LAB 10

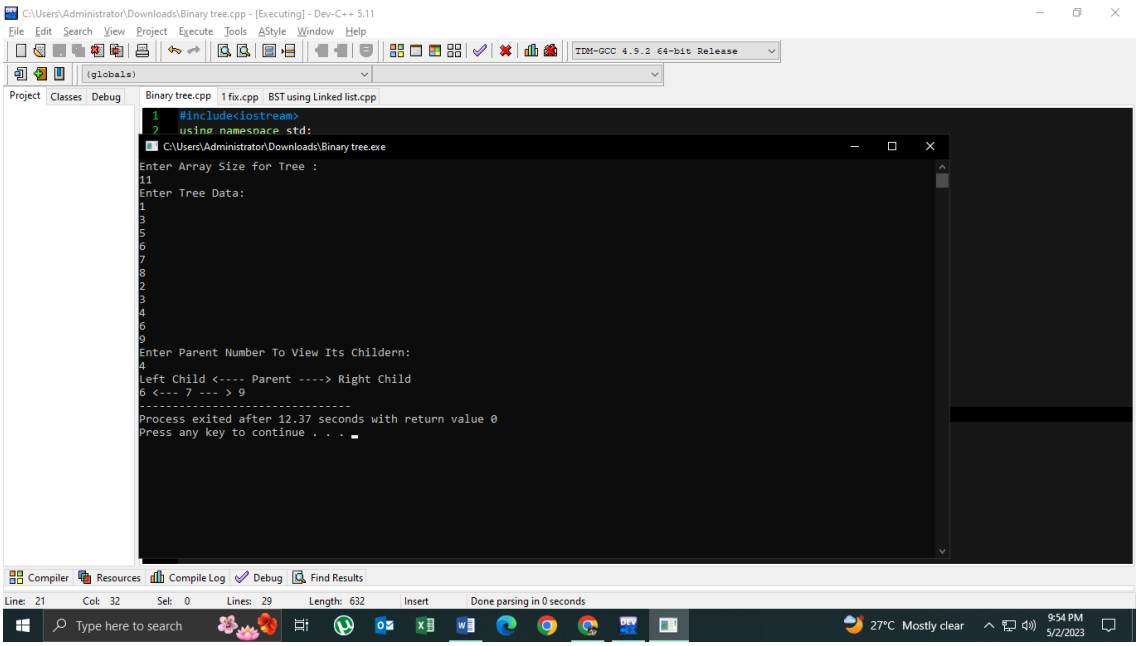
M.HUZAIFA MUSTAFA
SECTION AM
SP22-BSCS-0046

1. Binary Search Tree.

SOURCE CODE:

```
#include<iostream>
using namespace std;

int main(){
	int arr_size;
	cout<<"Enter Array Size for Tree : "<<endl;
	cin>>arr_size;

	int arr[arr_size];
	cout<<"Enter Tree Data: "<<endl;
	for(int i=0; i < arr_size; i++){
		cin>>arr[i];
	}

	int parentNumber;
	cout<<"Enter Parent Number To View Its Childern: "<<endl;
	cin>>parentNumber;

	if(parentNumber < (arr_size-1)/2){

		cout<<"Left Child <---- Parent ----> Right Child\n";
		cout<<arr[(parentNumber*2)+1]<<" <--- "<<arr[parentNumber]<<" --- >
"<<arr[(parentNumber*2)+2];
  }
  else{
		cout<<"Invalid Parents . "<<endl;
  }
	return 0;
}
```

PICTURE:



