

Lista de exercícios (5 pontos)
Entrega do trabalho no Canvas (PDF): 10/12/2025

Atenção: para desenhar as árvores, utilize o programa yEd Graph Editor.

OBS: entregue seu trabalho em formato PDF. Trabalhos em outros formatos não serão aceitos.

Parte I – Árvores binárias e AVL

1 – Para cada sequência abaixo pede-se:

- a) Desenhe a ABP
 - a. Monte uma tabela com a altura e profundidade de cada nó
 - b. Qual a ordem de impressão dos nós nos percursos pré-ordem e pós-ordem
 - b) Desenhe a árvore AVL
 - a. Monte uma tabela com a altura e profundidade de cada nó
 - b. Qual a ordem de impressão dos nós nos percursos pré-ordem e pós-ordem
- i. 1, 12, 7, 19, 4, 15, 8, 6, 10, 2, 30, 26, 17, 5, 3 e 13
 - ii. 32, 50, 63, 41, 23, 16, 27, 31, 1, 5, 11, 27, 83, 55 e 20
 - iii. 35, 30, 126, 32, 67, 79, 15, 18, 26, 29, 70, 80, 19, 21, 8, 16, 1, 50, 44 e 74
 - iv. Maria, José, João, Ana, Savio, Tiago, Nico, Taís, Davi, Tião, André
 - v. Q W E R T Y U I O P A S D F G H J K L Z X C V B N M

2 – Para cada sequência abaixo, mostre a ordem de impressão dos elementos da ABP nos percursos pré-ordem e pós-ordem após a remoção do número 23. *Obs: caso o elemento a ser removido tenha 2 filhos, use a regra da substituição do nó removido pelo nó mais à esquerda da subárvore direita.*

- i. 1, 12, 7, 19, 4, 15, 23, 6, 10, 2, 30, 26, 17, 5, 3 e 13
- ii. 42, 61, 73, 52, 23, 26, 37, 41, 12, 15, 21, 37, 93, 65 e 29
- iii. 23, 30, 126, 32, 67, 79, 15, 18, 26, 29, 70, 80, 19, 21, 8, 16, 1, 50, 44 e 74

Parte II – Hashing

1 – Assista o vídeo sobre hashing disponível [aqui](#).

2 – Agora você deve assistir o vídeo sobre implementação de hash e reproduzir toda a implementação em C# (clique [aqui](#) para ser direcionado para o vídeo). Você deve postar o programa implementado.

Parte III – Recursividade

- 1 – Faça uma função recursiva que receba uma string e retorne-a invertida.
- 2 – Faça uma função recursiva que receba uma string e verifique se é um palíndromo (lê-se igual de trás para frente).
- 3 – Faça uma função recursiva que receba um vetor de inteiros (de qualquer tamanho) e retorne o maior número de um vetor não ordenado.
- 4 – Faça uma função recursiva que calcule a soma de todos os elementos de uma lista.
- 5 – Faça uma função recursiva que converta um número decimal para sua representação binária (retornando como string ou inteiro) usando divisões sucessivas por 2.
- 6 – Faça uma função recursiva que implemente o Algoritmo de Euclides para encontrar o MDC entre dois números.
- 7 – Faça uma função recursiva que implemente a busca binária recursiva. Dado um array **ordenado** e um valor alvo, retorne o índice do valor ou -1 se não encontrado.
- 8 – Faça uma função recursiva que gere todas as permutações possíveis de uma string dada.