



Lista de Exercícios 12 (Árvores/P2)

1. Apresente teste de mesa que demonstra a execução do algoritmo para criar uma árvore de expressão para $(4 * 8) / 6 - 3$.
2. Estenda a função `build_parse_tree` para lidar com expressões matemáticas que não possuem espaços entre cada caractere.
3. Modifique as funções `build_parse_tree` e `evaluate` para lidar com instruções booleanas (`and`, `or` e `not`). Lembre-se que `not` é um operador unário, então isso vai complicar um pouco o seu código.
4. Descreva os passos para construir uma árvore de análise a partir de uma expressão matemática parentizada.
5. Qual é a utilidade de percursos em-ordem, pré-ordem e pós-ordem em árvores? Dê exemplos de aplicações.
6. Construa um programa que determine a altura de uma árvore binária.
7. Crie uma função que conte o número de nós em uma árvore binária.
8. Explique como o conceito de recursão está intrinsecamente ligado à implementação de árvores.
9. Quais são os desafios principais ao implementar árvores como estruturas de dados em linguagens de programação?
10. Crie uma função que retorne o número de folhas de uma árvore binária.