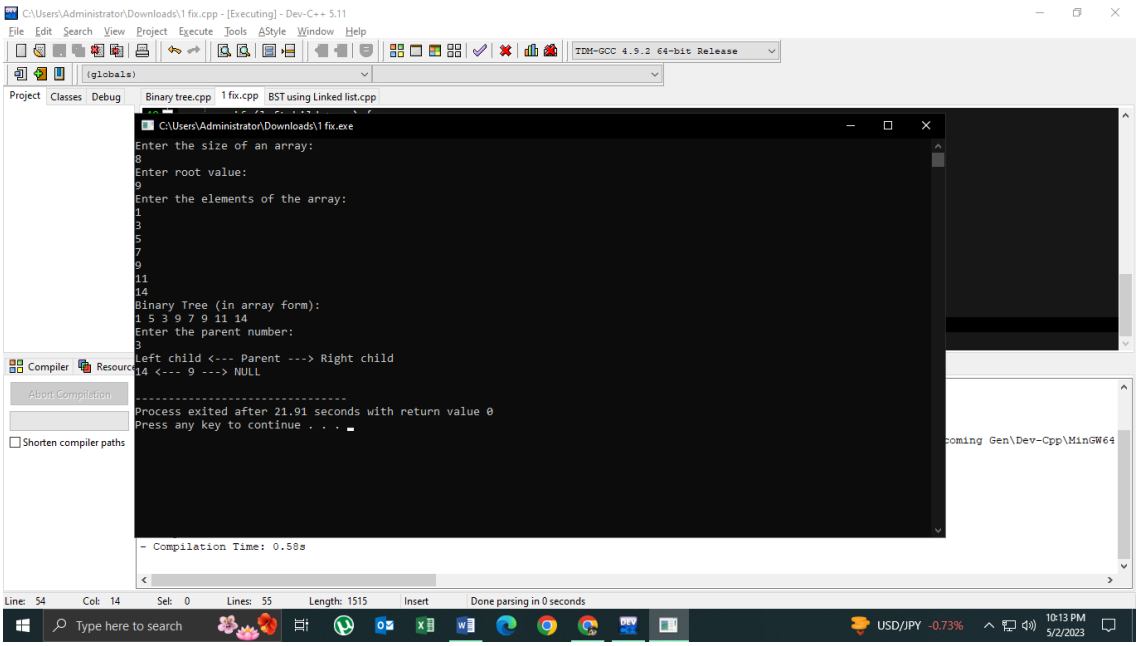
2. Binary Search Tree using array.

SOURCE CODE:

```
#include <iostream>
using namespace std;

int main() {
   int n, root;
   cout << "Enter the size of an array: "<<endl;
   cin >> n;

   int arr[n];
   cout << "Enter root value: "<<endl;
   cin >> root;
   arr[0] = root;

   cout << "Enter the elements of the array: "<<endl;
   for (int i = 1; i < n; i++) {
      cin >> arr[i];
      int parent = (i - 1) / 2;
      if (arr[i] < arr[parent]) {
         // swap parent and child
         int temp = arr[i];
         arr[i] = arr[parent];
         arr[parent] = temp;
      }
   }
```

```
    cout << "Binary Tree (in array form):" << endl;
    for (int i = 0; i < n; i++) {
        cout << arr[i] << " ";
    }
    cout << endl;

    int parentnumber;
    cout << "Enter the parent number: "<<endl;
    cin >> parentnumber;
    if (parentnumber >= n || parentnumber < 0) {
        cout << "Invalid parent number" << endl;
    } else {
        int leftchild = 2 * parentnumber + 1;
        int rightchild = 2 * parentnumber + 2;
        if (leftchild >= n) {
            cout << "Parent has no left child" << endl;
        } else {
            cout << "Left child <--- Parent ---> Right child" << endl;
            cout << arr[leftchild] << " <--- " << arr[parentnumber] << " ---> ";
            if (rightchild >= n) {
                cout << "NULL";
            } else {
                cout << arr[rightchild];
            }
            cout << endl;
        }
    }

    return 0;
}
```

PICTURE:



3. Binary Search Tree using Linked-list.

SOURCE CODE:

