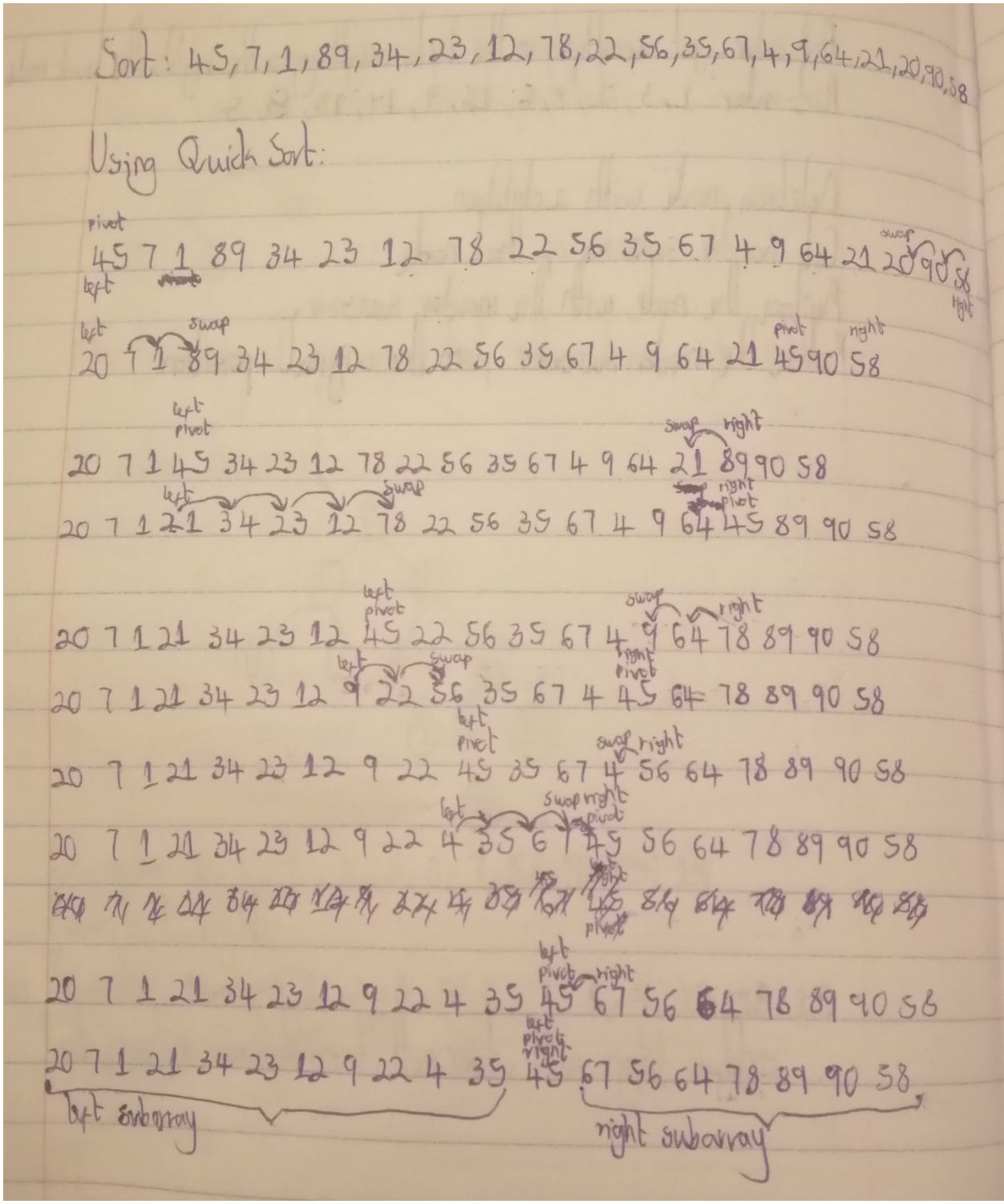
