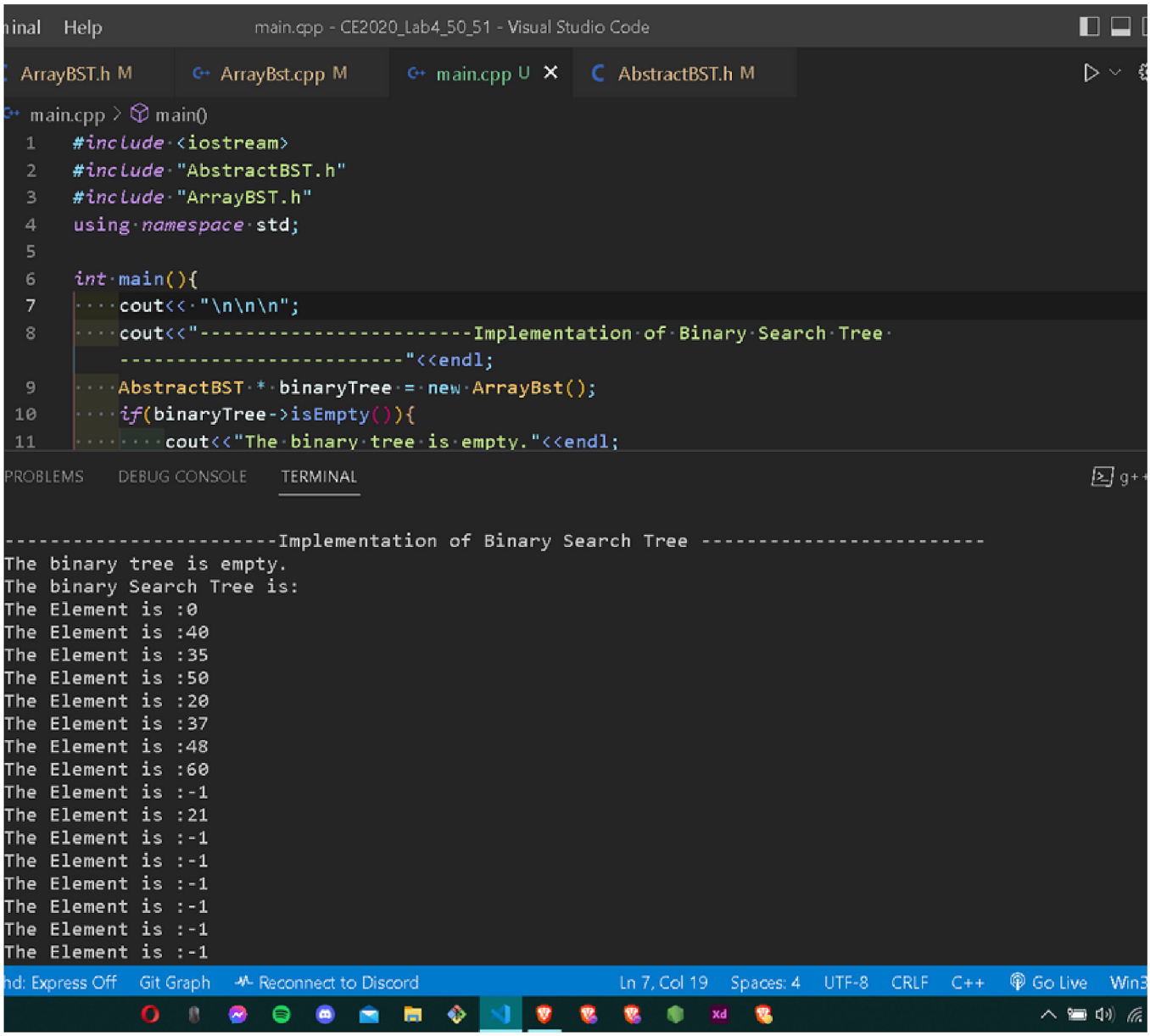
LAB 4

Implementaion of A BST

1.isEmpty()

