` é 17.

5.15 Implemente a função `sum_within(x, a, b)` que calcula a soma dos valores da lista x que estão compreendidos entre a e b .

Por exemplo, o resultado de `sum_within([4, 7, 44, 23], 17, 46)` é 67.

5.16 Implemente a função `count_if(x, a)` que retorna no número de valores da lista x que são iguais a a .

Por exemplo, o resultado de `count_if([4, 1, 3, 0, 3, 3], 3)` é 3.

5.17 Implemente a função `prod_nomult_k(k, n)` que calcula o produto dos números naturais entre 1 e n (inclusive) que não são múltiplos de k . Por exemplo, o resultado de `prod_nomult_k(3, 8)` é 2240.

Na aula terá exercícios de avaliação contínua baseados nesta folha.