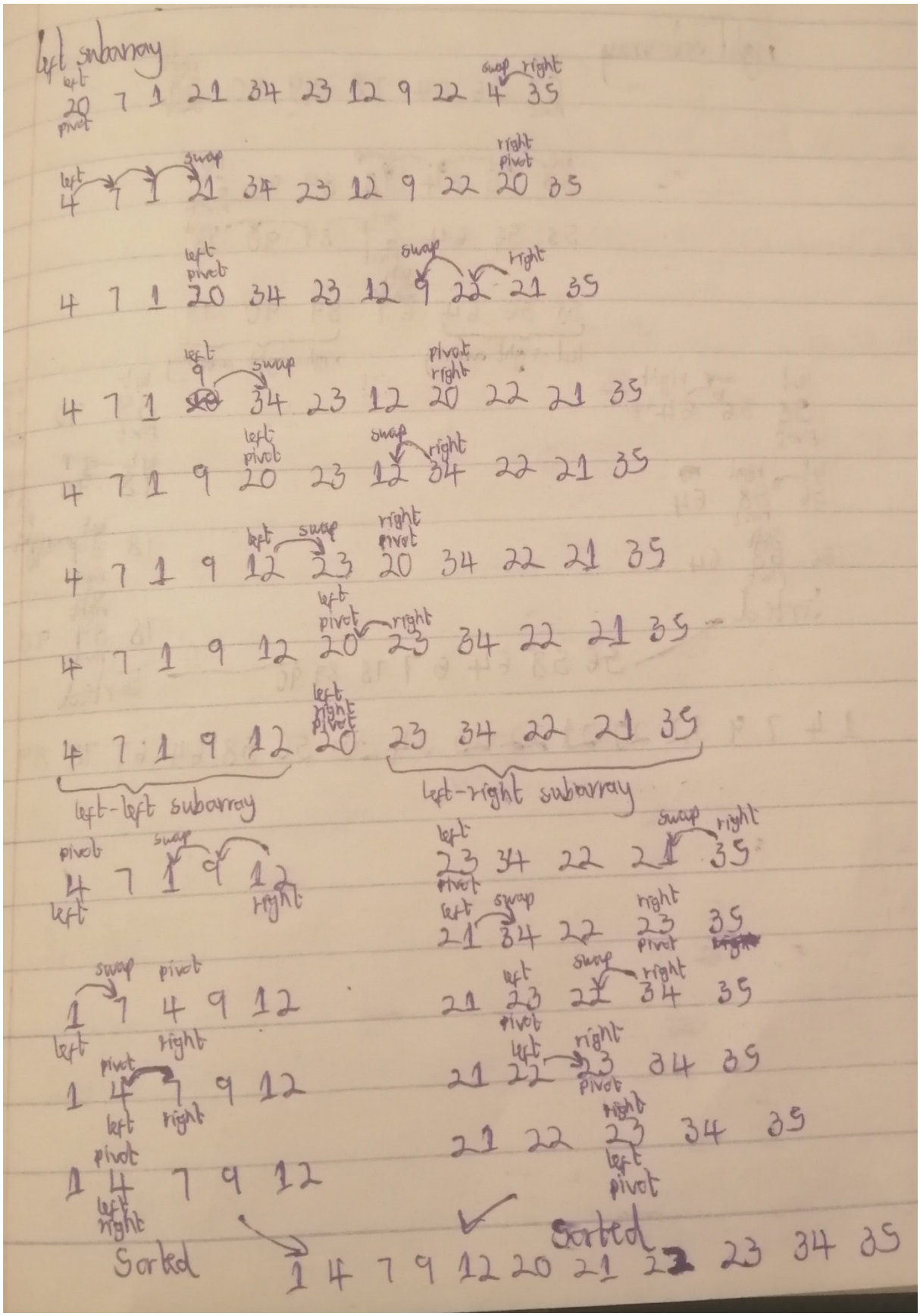
