**BINARY TREES**

**Aim:**To implement a binary tree.

**Theory:** A **tree** is a popular data structure that is non-linear in nature. Unlike other data structures like an array, stack, queue, and linked list which are linear in nature, a tree represents a hierarchical structure. The ordering information of a tree is not important. A tree contains nodes and 2 pointers. These two pointers are the left child and the right child of the parent node. Let us understand the terms of tree in detail.

* **Root:**The root of a tree is the topmost node of the tree that has no parent node. There is only one root node in every tree.
* **Parent Node:**  The node which is a predecessor of a node is called the parent node of that node.
* **Child Node:** The node which is the immediate successor of a node is called the child node of that node.
* **Sibling:** Children of the same parent node are called siblings.
* **Edge:**Edge acts as a link between the parent node and the child node.
* **Leaf:**A node that has no child is known as the leaf node. It is the last node of the tree. There can be multiple leaf nodes in a tree.
* **Subtree:**The subtree of a node is the tree considering that particular node as the root node.
* **Depth:** The depth of the node is the distance from the root node to that particular node.
* **Height:**The height of the node is the distance from that node to the deepest node of that subtree.
* **Height of tree:** The Height of the tree is the maximum height of any node. This is the same as the height of the root node.
* **Level:**A level is the number of parent nodes corresponding to a given node of the tree.
* **Degree of node:**  The degree of a node is the number of its children.
* **NULL:** The number of NULL nodes in a binary tree is (N+1), where N is the number of nodes in a binary tree.

*A binary tree is a tree data structure in which each node can have at most two children, which are referred to as the left child and the right child. The topmost node in a binary tree is called the root, and the bottom-most nodes are called leaves. A binary tree can be visualized as a hierarchical structure with the root at the top and the leaves at the bottom.*

*Binary trees have many applications in computer science, including data storage and retrieval, expression evaluation, network routing, and game AI. They can also be used to implement various algorithms such as searching, sorting, and graph algorithms.*



**Program:**

#include<stdio.h>

#include<stdlib.h> //preprocessor directives

struct node{ //creating a node of a tree

int data;

struct node \*left;

struct node \*right; //left and right pointers

};

void traversal(struct node \*root){ //traversal function

if(root==NULL){

return ;

}

else{

traversal(root->left); //traverse root's left and prints value

printf("%d->",root->data);

traversal(root->right); //same for right

}

}

struct node \* create() //function to create

{

int x;

struct node \*newnode;

newnode=(struct node \*)malloc(sizeof(struct node)); //allocating memory for node

printf("enter data(-1 for no node):");

scanf("%d",&x);

if(x==-1)

{

return 0;

}

else

{

newnode->data=x; //data part has value

printf("enter left child of %d",x);

newnode->left=create(); // creates left side

printf("enter right child of %d",x);

newnode->right=create(); //creation of right side of tree

return newnode;

}

}

int main(){

struct node \*root; //forming root pointer

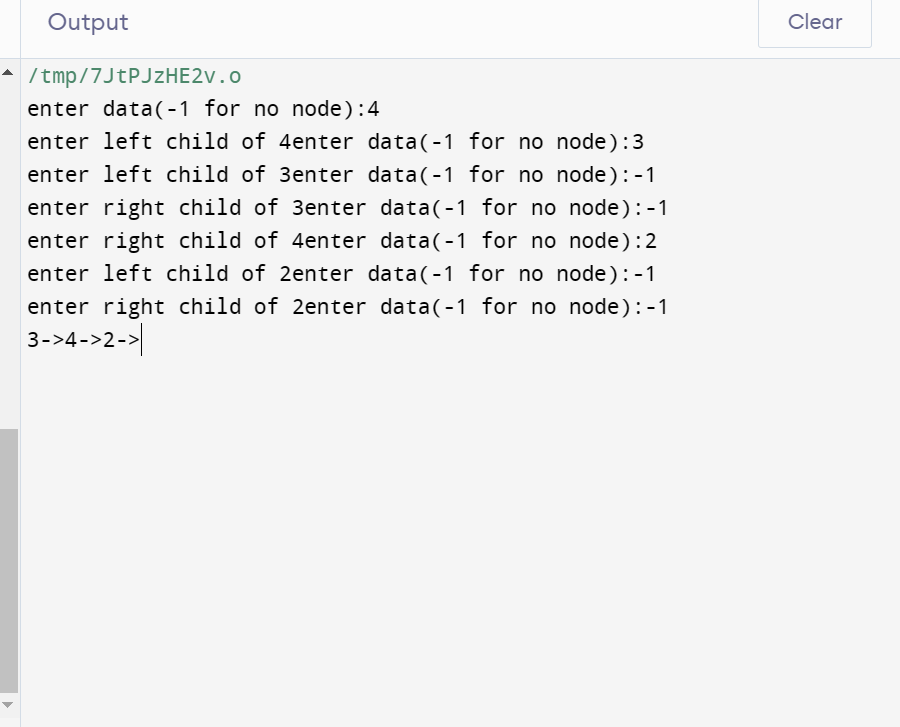
root=0;

root=create(); //function call for root

traversal(root); //function call for traversal

}

**Output:**

****