**Data Structures and Algorithms**

Logo, company name

Description automatically generated

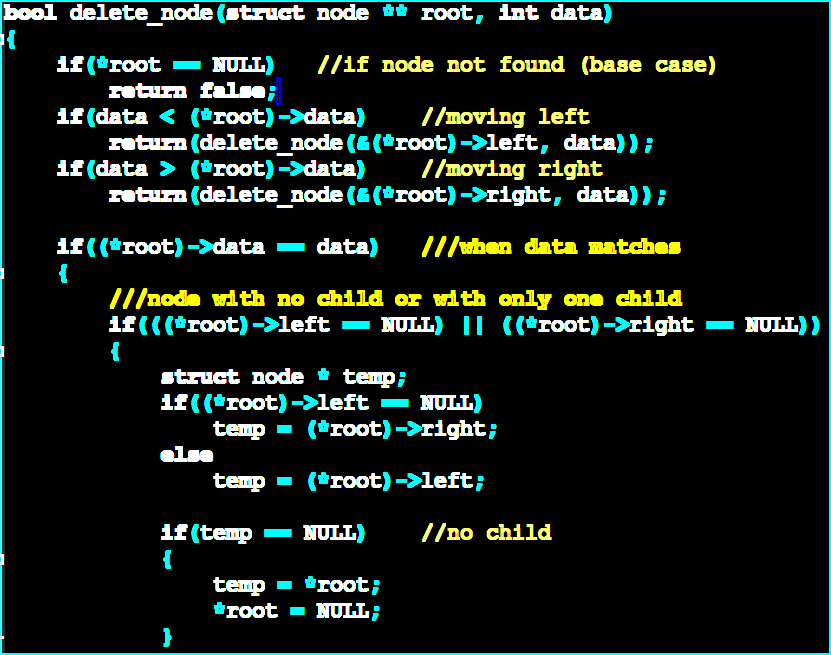
**Lab report: 9**

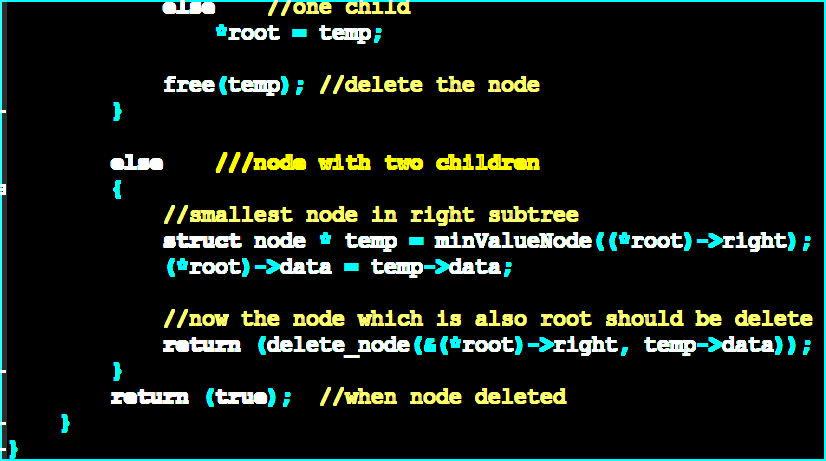
|  |  |
| --- | --- |
| **Name:** | **Ali Salman** |
| **Reg no:** | **FA22-BCE-005** |
| **Class:** | **BCE-3A** |
| **Lab Instructor:** | **Dr. Ali Mustafa** |

