

Problemas Inversos em Python – 2º semestre 2024
Professora: Josiele da Silva Teixeira

TRABALHO COMPUTACIONAL 1

1. Orientações Gerais sobre o trabalho e o seminário

- 1.1. O trabalho computacional consistirá em uma lista de exercício, onde cada aluno deverá implementar em Python individualmente todos os exercícios.
- 1.2. Deverá escrever um relatório técnico individual comentando todas as soluções apresentadas, onde deverá conter também prints de telas do código compilado e dos resultados obtidos.
- 1.3. A apresentação do seminário também será individual, onde cada aluno deverá estar com todos os códigos implementados em mãos (pendrive). Será sorteado aleatoriamente para cada aluno até 4 questões da lista. O aluno deverá explicar oralmente a solução e mostrar o código compilando corretamente. Durante essa etapa, questionamentos e sugestões de códigos adicionais poderão ocorrer e se necessário o aluno deverá implementar no exato momento da apresentação do código.
- 1.4. A apresentação do seminário será no dia **27/11/2024** no horário normal da aula.

2. Lista de exercícios

Todas as funções que serão implementadas nessa lista devem ser chamadas em um programa principal.

- 2.1. Escreva uma função que receba um número inteiro e retorne o número invertido. Ex.: 1234 => 4321.
- 2.2. Um número é meliante se contém a sequência 171 de dígitos. Ex.: 91713, 5617149. Escreva uma função que receba um número inteiro, identifique e informe ao usuário se esse número é meliante.
- 2.3. Um número é legal se não contém dígitos pares e se a soma dos seus dígitos for um número par. Escreva uma função que receba um número inteiro, identifique e informe ao usuário se esse número é legal.
- 2.4. Escreva uma função que receba um número inteiro e faça a divisão de todos os seus dígitos por 2 produzindo um novo número. Ex.: 4722 => 2311.
- 2.5. Escreva uma função para imprimir todos os múltiplos de 5 menores que um número inteiro positivo dado.

- 2.6. Escreva um programa que leia os elementos de uma matriz quadrada e em seguida imprima os elementos da diagonal principal.
- 2.7. Escreva um programa que leia os elementos de uma matriz e em seguida imprima a soma dos elementos de cada coluna.
- 2.8. Escreva uma função para ordenar (crescente) um vetor recebido como argumento de entrada em seguida imprima esse vetor ordenado.
- 2.9. Escreva uma função que receba um vetor e determine se a soma de todos os seus elementos é maior do que a multiplicação de seus elementos de índice par.
- 2.10. Escreva uma função recursiva para identificar se todos os dígitos de um número inteiro são iguais.
- 2.11. Escreva uma função com e sem recursividade para retornar o maior elemento de um vetor.
- 2.12. Escreva uma função com e sem recursividade para retornar o total de dígitos de um número inteiro.
- 2.13. Implemente o método de ordenação Bubble sort.
- 2.14. Implemente o método de ordenação Bucket sort.