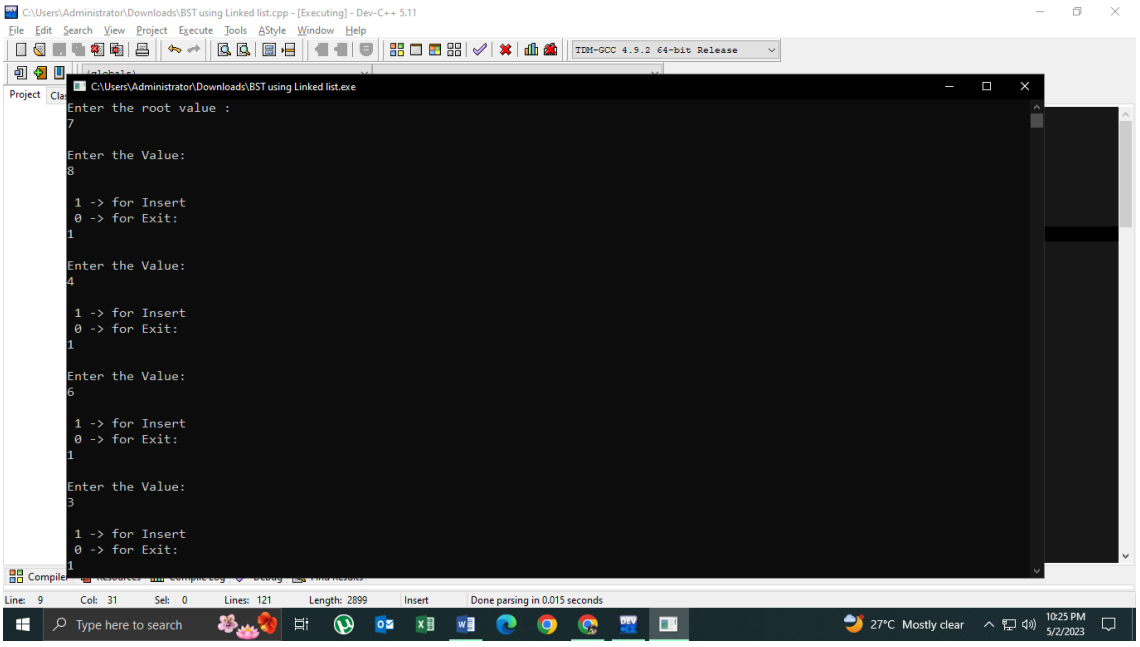
```
#include<iostream>
using namespace std;
class Node{
   public:
      int data;
      Node *left;
      Node *right;
      Node(int data){
         this->data = data;
         this->left = NULL;
         this->right = NULL;
      }
};
class tree{
   public:
      Node *head;
      tree(){
         this->head = NULL;
      }
      void insertNode(Node *node, int value){
         if(value < node->data){
           if(node->left==NULL){
                    Node *newNode = new Node(value);
                    node->left = newNode;
```