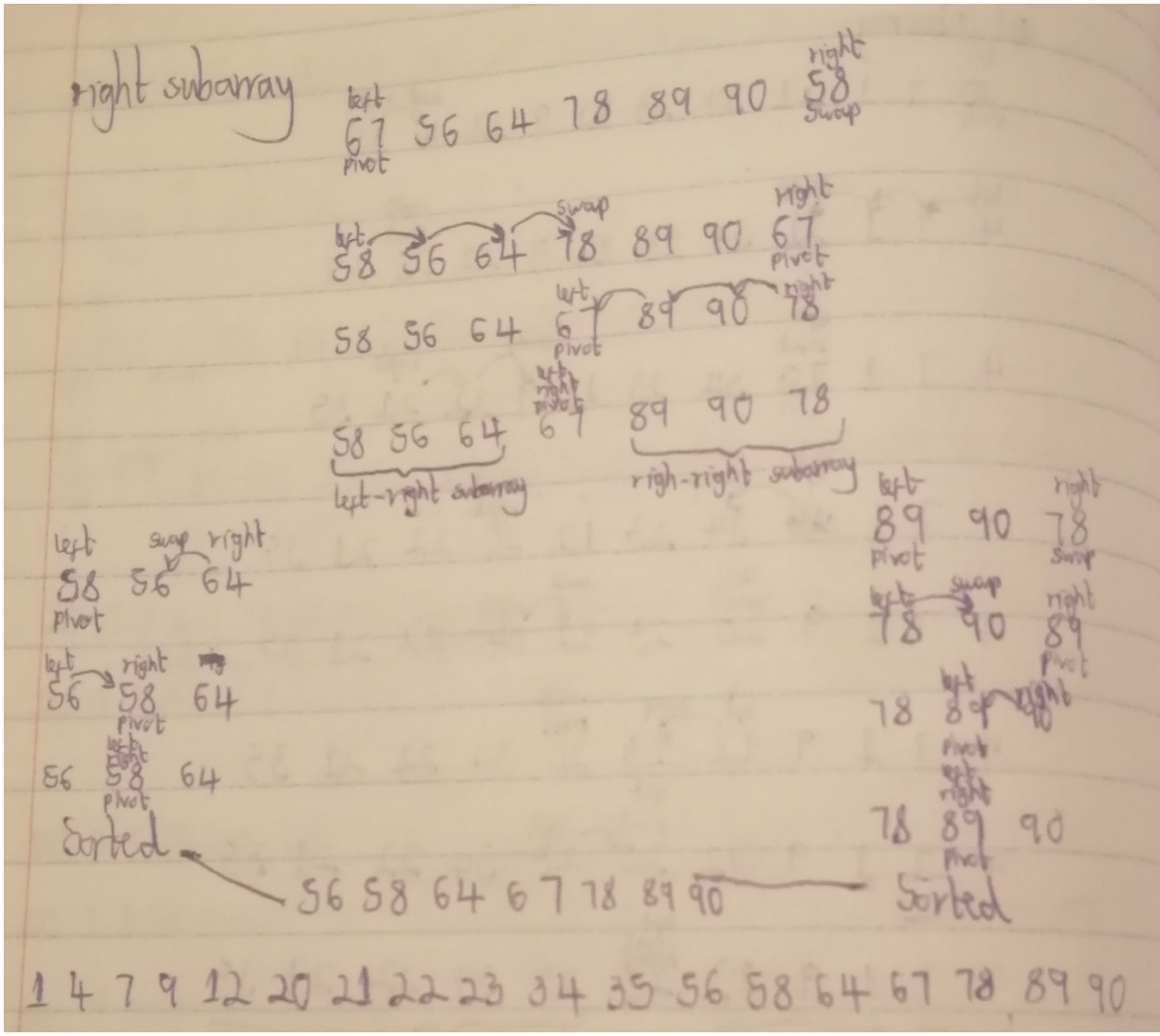
Assignment 2 (20 marks)