**Lab 10 Binary Search Tree Implementation**

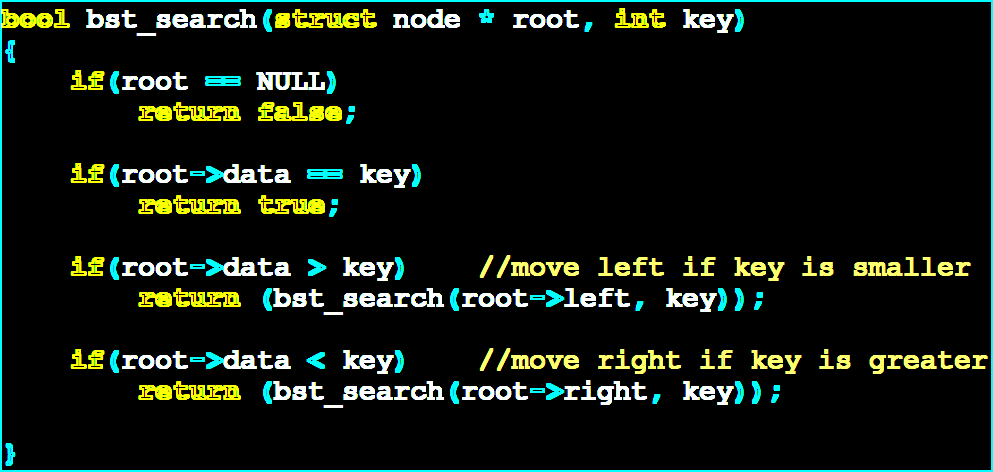
**In-Lab Task 1**

1. **Node Deletion**

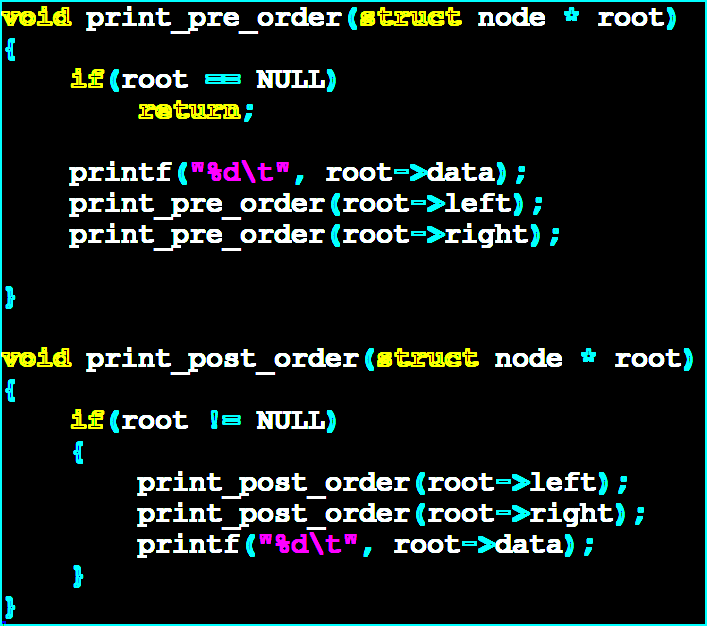




1. **Node Search:**

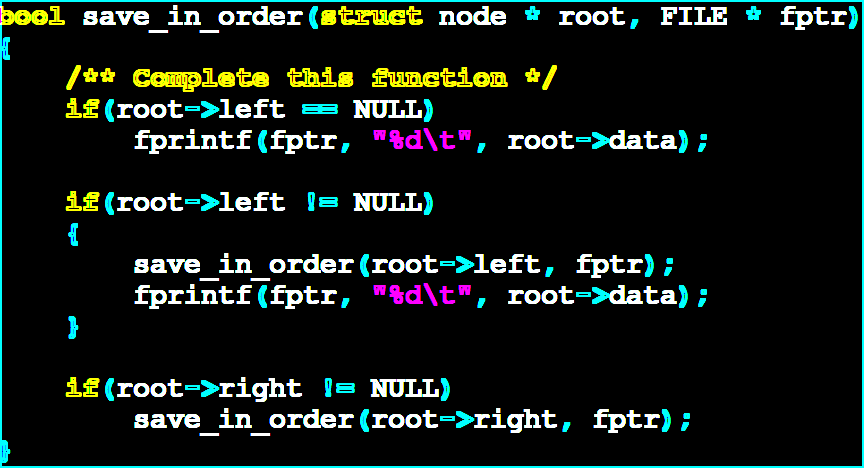


1. **Pre-Order and Post-Order printing:**

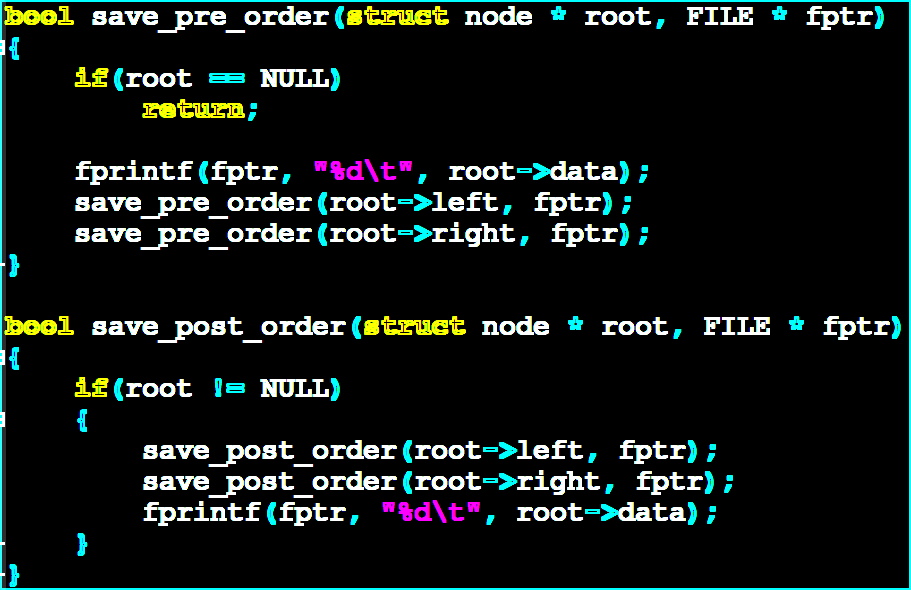


**Post Lab:**

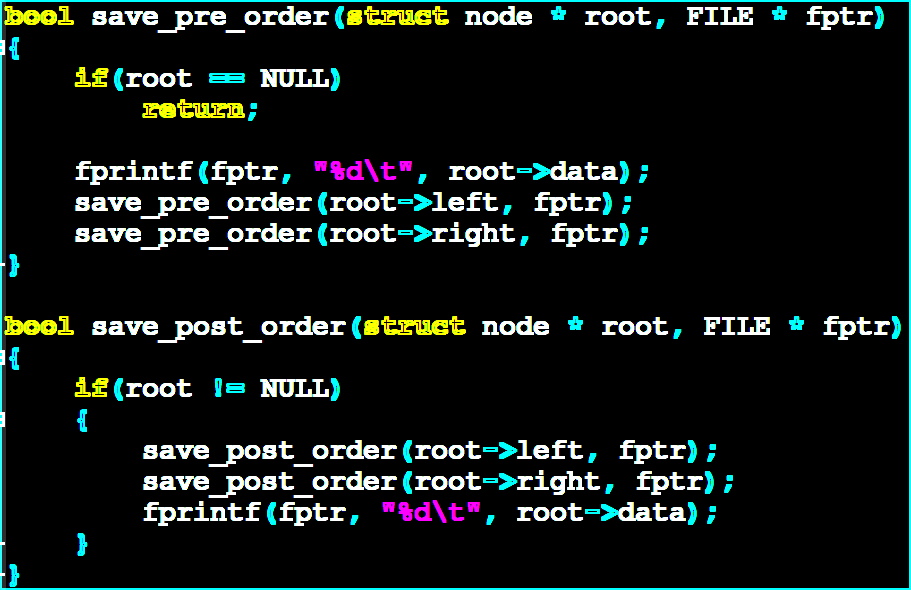
**Save in order:**



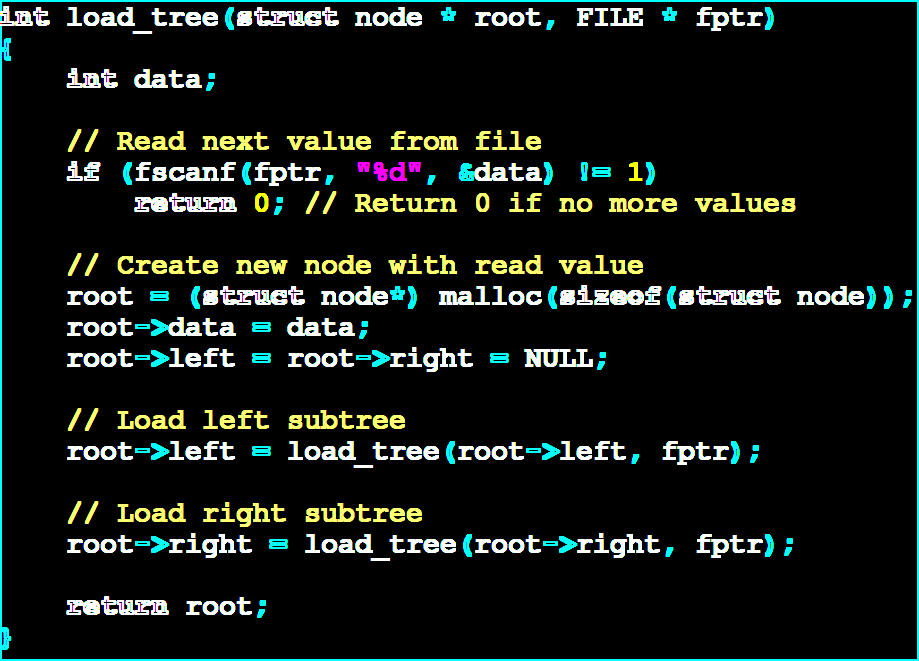
**Save Pre-Order:**



**Save Post-Order:**



**Load tree from a file**



**Critical Analysis:**

In this lab, we embarked on a journey to comprehend the intricacies of Binary Search Trees (BSTs) and the functions intricately associated with them. The primary objectives were to instill a profound understanding of BST properties, foster the development of C programs for the implementation of BSTs, and equip students with the skills to manipulate data on a hard disk using these trees. Throughout the in-lab session, we were provided with a skeleton code facilitating the creation of a BST with 10 initial nodes. The groundwork was laid with pre-implemented functions for node insertion and in-order traversal tree printing. However, the onus fell upon the students to complete critical functions, including node deletion, node search, and the extension of printing methods to include pre-order and post-order traversal. The post-lab tasks further extended the challenge, requiring students to implement functionality to save the tree data to a file, including in-order, pre-order, and post-order traversals, as well as loading a tree from a file containing numerical data. This comprehensive lab aimed not only to bolster theoretical knowledge but also to hone practical programming skills in the realm of BSTs.