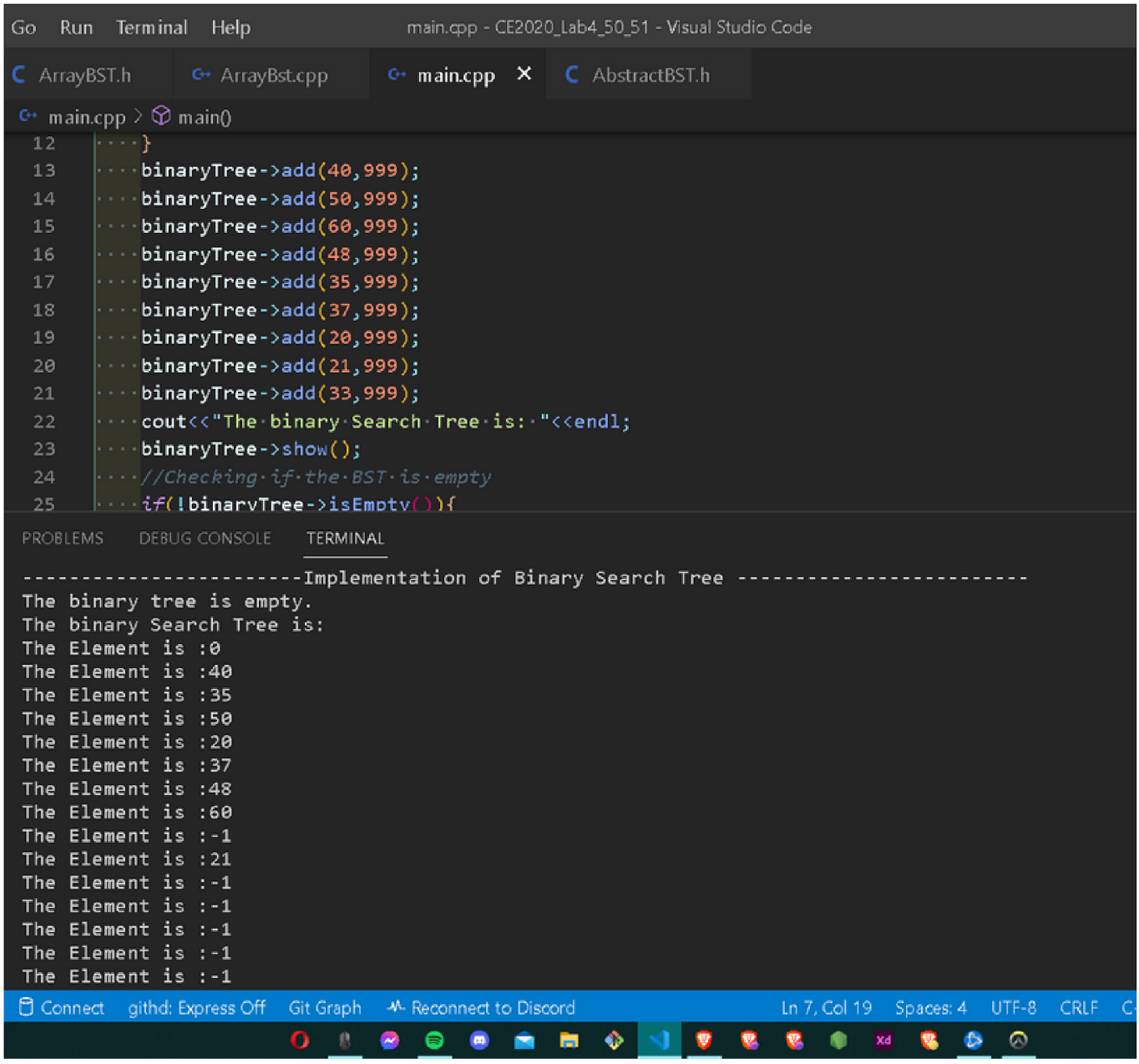
The function returns true if the search Tree is empty.
With ARRAY:
The function checks if the element at the first position is -1 (considering -1 as a null value). IF so the function returns true.



