```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
 struct node {
       int data ;
      struct node * left ;
      struct node * right ;

};

 typedef struct nod * Node

 Node  get Node () {
      Node x ;
    x = (Node) malloc (sizeof (struct node));
    if ( x== NULL) {
          pf ("Memory full \n");
        exit(0) ;
     }
   return x ;
 }
```

Scanned with CamScanner

```
Node insert (Node root, int data){
        Node temp, cur, prev;
      temp = getNode();
     temp -> right = NULL;
    temp -> left = NULL;
     if(root == NULL){
        return temp;
    }
     prev = NULL;
     cur = root;
   while (cur != NULL){
         prev = cur;
          cur = (data < cur->data) ? cur->left
  }                                : cur->right;
   if( data < prev->data)
          prev-> left = temp;
  else
        prev-> right = temp;
 return root;
```

```
void preorder (Node root) {
    if (root != NULL) {
        Pf (" The item is %d \n", root-> data);
        preorder (root -> left);
        preorder (root -> right);
    }
}
void inorder (Node root) {
    if (root != NULL) {
        ~~Pf (" the~~
        inorder (root-> left);
        printf (" the item is %d \n", root->data);
    in ~~post~~order (root -> right);
    }
}
void postorder (Node root) {
    if ( root != NULL) {
        postorder (root -> left);
        postorder (root -> right);
        Pf (" The item is %d \n", root-> data);
    }
}
```

Scanned with CamScanner

```
void display (Node root, int i){
    int j;
    if (root != NULL){
        display (root->right,i+1);
        for (j=1 ; j<=i; j++){
            printf (" ");
        }
        Pf("%d\n", root->data);
        display (root-> left, i+1);
    }
}

void main (){
    int choice ,item, flag =1, ~~key~~;
    Node root = NULL;
    while ( flag==1 ){
        Pf ("1. insert \n 2. preorder \n 3. inorder \n
             4. postorder \n 5. display \n");
        printf(" Enter the choice : \n");
        scanf ("%d", & choice);

PTO
```

```
switch (choice){
        case 1:  Pf(" Enter the item :");
                 scanf ("%d", &item);
                 root = insert (root, item);
                 break

      case 2:  if(root == NULL){
                        printf (" Tree is empty");
               }else {
                     printf ("Given tree : \n");
                     display (root, 1);
                     Pf (" Pre order : \n");
                     preorder (root);
            }
            break;
   Case 3:  if (root == NULL){
                 printf ("Tree is empty \n");
          }else {
                 Pf (" giventree : \n");
                 display (root, 1);
                 Pf("inorder : \n");
                 ~~Pre~~inorder (root);
        }
        break;
```

Scanned with CamScanner

```
case 4: if (root == NULL){
            printf ("Tree is empty");
    } else {
            pf("Given tree :\n");
            display (root, 1);
            printf ("postorder : \n");
            postorder (root);
    }
    break;

case 5: display (root, 1);
        break;
default : exit(0)
```