a) Demonstrate how to sort the following data :
45,7,1,89,34,23,12,78,22,56,35,67,4,9,64,21,20,90, 58 using the following sorting techniques

(i) Quicksort




left subarray
left
pivot
swap
right
left-left subarray
left-right subarray
Sorted
1 4 7 9 12 20 21 22 23 34 35



(ii) Mergesort

Merge sort 45, 7, 1, 89, 34, 23, 12, 78, 22, 56, 35, 67, 4, 9, 64, 21, 20, 90, 58

45, 7, 1, 89, 34, 23, 12, 78, 22, 56 | 35, 67, 4, 9, 64, 21, 20, 90, 58

45, 7, 1, 89, 34 | 23, 12, 78, 22, 56 | 35, 67, 4, 9, 64 | 21, 20, 90, 58

45, 7, 1 | 89, 34 | 23, 12, 78 | 22, 56 | 35, 67, 4 | 9, 64 | 21, 20 | 90, 58

45, 7 | 1 | 89 | 34 | 23, 12 | 78 | 22 | 56 | 35, 67 | 4 | 9 | 64 | 21 | 20 | 90 | 58

45 | 7 | 1 | 89 | 34 | 23 | 12 | 78 | 22 | 56 | 35 | 67 | 4 | 9 | 64 | 21 | 20 | 90 | 58

7, 45 | 1, 89 | 23, 34 | 12, 78 | 22, 56 | 35, 67 | 4, 9 | 21, 64 | 20, 90 | 58

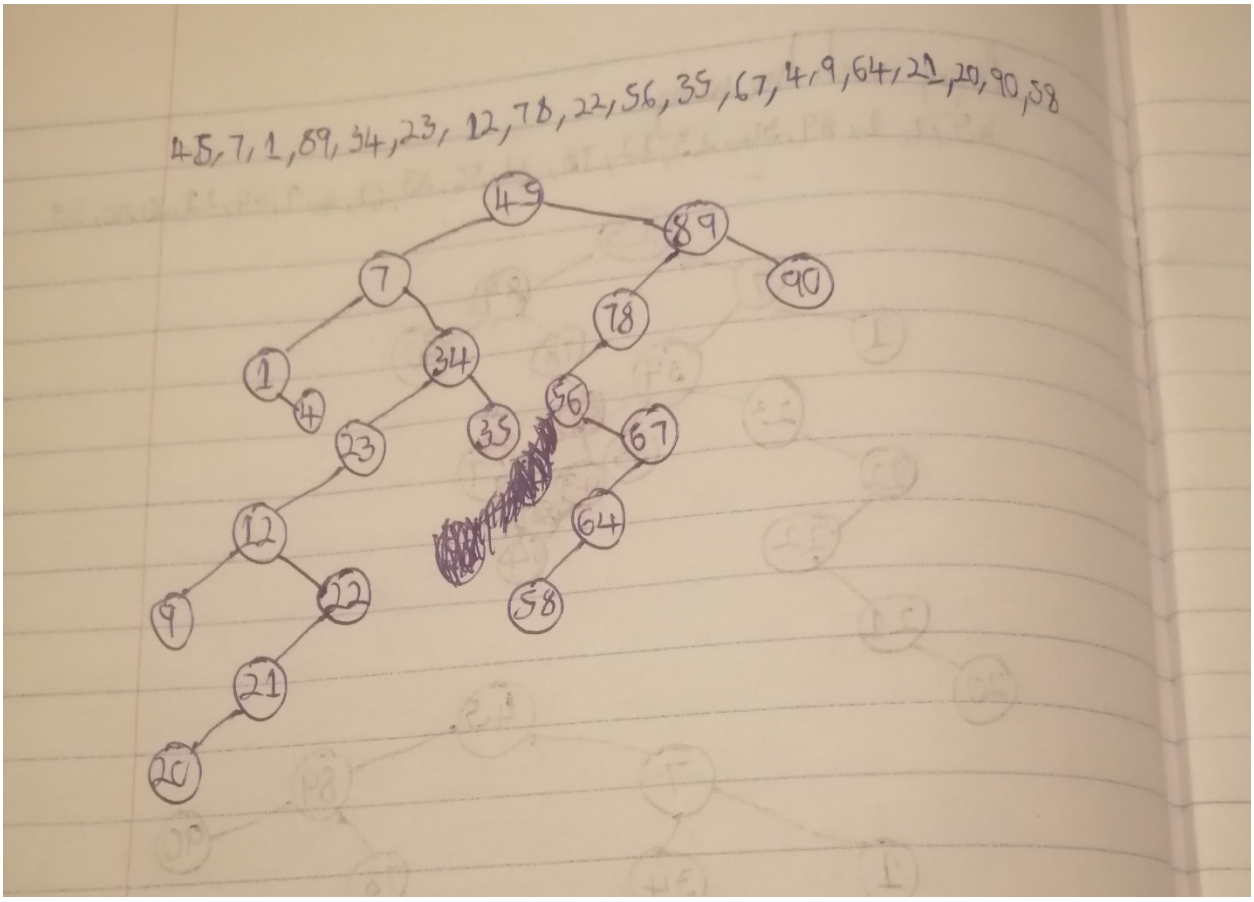
1, 7, 45, 89 | 12, 23, 34, 78 | 22, 35, 56, 67 | 4, 9, 21, 64 | 20, 58, 90