2.Add(key,value)

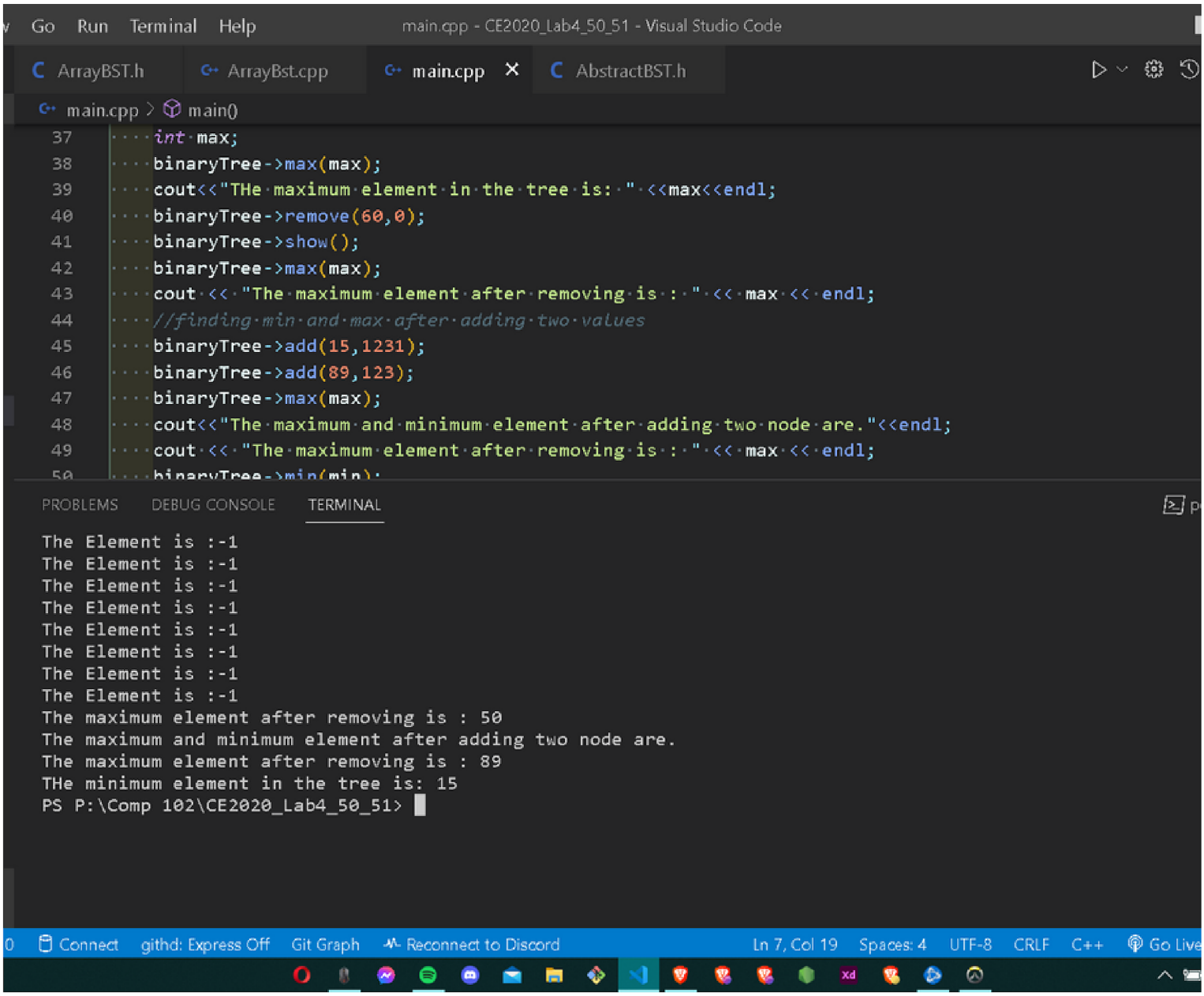
The function adds an element on the correct position in the binary search tree.

At first the function checks if the tree is empty .A while loop runs until the key is -1: If so the new node is added as the root of the tree. If the tree is not empty, the key of the node ToBeAdded is compared with the key of the current root.
if root.key > nodeToBeAdded.key :
New root is set to be the left child:
else:
new root is set to be the right child:
This process is iterated until an empty node is found.



