//Print left view of the tree

#include <iostream>

using namespace std;

typedef struct node{

int data;

struct node \*left,\*right;

}node;

node\* createNode(int data)

{

node\* newNode=(node\*)malloc(sizeof(node));

if(newNode==NULL)

{

cout<<"unable to allocate memory";

return NULL;

}

newNode->data=data;

newNode->left=NULL;

newNode->right=NULL;

return newNode;

}

node\* insertTreeNode(node\* root,int data)

{

if(root==NULL)

{

node\* newNode=createNode(data);

return newNode;

}

else if(data<root->data)

root->left=insertTreeNode(root->left,data);

else

root->right=insertTreeNode(root->right,data);

return root;

}

void printLeftView(node\* head,int level,int \* maxlevel)

{

if(head==NULL)

return;

if (\*maxlevel<level)

{

cout<<head->data<<" ";

\*maxlevel=level;

}

printLeftView(head->left,level+1,maxlevel);

printLeftView(head->right,level+1,maxlevel);

}

void printTree(node\* head)

{

if(head==NULL)

return;

cout<<head->data<<" ";

printTree(head->left);

printTree(head->right);

}

int main()

{

node\* head=createNode(12);

insertTreeNode(head,10);

insertTreeNode(head,30);

insertTreeNode(head,25);

insertTreeNode(head,40);

insertTreeNode(head,35);

insertTreeNode(head,50);

insertTreeNode(head,60);

printTree(head);

cout<<endl<<"left view"<<endl;

//print left view;

int maxlevel=0;

printLeftView(head,1,&maxlevel);

return 0;

}