Name – Shrinivas Hatyalikar
Div - CS-B
Roll no.- 24
PRN- 12110883

WAP to implement TBT and Perform different Traversals on it without using a Stack.

Code:

```
#include <stdio.h>
#include <stdlib.h>

typedef struct Node {
   int lbit, rbit, value;
   struct Node *left, *right;
} Node;

Node *createNode(int data) {
   Node *node = (Node *)malloc(sizeof(Node));
   node->value = data;
   node->lbit = node->rbit = 0;
   node->left = node->right = NULL;
   return node;
}

void insert(Node **root, int data) {
   if (*root == NULL) {
      *root = createNode(data);
      return;
   }
   if (data < (*root)->value) {
      if ((*root)->lbit == 0) {
         Node *node = createNode(data);
         node->left = (*root)->left;
         node->right = *root;
         (*root)->lbit = 1;
         (*root)->left = node;
      } else {
         insert(&((*root)->left), data);
      }
   } else {
```

```
        if ((*root)->rbit == 0) {
            Node *node = createNode(data);
            node->right = (*root)->right;
            node->left = *root;
            (*root)->rbit = 1;
            (*root)->right = node;
        } else {
            insert(&((*root)->right), data);
        }
    }
}

Node *inorderSuccessor(Node *node) {
    if (node->rbit == 0) {
        return node->right;
    } else {
        node = node->right;
        while (node->lbit == 1) {
            node = node->left;
        }
        return node;
    }
}

void inorderTraversal(Node *root) {
    Node *current = root;
    while (current->lbit == 1) {
        current = current->left;
    }
    while (current != NULL) {
        printf("%d ", current->value);
        current = inorderSuccessor(current);
    }
}

void preorderTraversal(Node *root) {
    if (root != NULL) {
        printf("%d ", root->value);
        if (root->lbit == 1) {
            preorderTraversal(root->left);
        }
        if (root->rbit == 1) {
            preorderTraversal(root->right);
        }
```

```
    }
}

void postorderTraversal(Node *root) {
    if (root != NULL) {
        if (root->lbit == 1) {
            postorderTraversal(root->left);
        }
        if (root->rbit == 1) {
            postorderTraversal(root->right);
        }
        printf("%d ", root->value);
    }
}

int main() {
    Node *root = NULL;
    insert(&root, 10);
    insert(&root, 15);
    insert(&root, 12);
    insert(&root, 51);
    insert(&root, 90);
    insert(&root, 13);
    insert(&root, 17);
    insert(&root, 2);
    insert(&root, 1);

    printf("Inorder Traversal: ");
    inorderTraversal(root);
    printf("\n");

    printf("Preorder Traversal: ");
    preorderTraversal(root);
    printf("\n");

    printf("Postorder Traversal: ");
    postorderTraversal(root);
    printf("\n");

    return 0;
}
```

Output:

```
Inorder Traversal: 1 2 10 12 13 15 17 51 90
Preorder Traversal: 10 2 1 15 12 13 51 17 90
Postorder Traversal: 1 2 13 12 17 90 51 15 10
```