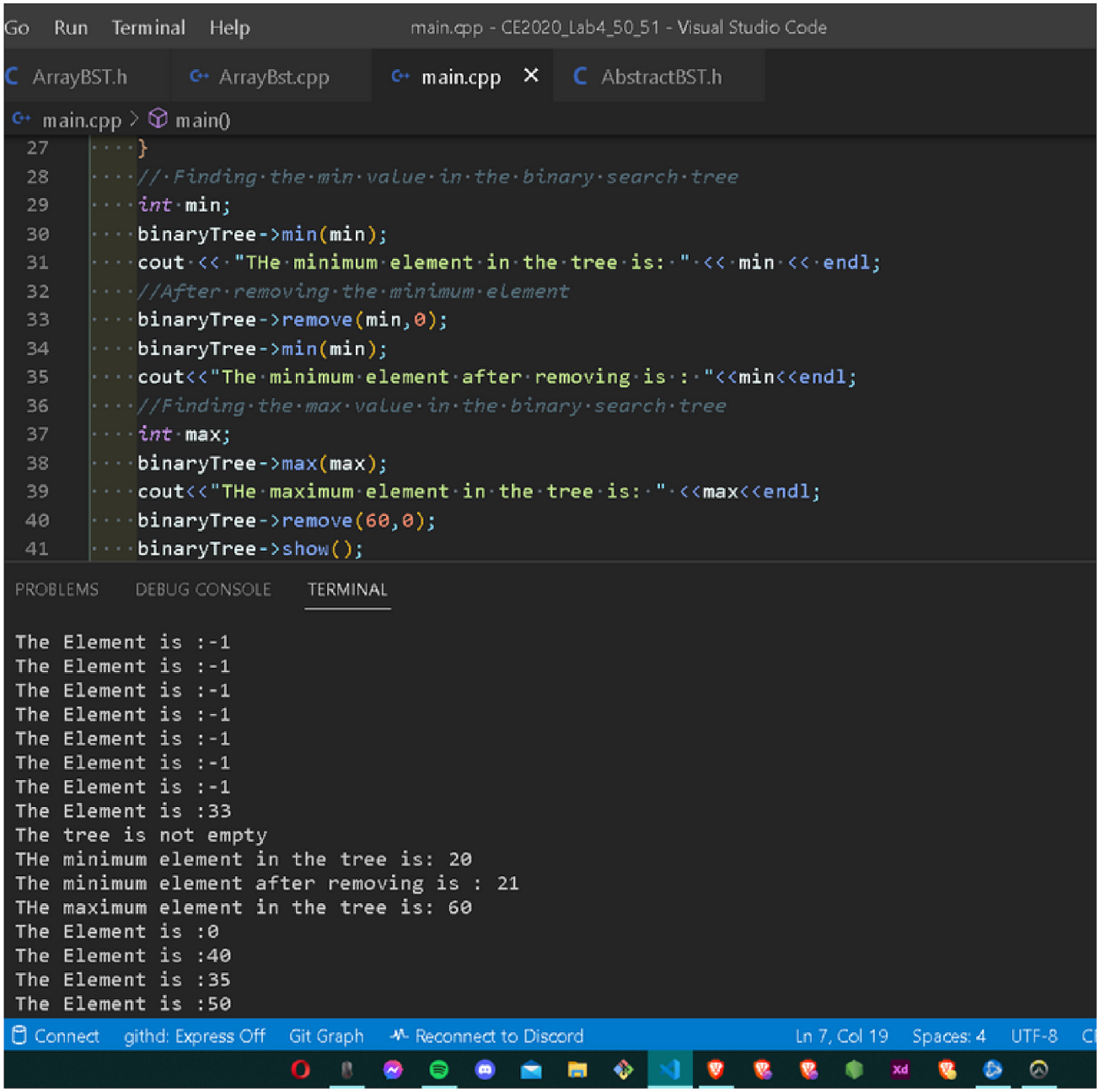
3.max()

A property of a binary search tree states that the element greater than the node is kept as the right child of the parent node. So the leftmost node is the maximum element in the binary search tree.
A for loop runs until the left most element is figured out.



