

# include <stdio.h>

# include <stdlib.h>

typedef struct BST {

int data;

struct BST \*lchild, \*rchild;

} node;

void insert(node \*, node \*);

void inorder(node \*);

void preorder(node \*);

void postorder(node \*);

int main()

{

int choice;

char ans = 'N';

int key;

node \*new\_node, \*root, \*tmp, \*parent;

node \*get\_node();

root = NULL;

printf("\nProgram For Binary Search Tree ");

do {

printf("\n1.Create");

printf("\n2.Recursive Traversals");

printf("\n3.Exit");

printf("\nEnter your choice :");

scanf("%d", &choice);

switch (choice) {

case 1:

do {

new\_node = get\_node();

printf("\nEnter The Element ");

scanf("%d", &new\_node->data);

if (root == NULL)

root = new\_node;

else

insert(root, new\_node);

printf("\nWant To enter More Elements?(y/n)");

ans = getch();

} while (ans == 'y');

break;

case 2:

if (root == NULL)

printf("Tree Is Not Created");

else {

printf("\nThe Inorder display : ");

inorder(root);

printf("\nThe Preorder display : ");

preorder(root);

printf("\nThe Postorder display : ");

postorder(root);

}

break;

}

} while (choice != 3);

return 0;

}

node \*get\_node() {

node \*temp;

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

temp->lchild = NULL;

temp->rchild = NULL;

return temp;

}

void insert(node \*root, node \*new\_node) {

if (new\_node->data < root->data) {

if (root->lchild == NULL)

root->lchild = new\_node;

else

insert(root->lchild, new\_node);

}

if (new\_node->data > root->data) {

if (root->rchild == NULL)

root->rchild = new\_node;

else

insert(root->rchild, new\_node);

}

}

void inorder(node \*temp) {

if (temp != NULL) {

inorder(temp->lchild);

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

inorder(temp->rchild);

}

}

void preorder(node \*temp) {

if (temp != NULL) {

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

preorder(temp->lchild);

preorder(temp->rchild);

}

}

void postorder(node \*temp) {

if (temp != NULL) {

postorder(temp->lchild);

postorder(temp->rchild);

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

}

}