Trees

```
#include <stdio.h>
#include <stdlib.h>

typedef struct node {
    int data;
    struct node *left, *right;
    node;
}

node *create(int data) {
    node *temp;
    temp = (node*) malloc(sizeof(node));
    temp->data = data;
    temp->left = temp->right = NULL;
    return temp;
}

Void inorder(node *root) {
    if (root != NULL) {
        inorder(root->left);
        Printf("%d", root->data);
        inorder(root->right);
    }
}
```

```
Void Preorder(node *root){
    if(root != NULL){
    Printf("%d", root->data);
    Preorder(root->left);
    Preorder(root->right);
    }
}

Void postorder(node *root){
    if(root != NULL){
    Postorder(root->left);
    Postorder(root->right);
    Printf("%d", root->data);
    }
}

Void insert(node *root, node *temp){
    if(temp->data < root->data){
    if(root->left != NULL)
        insert(root->left, temp);
      else
      root->left = temp;
      }
    }
```

```
int main(void) {
  node *root = NULL, *temp;

  int choice = 0;
  while(choice != 2)
  {
  temp =
   printf("1 - insert\n");
   printf("1 - Exit\n");

  printf("Enter your choice: ");
   scanf("%d", &choice);
   if (choice == 1)
   {
   int val;
  printf("Enter Value:");
   scanf("%d", &val);
  temp = create(val);
  if (root == NULL)
    root = temp;
    else
    insert(root, temp);
  }
```

```
else if (choice == 2)
    break;
}

Printf ("Inorder");
inorder(root);
Printf ("preorder");
preorder(root);
Printf ("postorder");
postorder(root);
```