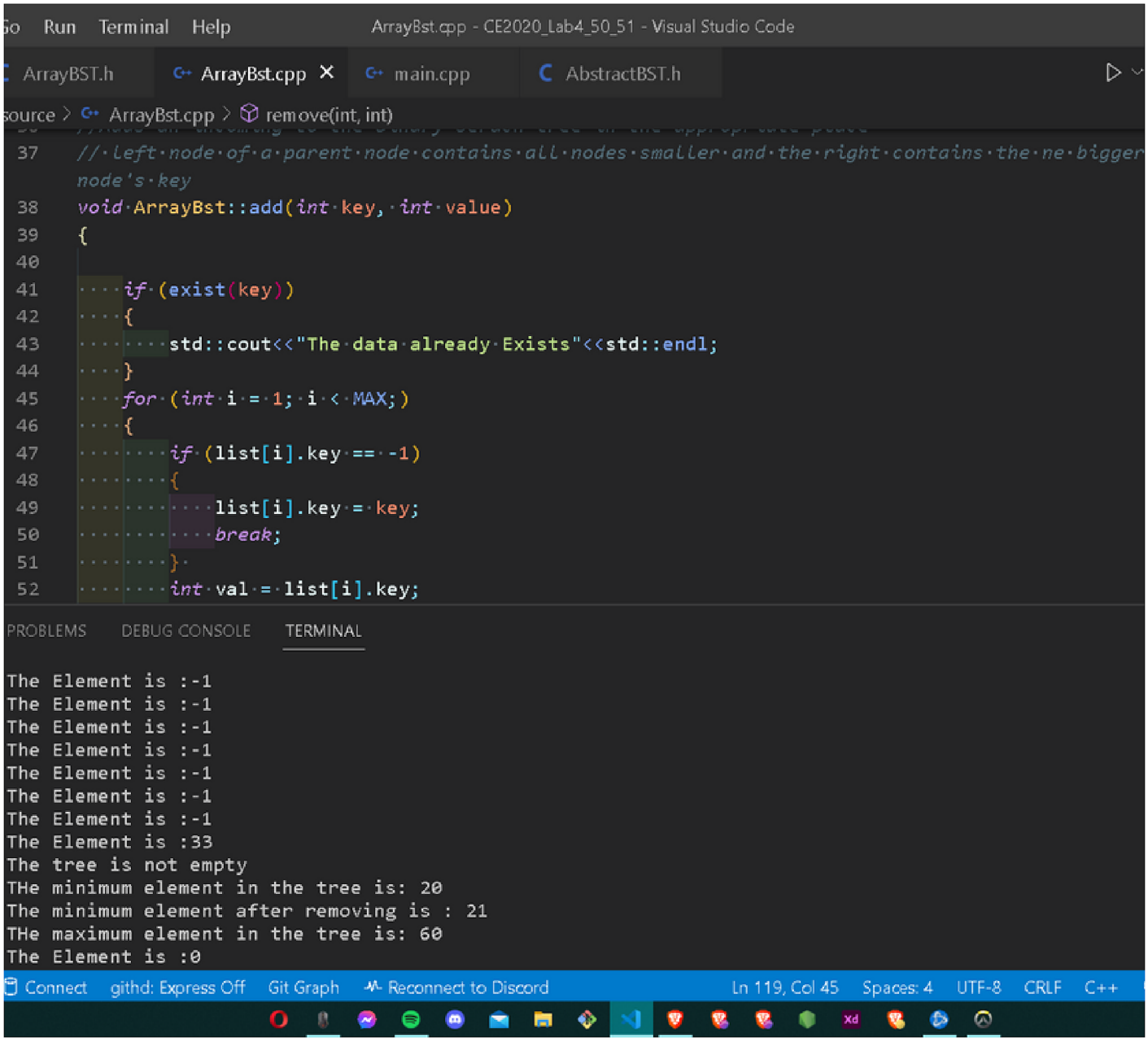
4.min()

A property of a binary search tree states that the element smaller than the node is kept as the left child of the parent node. So the rightmost node is the minimum element in the binary search tree. A for loop runs until the righ most element is figured out.



