

## P1 Alg 2024-2

Nome e turma: \_\_\_\_\_

1. O código abaixo foi escrito para contar quantas palavras em uma lista começam com uma letra maiúscula. No entanto, o programa contém **quatro** erros distintos, dois de lógica e dois de sintaxe. a) Encontre os erros b) Refaça o código sem os erros.

```
def contar_maiusculas(palavras):  
    contador = 0  
    for palavra in palavras:  
        if palavra.upper[0] == palavra[0]:  
            contador += 1  
    return contador  
palavras = ["Casa", "carro", "Janela", "bicicleta", "Computador"]  
resultado = contar_maiusculas(palavras)  
print("Número de palavras que começam com letra maiúscula é: " +  
resultado)
```

2. Escreva uma função `def divisibilidade(n)` que receba um número inteiro positivo `n`. Se o número for divisível por 2, mostre a saída `"dois"`. Se ele for divisível por 7, mostre a saída `"sete"`. Se o número for divisível por 2 e 7 ao mesmo tempo, mostre a saída `"quatorze"`. Entretanto, se o número for divisível por 42 (regra máxima e prioritária), mostre a saída `"resposta para tudo"`. Caso o número não se enquadre em nenhuma dessas condições, mostre apenas o número lido na saída.
3. Você decidiu começar a economizar para uma viagem, e todo mês consegue guardar 10% do seu salário líquido de R\$ 1500. No entanto, você também tem uma despesa extra que ocorre a cada 4 meses, onde você precisa gastar R\$ 200 com manutenção do seu carro. Além disso, a cada 6 meses, você recebe um bônus no trabalho de R\$ 500, que você decide adicionar ao valor acumulado. Faça um programa que calcule quantos meses serão necessários para juntar R\$ 5000 para a viagem, considerando essas despesas e o bônus. O programa deve imprimir o número de meses necessários e quanto dinheiro sobrou após atingir o objetivo.
4. Escreva uma função `def seguido(n)` que receba um número inteiro positivo `n` e verifique se existem dois dígitos **adjacentes consecutivos** (com diferença de 1, seja crescente ou decrescente). A função deve devolver `True` se houver dois dígitos consecutivos e `False` caso contrário. `seguido(909090) → False`. `seguido(441255) → True`. `seguido(654314) → True`. `seguido(135790) → False`. `seguido(12) → True`. `seguido(33) → False`.
5. Escreva uma função `def troca(a, b)` que troque dois inteiros `a` e `b` de lugar, sem usar uma nova variável. Ex.: `troca_valores(3, 5)` # Saída: 5, 3

6. Escreva uma função `def maior(lista)` que receba uma lista de números e retorne o maior deles. Se a lista estiver vazia, retorne "Lista Vazia". Não use `max`, `min` ou `sort` nesta questão.
7. Escreva uma função `mult2(nums)` que recebe uma lista de inteiros de qualquer tamanho e retorna o produto (multiplicação) dos dois primeiros elementos. Se a lista tiver menos de dois elementos, multiplique o que for possível. Se a lista estiver vazia, retorne 1 (pois o produto de um conjunto vazio é 1). `mult2([2, 3, 4])` #6. `mult2([5])` #5. `mult2([])` #1.
8. Escreva uma função `string_crescendo(s)` que recebe uma string `s` e retorna uma nova string formada da seguinte maneira: para cada caractere da string original, repita esse caractere tantas vezes quanto sua posição na string (contando a partir de 1). `string_crescendo('abc')` #"abbccc". `string_crescendo('xy')` #"xyy". `string_crescendo('z')` #"zz". `string_crescendo('')` #"".
9. Execute o teste de mesa (simulação) do algoritmo abaixo para `m(4, 3)`, isto é, faça passo a passo a execução dele. Mostre todos os valores intermediários de `x`, `y` e `z`, conforme está indicado nos prints.

```
def m(y, z):  
    x = 0  
    while z > 0 :  
        print (x, y, z)  
        if z % 2 == 1:  
            x = x + y  
            y = 2 * y  
            z = z // 2  
    print (x)  
    return x
```

10. Execute o teste de mesa (simulação) do algoritmo abaixo para `s(3)`, isto é, faça passo a passo a execução dele. Mostre os valores intermediários de `n` e `soma`, conforme está indicado nos prints.

```
def s(n):  
    soma = 0  
    while n != 1:  
        print (n, soma)  
        soma = soma + n  
        if n%2 == 0:  
            n = n // 2  
        else:  
            n = n*3 + 1  
    print (soma)  
    return soma
```