

IT DS 201 LAB

SUBMITTED BY ADITYA SINGH 2K19/EP/005

Program 16 : Write a program to implement the Binary Tree using linked list and perform In-order traversal.

CODE

```
void convertList2Binary(Node *head, TreeNode* &root){

    queue<TreeNode *> q;
    if (head == NULL){
        root = NULL;
        return;
    }

    root = newTreeNode(head->data);
    q.push(root);
    head = head->next;
    while (head){
        TreeNode* parent = q.front();
        q.pop();
        TreeNode *leftChild = NULL, *rightChild = NULL;
        leftChild = newTreeNode(head->data);
        q.push(leftChild);
        head = head->next;
        if (head){
            rightChild = newTreeNode(head->data);
            q.push(rightChild);
            head = head->next;
        }
        parent->left = leftChild;
        parent->right = rightChild;
    }
}
```

```

void printList(Node* n){
    while (n != NULL) {
        cout << n->data << " ";
        n = n->next;
    }
}

void inorderTraversal(TreeNode* root){
    if (root)
    {
        inorderTraversal( root->left );
        cout << root->data << " ";
        inorderTraversal( root->right );
    }
}

```

ALGORITHM

At every step, we take the parent node from the queue, make the next two nodes of the linked list as children of the parent node, and enqueue the next two nodes to the queue.

1. Create an empty queue.
2. Make the first node of the list as root, and enqueue it to the queue.
3. Until we reach the end of the list, do the following.
 - a) Dequeue one node from the queue. This is the current parent.
 - b) Traverse two nodes in the list, add them as children of the current parent.
 - c) Enqueue the two nodes into the queue.

INPUT/OUTPUT

```
int main(){  
  
    struct Node* head = NULL;  
    push(&head, 36);  
    push(&head, 30);  
    push(&head, 25);  
    push(&head, 15);  
    push(&head, 12);  
    push(&head, 10);  
  
    cout << "Linked List is: \n";  
    printList(head);  
  
    TreeNode *root;  
    convertList2Binary(head, root);  
  
    cout << "\nInorder Traversal of the constructed Binary Tree is: \n";  
    inorderTraversal(root);  
    return 0;  
}
```

```
Linked List is:  
10 12 15 25 30 36  
Inorder Traversal of the constructed Binary Tree is:  
25 12 30 10 36 15 [Finished in 753ms]
```

END