1. Write a program a) ToconstructabinarySearchtree. b) To traverse the tree using all the methods i.e., inorder, preorder and post order c) To display the elements in the tree.

Program:

#include <stdio.h> #include <stdlib.h> struct node

{

int data;

struct node \*left; struct node \*right;

};

struct node\* newNode(int data)

{

struct node\* node = (struct node\*)malloc(sizeof(struct node)); node->data = data;

node->left = node->right = NULL; return node;

}

struct node\* insert(struct node\* root, int data)

{

if (root == NULL)

return newNode(data);

if (data < root->data)

root->left = insert(root->left, data); else if (data > root->data)

root->right = insert(root->right, data);

return root;

}

void inorder(struct node\* root)

{

if (root != NULL)

{

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

}

}

void preorder(struct node\* root)

{

if (root != NULL)

{

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

}

}

void postorder(struct node\* root)

{

if (root != NULL)

{

postorder(root->left); postorder(root->right);

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

}

}

void display(struct node\* root, int choice)

{

switch (choice)

{

case 1:

printf("\nIn-order traversal: "); inorder(root);

break; case 2:

printf("\nPre-order traversal: "); preorder(root);

break; case 3:

printf("\nPost-order traversal: "); postorder(root);

break; default:

printf("\nInvalid choice\n"); break;

}

}

int main()

{

struct node\* root = NULL; int n, data, choice;

printf("Enter the number of nodes to insert in the BST: "); scanf("%d", &n);

for (int i = 0; i < n; i++)

{

printf("Enter value for node %d: ", i + 1); scanf("%d", &data);

root = insert(root, data);

}

while (1)

{

printf("\nChoose the type of traversal:\n"); printf("1. In-order\n");

printf("2. Pre-order\n"); printf("3. Post-order\n"); printf("4. Exit\n");

printf("Enter your choice (1/2/3/4): "); scanf("%d", &choice);

if (choice == 4)

{

printf("Exiting the program...\n"); break;

}

display(root, choice);

}

return 0;

}

