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

typedef struct treeNode{
    struct treeNode *left;
    int data;
    struct treeNode *right;
}TreeNode, *TreeNodePtr;

void inOrder(TreeNodePtr treePtr);
void preOrder(TreeNodePtr treePtr);
void postOrder(TreeNodePtr treePtr);

void inOrder(TreeNodePtr treePtr){
    if(treePtr != NULL){
        inOrder(treePtr->left);
        printf("%3d", treePtr->data);
        inOrder(treePtr->right);
    }
}

void preOrder(TreeNodePtr treePtr){
    if(treePtr != NULL){
        printf("%3d", treePtr->data);
        preOrder(treePtr->left);
        preOrder(treePtr->right);
    }
}

void postOrder(TreeNodePtr treePtr){
    if(treePtr != NULL){
        postOrder(treePtr->left);
        postOrder(treePtr->right);
        printf("%3d", treePtr->data);
    }
}

void insertNode(TreeNodePtr *treePtr, int value){
    if(*treePtr == NULL){
        *treePtr = malloc( sizeof(TreeNode) );
        (*treePtr)->data = value;
```

```c
        (*treePtr)->left = NULL;
        (*treePtr)->right = NULL;
    }
    else{
        if( value < (*treePtr)->data){
            insertNode( &((*treePtr)->left) ,value);
        }
        else if( value > (*treePtr)->data){
            insertNode( &((*treePtr)->right) ,value);
        }
        else{
            printf("DUP");
        }
    }
}

int main(void){
    TreeNodePtr root = NULL;
    int index, item;

    srand(time(NULL));
    printf("Adicionando valores na arvore\n");
    for(index = 0; index <= 15; index++){
        item = rand() % 15;
        printf("%3d", item);
        insertNode(&root, item);
    }

    printf("\n\nAtravessando a arvore em Pre Ordem\n");
    preOrder(root);

    printf("\n\nAtravessando a arvore em In Ordem\n");
    inOrder(root);

    printf("\n\nAtravessando a arvore em Pos Ordem\n");
    postOrder(root);
}
```