1, 7, 12, 23, 34, 45, 78 | ~~22, 35, 56, 67~~ | 4, 9, 21, 22, 35, 56, 64, 67 | 20, 58, 90

1, 4, 7, 9, 12, 21, 22, 23, 34, 35, 45, 56, 64, 67, 78 | 20, 58, 90

1, 4, 7, 9, 12, 20, 21, 22, 23, 34, 35, 45, 56, 58, 64, 67, 78, 90

b) Construct a Binary Search Tree using the data:
45,7,1,89,34,23,12,78,22,56,35,67,4,9,64,21,20,90, 58



c) Using (b) above show the data output after performing **Preorder, Inorder** and **Postorder** traversals

Preorder – 45, 7, 1, 4, 34, 23, 12, 9, 22, 21, 20, 35, 89, 78, 56, 67, 64, 58, 90

Inorder – 1, 4, 7, 9, 12, 20, 21, 22, 23, 34, 35, 45, 56, 58, 64, 67, 78, 89, 90

Postorder - 4, 1, 9, 20, 21, 22, 12, 23, 35, 34, 7, 58, 64, 67, 56, 78, 90, 89, 45

d) Using (b) above write a BST search program for searching the value **21**. The program should **automatically compute** the number of times moves were made to the left and the number of time moves were made to the right in order to locate the value **21**

```
#include <iostream>
using namespace std;
int lcount = 0, rcount = 0;

struct node
{
        int data;
        node *left, *right;
};
```

```
node *insert(node *rootnode, int value)
{
	if (rootnode==NULL)
	{
		node *newnode = new node;
		newnode->data = value;
		newnode->left = newnode->right = NULL;
		rootnode = newnode;
		return rootnode;
	}

	else if (value<rootnode->data)
	{
		rootnode->left = insert(rootnode->left, value);
	}

	else
	{
		rootnode->right = insert(rootnode->right, value);
	}
}

node *search(node *rootnode, int value)
{

	if (value==rootnode->data)
	{
		cout<<"The value was found!";
	}

	else if (value<=rootnode->data)
	{
		rootnode->left = search(rootnode->left, value);
		lcount++;
	}

	else
	{
		rootnode->right = search(rootnode->right, value);
		rcount++;
	}
}

int main (){
```

```
	node *rootnode = NULL;
	int list[19] ={45,7,1,89,34,23,12,78,22,56,35,67,4,9,64,21,20,90,58};
	cout<<"The values are: ";
	for(int i=0; i<19; i++)
	{
		cout<<list[i]<<" ";
	}

	for (int i= 0; i<19; i++)
	{
		rootnode = insert(rootnode, list[i]);
	}

	cout << "\nThe value being searched for is 21"<<endl;
	search(rootnode, 21);
	cout<<endl;
	cout<<"The No of shifts made from the left are: "<<lcount<< endl<< "The No of shifts made from the right are: "<< rcount;
}
```