2.1 Normalmente calcula-se juros com base num número inteiro de períodos (o número de anos, por exemplo). No entanto, em algumas situações é útil calcular o juro como uma função contínua do tempo. A fórmula para calcular o valor de um investimento ao longo do tempo t é a seguinte:

$$P(t) = P_0\, e^{r\,t}$$

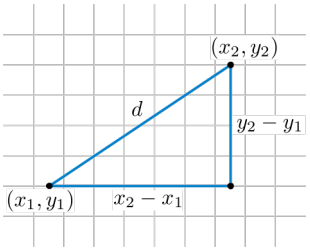
onde P_0 é o montante inicial investido, r é taxa de juro e t é tempo que passou desde o momento em que se fez o investimento.
Escreva uma função `P(P0, r, t)` que retorne o valor de um investimento P_0 no instante t, a uma taxa de juro r.

2.2

A distância entre dois pontos no plano de coordenadas (x_1, y_1) e (x_2, y_2) é:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Implemente uma função `dist(x1,y1,x2,y2)` que calcule a distância usando esta fórmula.



2.3 Escreva uma função `radianos(graus,mins,segs)` que, dado o valor de um ângulo em graus, minutos e segundos, converte-o para radianos. Relembre que 360^o corresponde a 2π radianos, cada grau tem 60 minutos e cada minuto tem 60 segundos.

2.4 Escreva uma função `segundos(horas,mins,segs)` que, dada uma duração em horas, minutos e segundos, calcula e retorna essa mesma duração em segundos.

2.5 Escreva um programa que lê uma quantia inteira de euros e mostra como pagar essa quantia em notas de 20€, 10€, 5€e moedas de 1€. Exemplo:

```
Quantia em EUR? 93
notas EUR 20: 4
notas EUR 10: 1
notas EUR  5: 0
moedas EUR 1: 3
```

Sugestão: O quociente da divisão da quantia total por 20 dá-nos o número de notas de 20€. Repita o processo para a quantia sobrante (o resto da divisão por 20) usando as notas mais pequenas (10€ e 5€). Tenha o cuidado de usar divisões inteiras e não vírgula-flutuante.

2.6 Considere um programa que começa com a seguinte atribuição:

```
xs = [5, 0, 42 ,10, 24, 30, 81]
```

(a) Escreva um ciclo `for` que imprime cada um dos valores da lista `xs` numa linha separada.

(b) Escreva um outro ciclo `for` que em cada linha, imprima o número, o seu quadrado, e a sua raiz quadrada.

(c) Escreva um ciclo que soma todos os números em `xs` usando uma variável auxiliar `total`, e imprime numa linha separada cada um dos números da lista e a soma parcial até esse número.

2.7 Escreva uma função `tabela_quadrados(n)` que, para os `n` primeiros números inteiros positivos, imprime em cada linha o número e o seu quadrado, separados por um espaço. Pode assumir que `n > 0`.

2.8 A fórmula para calcular o valor final de um depósito com juro composto capitalizado ao mês, a uma taxa anual r é

$$C_F = C_I \times (1 + r/12)^n$$

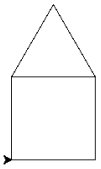
onde C_I é o capital inicial, C_F é o capital final, r é a taxa de juro e n é o número de meses de duração do depósito. Por exemplo: para um capital inicial de 1000€, uma taxa de juro anual de 4% durante 24 meses obtemos $C_F = 1000 \times (1 + 0.04/12)^{24} \approx 1083$€.

Escreva um programa que pergunta o capital inicial e a taxa de juro e que imprime uma tabela com o capital final após 0, 1,..., 24 meses.

2.9 Usando o módulo *turtle*, escreva uma função `poligono_reg(n,lado)`, sem valor de retorno, que desenha um polígono regular com n lados de comprimento *lado*. Por exemplo, `poligono_reg(3, 100)` desenha um triângulo equilátero com 100 *pixels* de lado.

Nota: a soma dos ângulos externos de um polígono é 360°.

2.10 Usando o módulo *turtle*, escreva uma função `casa(lado)`, sem valor de retorno, que desenha uma casa como na figura abaixo (um triângulo equilátero sobre um quadrado, ambos com o mesmo comprimento de lado). Sugestão: utilize a função `poligono_reg` do exercício 2.9.



2.11 Usando o módulo *turtle*, escreva uma função `friso(n,lado)`, sem valor de retorno, que desenha um friso em forma de muralha com n ameias em que a largura de cada segmento é *lado*. Por exemplo: `friso(3, 50)` produz o desenho da figura seguinte. Note que a tartaruga deve terminar com a orientação original.

2.12 A soma dos desvios quadrados é uma medida comum para avaliar erros. Dada uma lista de n desvios $(d_1, d_2, ..., d_n)$, esse valor é dado por $\sum_{i=1}^{n} d_i^2$. Implemente a função `sdq(d)`, que dada uma lista de desvos d, retorna a soma dos desvios quadrados.
Por exemplo, o resultado de `sdq([-6.9, 4.7])` é 69.7.

2.13 O preço atual da gasolina é 1.72 euros por litro. Implemente a função `valor(v)` que, dada a lista v de litros abastecidos numa viagem, retorna o valor total despendido.
Por exemplo, o resultado de `valor([24.8, 49.2])` é 127.28.

2.14 A área A de um triângulo cujos lados medem a, b e c pode ser calculada usando a fórmula (atribuida 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 uma função `area_triangulo(a,b,c)` que calcule a área de um triângulo usando esta fórmula.

2.15 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`.

2.16 Avalie as seguintes expressões:

(a) `3 == 3`

(b) `3 != 3`

(c) `3 >= 4`

(d) `not (3 < 4)`

2.17 Sem utilizar o operador `not`, escreva expressões que sejam o contrário (i.e., a negação) das que se seguem:

(a) `a > b`

(b) `a >= b`

(c) `a >= 18 and day == 3`

(d) `a >= 18 and day != 3`

2.18 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:

pontuação	**mensagem**
< 0 ou > 100	`"inválido"`
< 50	`"insuficiente"`
≥ 50, < 70	`"suficiente"`
≥ 70, < 80	`"bom"`
≥ 80, < 90	`"muito bom"`
≥ 90	`"excelente"`