Program 8

Write a program

- a) To construct a binary Search tree.
- b) To traverse the tree using all the methods i.e., in-order, preorder and post order
- c) To display the elements in the tree.

```
Code:
```

```
#include <stdio.h>
#include <stdlib.h>
struct Node {
       int data;
       struct Node* left;
       struct Node* right;
};
struct Node* createNode(int data) {
       struct Node* newNode = (struct Node*)malloc(sizeof(struct Node));
  newNode->data = data;
  newNode->left = NULL;
  newNode->right = NULL;
       return newNode;
}
struct Node* insert(struct Node* root, int data) {
       if (root == NULL) {
    return createNode(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);
       }
}
int main() {
       struct Node* root = NULL;
       int n, value;
printf("Enter the number of elements to insert in the BST: ");
  scanf("%d", &n);
     printf("Enter %d elements:\n", n);
       for (int i = 0; i < n; i++) {
     scanf("%d", &value);
     root = insert(root, value);
       }
printf("\nIn-order Traversal: ");
  inOrder(root);
printf("\nPre-order Traversal: ");
  preOrder(root);
       printf("\nPost-order Traversal: ");
       postOrder(root);
```

```
return 0;
```

}

Enter the number of elements to insert in the BST: 5
Enter 5 elements:

12 23 45 65 3

In-order Traversal: 3 12 23 45 65
Pre-order Traversal: 12 3 23 45 65
Post-order Traversal: 3 65 45 23 12

