Tree

#include<stdio.h>

#include<conio.h>

typedef struct node

{

int data;

struct node \*left;

struct node \*right;

};

struct node \*root=NULL,\*p,\*temp;

void preorder(struct node \*root)

{

if(root!=NULL)

{

printf("%d ",root->data);

preorder(root->left);

preorder(root->right);

}

}

void inorder(struct node \*root)

{

if(root!=NULL)

{

inorder(root->left);

printf("%d ",root->data);

inorder(root->right);

}

}

void postorder(struct node \*root)

{

if(root!=NULL)

{

postorder(root->left);

postorder(root->right);

printf("%d ",root->data);

}

}

struct node \*newnode(int item)

{

temp=(struct node\*)malloc(sizeof(struct node));

temp->data=item;

temp->left=temp->right=NULL;

return temp;

}

struct node\* insert(struct node\* node,int key)

{

if(node==NULL)

{

return newnode(key);

}

else if(key<node->data)

node->left=insert(node->left,key);

else if(key>node->data)

node->right=insert(node->right,key);

return node;

}

void main()

{

clrscr();

root=insert(root,5);

insert(root,6);

insert(root,7);

insert(root,4);

insert(root,3);

insert(root,2);

printf("InOrder : ");

inorder(root);

printf("\nPostOrder : ");

postorder(root);

printf("\nPreOrder : ");

preorder(root);

getch();

}

OUTPUT:

