

INSTITUTO FEDERAL DO PIAUÍ
Curso de Tecnologia em Análise e Desenvolvimento de Sistemas
Algoritmos e Programação, Prof. Rogério Silva, ADS I – 2019.1

Atividade de Individual – "Meio de Disciplina"

Organização/Estrutura/Nomes vale de [0 a 1], exemplo 0.7, a ser multiplicado pelo valor de cada questão. Usar apenas os itens disponíveis conforme explicado nas aulas.

QUESTÕES (1 - 4 vale 1,0 & 5 - 8 vale 1,5)

1. (**primeira.txt**) Escreva (definição, exemplo, características, componentes, citar todo, etc) sobre:
- a. Instrução
 - b. Expressão
 - c. Funções
 - d. Expressão Booleana
 - e. Operadores
 - f. Recursividade

2. (**sequencia.py**) Faça um programa que calcule a soma da sequência abaixo:

Exemplo: Dado $N = 5$, por exemplo, teremos 5 elementos, $S = 1 - 2/4 + 3/9 - 5/16 + 8/25$

3. (**primo.py**) Dizemos que um inteiro positivo N é primo se possui apenas dois divisores positivos distintos. Há uma Regra R (dica) que diz que não se precisa verificar o universo de possibilidades para determinar se um número N é primo, basta verificar os números iguais ou inferiores a sua raiz quadrada. Implemente a regra R e imprima todos os primos entre 10 e 1000.

4. (**mdc.py**) Implemente o Algoritmo de Euclides para determinar o Máximo Divisor Comum entre dois números inteiro positivos.

Tabela do algoritmo de Euclides:

	1	1	1	2
24	15	9	6	3
9	6	3	0	

 = mdc(24,15)

5. (**pep8.py**) O Guia de Codificação Python, PEP8 recomenda usar para variáveis e nomes de funções o padrão **nome_da_variavel**, ou seja, separar palavras por underline(_), enquanto o estilo **CamelCase** diz para simplesmente iniciar as novas palavras com letra maiúscula

Você foi convocado para criar um programa converte de Pep-8 para o estilo camelCase

Entrada: bolo_de_milho_eh_bom_demais **Saída:** boloDeMilhoEhBomDemais

6. (**somatorio.py**) Escreva uma função recursiva que verifique todos os números perfeitos de n até m , escreva ao lado de cada número se é ou não perfeito. (Número é perfeito se for igual a soma de seus divisores positivos, exceto ele mesmo. Ex.: 6 é perfeito pois é a soma de $1 + 2 + 3$)
7. (**equacao_2grau.py**) Implemente o algoritmo da equação do segundo grau ($ax^2 + bx + c = 0$). Se

$$x = \frac{-b \pm \sqrt{\Delta}}{2.a}$$

Receba os elementos da equação, e apresente as raízes (quando for o caso) e ainda, o valor do Delta, e se a resolução tem duas raízes ($\Delta > 0$), somente uma raiz ($\Delta = 0$) ou não tem raízes reais ($\Delta < 0$)

$$\Delta = b^2 - 4.a.c$$

8. (**criptografia.py**) Elabore um algoritmo que faça a transformação de uma frase de acordo com as seguintes regras:
- a. As vogais devem ser substituída por seu valor(código) ASCII, caso seja mais de um dígito, substituir pelo somatório dos dígitos até ficar em 1 dígito.
 - b. Todas as letras devem mudar uma posição no alfabeto. Observe que o Z vira A e o a vira z
 - c. Numerais devem ser substituídos por seu valor por extenso (0 a 9) entre colchetes.
 - d. Desloque na tabela ASCII P posições (-10 até +10) as Letras de um Nome N , ambos recebidos do usuário.
 - e. Escreva a frase final
 - f. Escreva Quantas letras, Quantos Numerais, Quantos Caracteres Especiais (defina sua lista), e Quantos Demais a frase ficou após as transformações