RVCE24MCA095

Binary Tree

Shreya.P. Negur.

* To perform preorder traversal in binary tree, by visiting nodes root, left subtree and right subtree.

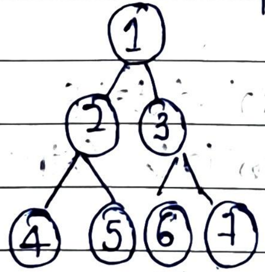
*. Algorithm :

```
Step 1:  Create a binary tree.
Step 2:  Create a function createNode(data) to allocate
         memory for a new node.
          Set left and right pointers to NULL.
Step 8:  Implement preorderTraversal(root):
            if the root is Null then return,
            Recursively calling preorderTraversal(root → left).
            Recursively calling preorderTraversal(root → right)
Step 4:   Print the traversal result.
```

* Program :

```
#include<stdio.h>
#include<stdlib.h>
struct Node{
        int data;
        struct Node* left;
        struct Node* right;
};
struct Node *  createNode (int data){
            struct  Node*  newNode = (struct Node *) malloc (sizeof
            (struct Node));
            newNode → data = data;
            newNode → ~~data~~ left = Null;
        newNode -> right = NULL;
        return newNode;
}
void preorderTraversal (struct Node* root) {
        if (root == NULL)
        {
            return;
        }
        printf ("%d", root -> data);
        preorderTraversal (root -> left);
        preorderTraversal (root -> right);
}
int main() {
    struct Node* root = createNode(1);
    root -> left = createNode(2);
    root -> right = createNode(3);
    root -> left -> left = createNode(4);
    root -> left -> right = createNode(5);
    root -> right -> left = createNode(6);
    root -> right -> right = createNode(7);
    printf( "Preorder Traversal of Binary Tree: ");
    preorderTraversal (root);
    printf ("\n");
    return 0;
}
```

Output:- Preorder Traversal of Binary Tree: 1 2 4 5 3 6 7



Preorder traversal follows the order root -> left -> right.