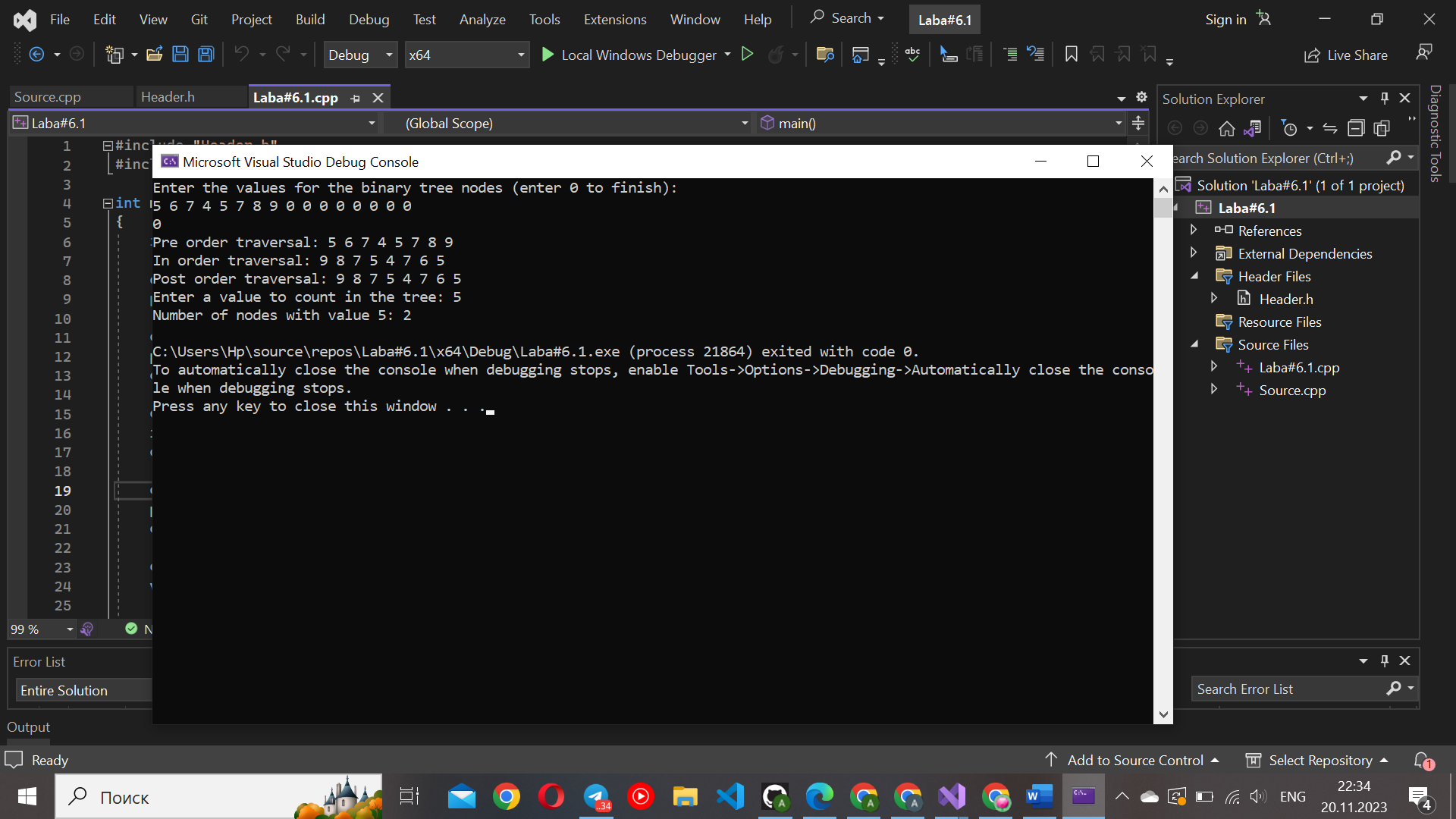
Documentation for Laba#6

Zakharchenko Anna

In this document I will describe the functionality of my code and what each function does. At first, I will show the input and output of functions in my code.



1**) createTree():**

This function is used to create a binary tree using recursive input.

It prompts the user to enter values for the nodes of the tree until the user enters 0.

It dynamically allocates memory for each node, assigns the user input to the info field, and recursively creates the left and right subtrees.

2)**preOrder(ptree root):**

This function performs a pre-order traversal of the binary tree.

It takes the root of the tree as a parameter.

It prints the info field of the current node, then recursively calls itself on the left and right subtrees.

3)**inOrder(ptree root):**

This function performs an in-order traversal of the binary tree.

It takes the root of the tree as a parameter.

It recursively calls itself on the left subtree, prints the info field of the current node, and then recursively calls itself on the right subtree.

4)**postOrder(ptree root):**

This function performs a post-order traversal of the binary tree.

It takes the root of the tree as a parameter.

It recursively calls itself on the left and right subtrees and then prints the info field of the current node.

5)**getValue():**

This function is used to get an integer value from the user.

It prompts the user to enter a value and returns the entered value.

6)**countNumOfValues(ptree root, int value):**

This function counts the number of occurrences of a specific value in the binary tree.

It takes the root of the tree and the target value as parameters.

It recursively traverses the tree and increments the count whenever a node with the specified value is encountered.

7)**deleteTree(ptree root):**

This function is used to deallocate memory and delete the entire binary tree.

It takes the root of the tree as a parameter.

It recursively calls itself to delete the left and right subtrees and then deletes the current node.