#include<stdio.h>//standard input output header file//

#include<stdlib.h>//standard library header file//

struct node

{

int element;//declaring the variables//

struct node \*left;//declaring the left pointer//

struct node \*right;//declaring the right pointer//

};

struct node \*createNode(int val)//create function//

{

struct node Node=(struct node)malloc(sizeof(struct node));//dynamic memory allocation//

Node->element=val;

Node->left=NULL;

Node->right=NULL;

return (Node);

}

void traversepreorder(struct node \*root)//for pre order traversal//

{

if(root==NULL)

return;

printf("%d\t",root->element);

traversepreorder(root->left);

traversepreorder(root->right);

}

void traversepostorder(struct node \*root)//for post order treaversal//

{

if(root==NULL)

return;

traversepostorder(root->left);

traversepostorder(root->right);

printf("%d\t",root->element);

}

void traverseinorder(struct node \*root)//for in order traversal//

{

if(root==NULL)

return;

traverseinorder(root->left);

printf("%d\t",root->element);

traverseinorder(root->right);

}

int main()//main function//

{

struct node \*root=createNode(36);

root->left=createNode(26);

root->right=createNode(46);

root->left->left=createNode(21);

root->left->right=createNode(31);

root->left->left->left=createNode(11);

root->left->left->right=createNode(24);

root->right->left=createNode(41);

root->right->right=createNode(56);

root->right->right->left=createNode(51);

root->right->right->right=createNode(66);

printf("Preorder traversal\n");//prints the statement//

traversepreorder(root);

printf("\nPostorder traversal\n");//prints the statement//

traversepostorder(root);

printf("\ninorder traversal\n");//prints the statement//

traverseinorder(root);

return 0;

}