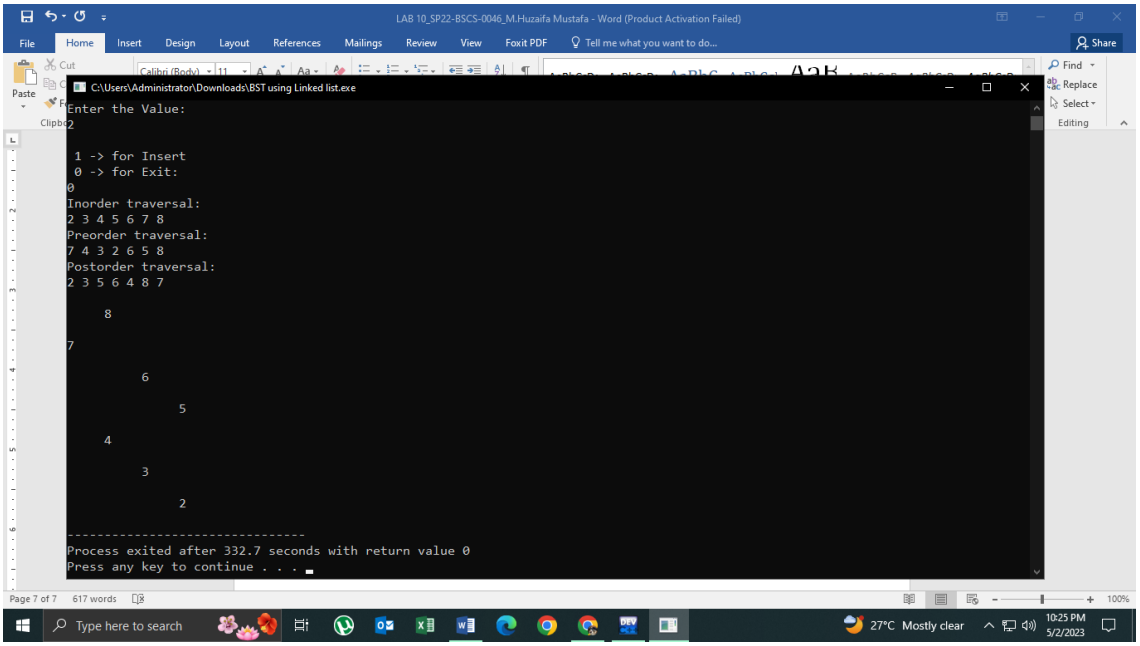
```
                }
                else{
                    insertNode(node->left,value);
                }
            }
            else{
                if(node->right==NULL){
                    Node *newNode = new Node(value);
                    node->right = newNode;
                }
                else{
                    insertNode(node->right,value);
                }
            }
        }
        void inorderTraversal(Node *temp){
            if(temp!=NULL){
                inorderTraversal(temp->left);
                cout << temp->data << " ";
                inorderTraversal(temp->right);
            }
        }
        void preorderTraversal(Node *temp){
            if(temp != NULL){
                cout<<temp->data<<" ";
                preorderTraversal(temp->left);
                preorderTraversal(temp->right);
            }
        }
        void postorderTraversal(Node *temp){
            if(temp!=NULL){
            postorderTraversal(temp->left);
            postorderTraversal(temp->right);
            cout << temp->data << " ";
        }
        }
        void print2DUtil(Node* root, int space)
{
    int COUNT = 5;
  // Base case
  if (root == NULL)
    return;
```

```
        // Increase distance between levels
        space += COUNT;

        // Process right child first
        print2DUtil(root->right, space);

        // Print current node after space
        // count
        cout << endl;
        for (int i = COUNT; i < space; i++)
           cout << " ";
        cout << root->data << "\n";

        // Process left child
        print2DUtil(root->left, space);
    }

    // Wrapper over print2DUtil()
    void print2D(Node* root)
    {
        // Pass initial space count as 0
        print2DUtil(root, 0);
    }
        };
int main(){
        tree newtree;
        int rootvalue;
        cout << "Enter the root value : " << endl;
        cin >> rootvalue;
        Node*root = new Node(rootvalue);
        int value;
        int i = 1;
        while(i == 1){
                cout<<"\n";
                cout<<"Enter the Value: "<<endl;
                cin>>value;
                newtree.insertNode(root, value);
                cout<<"\n";
                cout<<" 1 -> for Insert\n 0 -> for Exit: "<<endl;
                cin>>i;
        }
```

```
        cout << "inorder traversal: "<<endl;
        newtree.inorderTraversal(root);
        cout << endl;
        cout << "preorder traversal: "<<endl;
        newtree.preorderTraversal(root);
        cout << endl;
        cout << "postorder traversal: "<<endl;
        newtree.postorderTraversal(root);
        cout << endl;
        newtree.print2D(root);
        return 0;
}
```

PICTURE:




LAB 10_SP22-BSCS-0046_M.Huzaifa Mustafa - Word (Product Activation Failed)
C:\Users\Administrator\Downloads\BST using Linked list.exe
Enter the Value:
2
1 -> for Insert
0 -> for Exit:
0
Inorder traversal:
2 3 4 5 6 7 8
Preorder traversal:
7 4 3 2 6 5 8
Postorder traversal:
2 3 5 6 4 8 7
8
7
6
5
4
3
2
Process exited after 332.7 seconds with return value 0
Press any key to continue . . .
Page 7 of 7
617 words
Type here to search
27°C Mostly clear
10:25 PM
5/2/2023