

P1 Alg 2025-1

Nome e turma: _____

1) Escreva uma função chamada `maior_menor_que(lista, limite)` que receba uma lista de números inteiros e um número limite. A função deve retornar o maior número da lista que seja menor que o limite. Se todos os elementos forem maiores ou iguais ao limite, retorne `None`. Não use `sort` ou `max`. Exemplo: `maior_menor_que([10, 20, 30, 40], 35) → 30` e `maior_menor_que([50, 60, 70], 40) → None`.

2) Escreva uma função `gato_atrapalha_sono(miando, hora, feriado)` que receba três argumentos: `miando`, que é `True` quando o gato está miando e `False` caso contrário; `hora`, um inteiro de 0 a 23 representando a hora do dia; e `feriado`, que é `True` se for feriado e `False` caso contrário. Considere que em dias normais o aluno dorme entre 21h e 5h, e em feriados entre 23h e 8h. A função deve retornar `True` se o gato estiver miando durante o horário de sono, atrapalhando o descanso do aluno. Note que o gato miar exatamente na hora de acordar não atrapalha.

3) Escreva um programa que leia uma palavra por vez, digitada pelo usuário, até que ele digite `"sair"`. Armazene as palavras em uma lista. Em seguida, imprima todas as palavras com mais de 5 letras e que contenham **ao menos uma letra da palavra "python"**. Suponha que todas as palavras estejam em letras minúsculas.

4) Escreva uma função `válido(idade, renda)` que receba dois argumentos: a idade de uma pessoa (inteiro) e sua renda mensal (float ou inteiro). A função deve retornar `True` se a pessoa tiver 18 anos ou mais e renda de pelo menos R\$2500, ou se tiver 21 anos ou mais, independentemente da renda. Caso contrário, a função deve retornar `False`.

5) O código abaixo foi escrito para somar os valores de uma lista de preços, mas contém vários erros distintos (de sintaxe e lógica). **a)** Encontre todos os erros, indicando se são de sintaxe ou de lógica. **b)** Reescreva o código corrigido.

```
def soma_preços(lista):
    soma = 0.0
    for preço in lista:
        total = total + preço
    return total
preços = "19.90 35.00 12.50 9.90".split()
print = ("Total: R$" + soma_preços(preços))
```

6) Preocupado com os gastos mensais de energia em sua casa, você irá criar um programa em Python. A cada mês, o consumo base é de 120 kWh. Um novo aparelho comprado aumenta o consumo mensal em 10 kWh (de forma cumulativa e permanente). Sua família pretende comprar um novo aparelho a cada 3 meses. Faça um programa que calcule após quantos meses o consumo de um único mês ultrapassará 240 kWh, e imprima esse mês e o consumo atingido. (Observação: 10 kWh/mês é uma média realista para pequenos eletrodomésticos ou eletrônicos.)

7) Crie uma função `zigue_zague(s)` que, dada uma string `s`, transforme em **maiúsculas** as letras que estiverem em posições pares (começando a contar em 0).
Ex: `zigue_zague("Python")` → "PyThOn" e `zigue_zague("bAnana")` → "BANaNa"

8) O código abaixo calcula o n -ésimo termo da sequência de Fibonacci, definida por $f(0) = 1$, $f(1) = 1$ e $f(n) = f(n-1) + f(n-2)$:

```
def f(n):
    a, b = 1, 1
    k = 2
    while k <= n:
        a, b = b, a + b
        k = k + 1
    return b
```

Adapte esse código para calcular o n -ésimo termo da sequência de Pell, definida por $p(0) = 0$, $p(1) = 1$ e $p(n) = 2 \times p(n-1) + p(n-2)$, para $n \geq 2$. Implemente a função `pell(n)` aproveitando a lógica do código acima, fazendo as modificações necessárias. Ex: se `n = 5`, a sequência de Pell é: 0, 1, 2, 5, 12, 29. Logo, `pell(5)` deve retornar 29.

9) Considere o trecho de código abaixo, no qual a variável `x` não foi definida. O programa testa uma combinação de condições lógicas usando `and` e `or`. A seguir, são apresentadas quatro saídas diferentes. Para cada uma, determine **um possível valor inteiro** para `x` que produza a saída correspondente.

```
x = ? # descobrir
if x % 2 == 0 and x < 0:
    print("par negativo")
elif x % 2 != 0 and x > 0 and x < 10:
    print("ímpar pequeno")
elif x % 3 == 0 or x > 100:
    print("divisível por 3 ou grande")
else:
    print("caso genérico")
```

10) Execute o teste de mesa da função `processa(lista)` para a lista `[2, 4, 1, 6, 42, 13, 10, 5]`.

```
def processa(lista):
    x = 0
    achou = False
    for i in range(len(lista)):
        n = lista[i]
        if n == 42:
            achou = True
            break
        if n % 2 == 0: x = x + 1
        print(i, n, x, achou)
    print(x)
    return x
```