

Universidade Federal do Maranhão  
Bacharelado Interdisciplinar em Ciência e Tecnologia  
**Algoritmos e Estrutura de Dados (CCCT0013) - 2018.2**  
Profa. Alana Oliveira

Aluno: \_\_\_\_\_ Matricula: \_\_\_\_\_

Lista de Exercícios 1 - (30%)

Instrução: Para uma correção mais eficiente, mandem cada questão numerada em formato do Python .py (Ex: lista1q1.py, lista1q2.py, lista1q3.py, ...). E depois compactar em um formato .zip com o nome de cada aluno ex: AlanaOliveira.zip (não enviar .rar porque não abre em todos os computadores) e enviar pelo SIGAA. **A não observância da instrução acima acarretará na anulação da atividade.**

**1ª)** Crie uma lista chamada `minha_lista` com os seguintes itens:

“Iron”, 200, “ola”, False, [], -1, 150. Para esta lista, escreva os seguintes comandos:

- a) Inserir “almoço” e 200 no final da lista.
- b) Inserir “unicornio” na posição de índice 2.
- c) Inserir 2018 no início da lista.
- d) Encontrar o índice de “olá”
- e) Contar o número de ocorrências de 200 na lista.
- f) Remover a primeira ocorrência do numero 200 da lista.

**2ª)** Escreva uma função `calcula_minimo_2` que receba como entrada uma lista de números inteiros (tamanho mínimo da lista de 3 elementos) e retorne soma dos dois menores valores na lista. Por exemplo, `calcula_minimo([4,3,6,1,2])` deve retornar 3 (1+2).

**3ª)** Escreva uma função `soma_quadrados` que receba uma lista de números e retorne a soma dos quadrados dos números na lista. Por exemplo, `soma_dos_quadrados([2, 3, 4])` deve retorna 4+9+16 que é 29.

**4ª)** Crie uma função `seleciona_alunos`, que recebe uma lista de alunos e um valor inteiro correspondente a sua nota (Ex: `[['Pedro', 8], ['Maria', 9.5 ], ... ]`), e retira todos os alunos da lista que possuem nota menor que 7.

**5ª)** Escreva uma função chamada `espelho` que recebe uma strings como parâmetro e anexa o conteúdo da string a si mesma em ordem inversa. Por exemplo: `[a, b, c] -> [a, b, c, c, b, a]`.

**6ª)** Escreva uma função `quantidade_palavras_8` que receba uma lista de palavras e retorne o número de palavras na lista que tenham comprimento 8. Por exemplo: `quantidade_palavras_8(['planilha', 'áreas', 'dados', 'Anaconda', 'Cientista'])` deve retornar 2.

**7ª)** Escreva uma função `radical` que receba um radical e uma lista de palavras e retorne o número de palavras na lista que tenham o radical fornecido. Por exemplo, `radical('part', ['partiu', 'parceiro', 'mesa', 'partida', 'parente'])` deve retornar 2.

8ª) Escreva uma função `palavras_inicio_fim_iguais` que receba como entrada uma lista de palavras e retorne a quantidade de palavras da lista que possuem 2 ou mais caracteres e cujos primeiro e últimos caracteres sejam iguais. Por exemplo, `palavras_inicio_fim_iguais(['aba', 'xyz', 'aa', 'x', 'bbb'])` retorna 3.

9ª) Um triângulo retângulo pode ter lados cujos comprimentos são todos inteiros. O conjunto de três valores inteiros para os comprimentos dos lados de um triângulo retângulo é chamado de Tripla de Pitágoras. Os comprimentos dos três lados devem obedecer à relação de que a soma dos quadrados de dois dos lados é igual ao quadrado da hipotenusa. Escreva um programa para identificar todas as triplas de Pitágoras para `lado1`, `lado2` e `hipotenusa`, não maiores que 500. Utilize um método de força bruta, com um loop for triplamente aninhado que tenta todas as possibilidades.

10ª) Escreva uma função `troca` que receba três strings `str1`, `velho` e `novo` e troca em `str1` todas as ocorrências de `velho` por `novo`. Por exemplo, `troca('Um aluno, dois alunos, tres alunos.', 'aluno', 'estudante')` deve retornar a string `'Um estudante, dois estudante, tres estudantes.'`. Observação: existe a função `replace` que faz isso, mas você não deve usá-la. (Dica: use os métodos `split` e `join`).

11ª) Escreva uma função `soma_impares` que receba uma lista de números inteiros e retorne a soma dos números ímpares na lista. Por exemplo, `soma_impares([11,20,36,41,55,6])` deve retornar 107.

12ª) Escreva uma função `quantidade_negativos` que receba uma lista de números inteiros e retorne a quantidade de números negativos na lista. Por exemplo, `quantidade_negativos([-1,-2,3,4,-5,-6])` deve retornar 4.

13ª) Qual o resultado de cada um dos comandos seguintes:

- a) `'Python'[1]`
- b) `'Strings são sequências de caracteres.'`[5]
- c) `len('maravilhoso')`
- d) `'Misterio'[:4]`
- e) `'p' in 'Pineapple'`
- f) `'apple' in 'Pineapple'`
- g) `'pear' not in 'Pineapple'`
- h) `'apple' > 'pineapple'`
- i) `'pineapple' < 'Peach'`

14ª) Escreva uma função `nao_eh_ruim` que receba como entrada uma string e encontre as primeiras ocorrências de `'nao eh'` e de `'ruim'`. Se `'nao'` vier antes de `'ruim'`, deve-se substituir a expressão `'nao eh...ruim'` por `'eh bom'`, retornando a nova string. Por exemplo, `nao_eh_ruim('Figado nao eh tao ruim')` deve retornar `'Figado eh bom'`.

15ª) Calcule o valor de  $\pi$  usando a série infinita:

$$\pi = 4 - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \dots$$

Construa um dicionário que guarde a quantidade de termos da série e o valor de  $\pi$  correspondente.  
Por exemplo: 1:4, 2: 2.666666666666667, 3: 3.466666666666667 , .... Quantos termos da série  
são necessários para obter 3,14? 3,141? 3,1415? 3,14159?