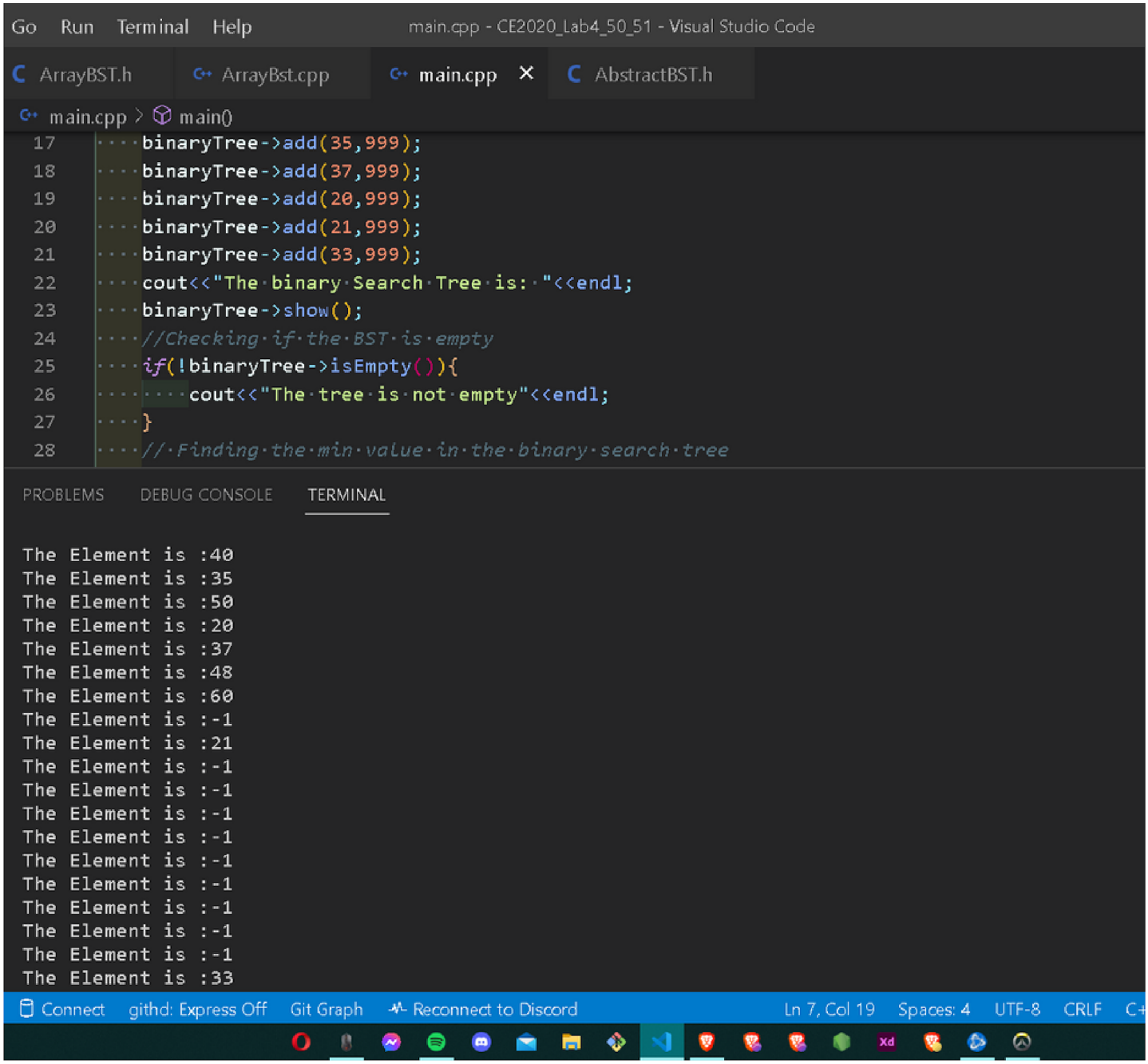
5.exist()

The exist function works similar to that of add. Only exceptions is that the key of node is also checked if it is equal to that of the root. If so true is returned , else iterates until a null node is found and false is returned in that case.



6.show()

The function iterates through each element of the array and prints it in the console. The null node is also printed in this case.



7.remove()

This function works similar to the exist function in the aspect of finding the node to be deleted.
After the node is found , node can have :
i) One child:
In this case , the node is removed and the child is kept as the new node in the parent's place. . And a function call i s made to delete child.
ii) Two children.
In this case the largest element from the left sub tree replaces the node to be deleted. And a function call i s made to delete the node with the largest element.
iii)No child:
The node is just deleted.