Minclude Caldicia? Minclude Caldicia? Minclude Caldicia? Struck Mode? Ind data; Struck Mode? 'Ieff; Struck Mode? 'right; Struck Mode? 'right; Struck Mode? 'right; Struck Mode? 'right; Struck Mode? Create Mode Cividat a) }	the if (data > rook > data) { rook > rights hove (rook > right, data); Ythere rook rook in the force (rook > rook) { rick rook rook > rook) { interfer (rook > right); print file; rook > right); just file; rook > right); } rook or Cooker (rook > right); } rook or Cooker (rook > right); if (rook rook > right); if (rook rook > right); if (rook rook > right);
D'In construct à 851 D'In America tha face using all preside i e princette preside prode tette. O'In display the desocute in the tere. Minclude Caldicia? Rinclude Caldicia? Rinclude Caldicia? Rinclude Caldicia? Street Node. 16th; Christ Node. 16th;	Tool - Dright's hoped (rost-Dright, days); Ithurn root; Tool in Inder (some that " Tool) it (root is Nove) it (root is Nove); Dright (root is Nove); John or Optivi Same that " Tool) it (root is Nove) it
D'Te transce the tree using all proposed in emiscoheropeople post estate. O'The display the descents in the tree. disclude Statism's discrude Statism's discrude Statism's districted Statism's dincorrected Statism's districted Statism's districted Statism's di	Theon-root; total inDeter (some Node + root)? is (root in Nove)? indeter (root > stat); print("sout, root > date); in Order (root > right); } total or Order (sout + Node + root)? If (root is note)?
post enter. O'la digitary the descript in the tere. Birchale (Sidi-in) Birchale (Sidi-in) Birchale (Sidi-in) Struck Node, 1	took in Desire (Source Node ** 1904) ? is (most in Nource) ? in Desire (most -> 1548); point ("Source -> 1548); in Order (1904) -> 1548); just on Desire (1904); if (1904) -> 1904); if (1904) -> 1904);
O'in display the desocate in the sece. Minclude Saldicia? Minclude Saldicia? Minclude Saldicia? Struct Made * left; Shout Made * left; Shout Made * left; Shout Mode * create Node * left; Shout Mode * create Node * left; Shout Mode * create Node Cividat a) }	took in Desire (Source Node ** 1904) ? is (most in Nource) ? in Desire (most -> 1548); point ("Source -> 1548); in Order (1904) -> 1548); just on Desire (1904); if (1904) -> 1904); if (1904) -> 1904);
#include (Sidio-in) #include (Sidio-in) #include (Sidio-in) #include (Sidio-in) #include (Sidio-in) #include Node * left; #include N	ri(moli-Null) inthdr (ross -> lits); paid(file), ross -data); intoder (ross -> right); } soid probarlisant Nate* rost) if (rost is near);
Minclude Caldicia? Minclude Caldicia? Minclude Caldicia? Struck Mode? Ind data; Struck Mode? 'Ieff; Struck Mode? 'right; Struck Mode? 'right; Struck Mode? 'right; Struck Mode? 'right; Struck Mode? Create Mode Cividat a) }	ri(moli-Null) inthdr (ross -> lits); paid(file), ross -data); intoder (ross -> right); } soid probarlisant Nate* rost) if (rost is near);
Strout Minds (Strout Minds (int dath) Strout Minds (Str	incoder (cost = left); paid (cost = region); MO care (cost = region); } and a coder (cost = region); If (rest = result)
Strout Minds (Strout Minds (int dath) Strout Minds (Str	point ("ist", root a data"); 3 NO other (root a aright); 3 101 on Order (struct Node* 1002)* 1(1 not 1 = 11012)*
Street Niede T int date: Street Niede T (Ph. Street Nied T (Ph. Street Nie	3 (MOrder (root - mighe); 3 Jana on Order (shout Node* root) ((root ! = NOU.))
Street Niede T int date: Street Niede T (Ph. Street Nied T (Ph. Street Nie	} ### ### ############################
Chench Midded int data. Shout Made " left; Shout Made " left; Shout Mode create Node vigat; ; Shouth Node create Node Cividat a) {	} ### ### ############################
ist data; Shour Made * left; Chese Noole* right; Shout Note* creat blode Cividate) }	fCrsen = 1 foor) si
Shour Node v left; from Node v left; }; Shout Note creat Node (Goldet a) {	fCrsen = 1 foor) si
chuse Node* right; ; ; ; ; ; ; ; ; ; ; ; ;	fCrsen = 1 foor) si
Sinut Noder create Node (inidata)	fCrsen = 1 foot) si
	in Order (root -> left-):
	orivet ("-fed", root adata),
Street Node + newblode = (street Node +) malloc (street)	in Order (rook -> right)
Struck Node));	3
newNeds -> data - data;	A CONTRACTOR OF THE PARTY OF TH
newhode -> left = NULL;	The second second
	ed as Order (struct Mode* root) 1
referen neutlades	((rept = NULL))
	printl("red", root ->data);
	The Order (sept -) left)
Struck Node+ insert (Smit Node+ root, int data) f	no Order (rook -> right);
if (yeot >= NULL){	The state of the s
Yelson trave Node (data);	
4	the state of the s
illdala (reot-adala) f	d postOrder (short Nede+root) \$
rept -> left: insuffront -> left idata).	Ulared 1 = NVLL) 1

75	thre. 67
post Order (2004 7 left); post Order (2004 - 1 right); spirit ("da", 2004 - 2 data);	Output:
and Drafter (1901 -> right)	
enint ("fed", 1004 7 data):	Ener the number of elements in the RST: 6
	Enkr 5 elements:
	12 25 45 65 3
ACTION OF THE PARTY OF THE PART	
int main () {	In-Order Traversals 3 12 23 45 65
Snuct Node Tood = Nozl;	Pre-Order Travesal 1 12 3 23 45 65
int n value:	Post - Order Transsel: 3 65 45 23 12
Clare min Commen	
print ("take the number of elements to indext in the	
651:");	
Stant ("dod", da);	
The state of the s	
Price ("Enter d'od element .\n", n); Par (int i = 0; icn; i+) {	
for lint is 0; ich; ith)?	
Stanf (vdod ", Avalue);	
(col = insert (root, nature);	A CONTRACTOR OF THE PARTY OF TH
erint ("In In-order Traverat."); inOrder (1904);	
1/1 Urd (* (*00))	
prints ("In Be - Order Traymon");	
pri Order (1904);	
Prints ("In Poss Order Traversal:");	
post Order (voot);	
return 0;	