**#include <stdio.h>**

**#include <stdlib.h>**

**typedef struct BST**

**{**

**int data;**

**struct BST \*lchild, \*rchild;**

**} node;**

**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);**

**}**

**}**

**node \*search(node \*root, int key, node \*\*parent)**

**{**

**node \*temp;**

**temp = root;**

**while (temp != NULL)**

**{**

**if (temp->data == key)**

**{**

**printf("\nThe %d Element is Present", temp->data);**

**return temp;**

**}**

**\*parent = temp;**

**if (temp->data > key)**

**temp = temp->lchild;**

**else**

**temp = temp->rchild;**

**}**

**}**

**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);**

**}**

**}**

**void main()**

**{**

**int choice;**

**int key;**

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

**node \*get\_node();**

**root = NULL;**

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

**do**

**{**

**printf("\n1.Create Node");**

**printf("\n2.Search");**

**printf("\n3.Recursive Traversals");**

**printf("\n4.Exit");**

**printf("\n\nEnter your choice : ");**

**scanf("%d", &choice);**

**switch (choice)**

**{**

**case 1:**

**new\_node = get\_node();**

**printf("\nEnter The Element ");**

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

**if (root == NULL)**

**root = new\_node;**

**else**

**insert(root, new\_node);**

**break;**

**case 2:**

**printf("\nEnter Element to be searched :");**

**scanf("%d", &key);**

**tmp = search(root, key, &parent);**

**printf("\nParent of node %d is %d", tmp->data, parent->data);**

**break;**

**case 3:**

**if (root == NULL)**

**printf("Tree Is Not Created");**

**else**

**{**

**printf("\nThe In-order display : ");**

**inorder(root);**

**printf("\nThe Pre-order display : ");**

**preorder(root);**

**printf("\nThe Post-order display : ");**

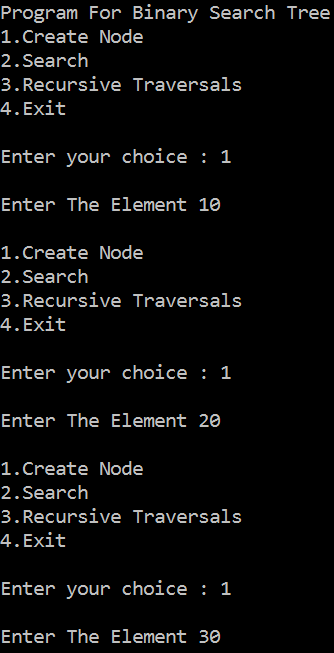
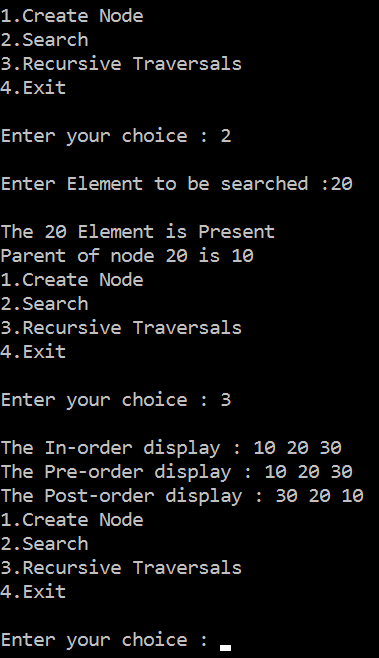
**postorder(root);**

**}**

**break;**

**}**

**} while (choice != 4);**

**} **