

Binary Trees/ Recursion/ Pointers

```
using namespace std;
#include <iostream> #include <string>

struct treeNode
{
    string name;           // Name of whatever
    treeNode *left;        // left child of node
    treeNode *right;       // Right child of node
};

void insertnode(treeNode *&nodeptr, treeNode *& newnode ); // Recursive function to insert node in binary tree
void printinorder(treeNode *nodeptr ); // Recursive function to print the binary tree

int main()
{
    int i;
    treeNode *root, *newnode;

    root=NULL;

    for (i = 0; i < 10; i++)
    {
        newnode= new treeNode; // Generate new node
        cout << "\nEnter Name ";
        getline(cin, newnode->name);
        newnode->left=NULL; // Set left child of new node to NULL
        newnode->right=NULL; // Set right child of new node to NULL
        insertnode(root,newnode ); // Recursive call to insert the newq node in the binary tree
        cout<<"\n\nCurrent list is \n\n";
        printinorder(root); // Recursive call to print the binary tree
    }

    return 0;
}

void insertnode(treeNode *&nodeptr, treeNode *& newnode )
{
    if (nodeptr==NULL) // add node to binary tree
        nodeptr=newnode;
    else if (newnode->name<nodeptr->name) // Recursive call to add node to left side of node
        insertnode(nodeptr->left,newnode);
    else // Recursive call to add node to right side of node
        insertnode(nodeptr->right,newnode);
    return;
};

void printinorder(treeNode *nodeptr )
{
    if (nodeptr)
    {
        printinorder(nodeptr->left); // Recursive call to print left side of node
        cout << nodeptr->name<<endl; // Print name of node
        printinorder(nodeptr->right); // Recursive call to print right side of node
    }
    return;
};
```