**PROGRAM 5)** Write a program to perform insertion and deletion operations on 2-3 trees.

CODE:

#include<bits/stdc++.h>

using namespace std;

class Node

{

public:

int lv,rv;

Node\* l;

Node\* m;

Node\* r;

};

Node\* newNode(int val)

{

Node\* node=new Node();

node->lv=val;

node->rv=-1;

node->l=NULL;

node->r=NULL;

node->m=NULL;

return node;

}

Node\* insert(Node\* root,int val)

{

if(root==NULL)

return newNode(val);

if(root->rv==-1&&val>root->lv)

{

root->rv=val;

}

else

{

if(val<root->lv)

root->l=insert(root->l,val);

else if(val>root->rv)

root->r=insert(root->r,val);

else if(val<root->rv&&val>root->lv)

root->m=insert(root->m,val);

}

return root;

}

Node\* deleteNode(Node\* root,int val)

{

if(root==NULL) return root;

if(val<root->lv)

root->l=deleteNode(root->l,val);

else if(val<root->rv&&val>root->lv)

root->m=deleteNode(root->m,val);

else if(val>root->rv)

root->r=deleteNode(root->r,val);

else if(val==root->lv||val==root->rv)

{

if(root->lv==val)

{

if(root->l==NULL)

{

cout<<"deleted"<<endl;

root->lv=root->rv;

root->rv=-1;

}

else

{

root->lv=root->l->rv;

root->l->rv=-1;

}

}

else if(root->rv==val)

{

if(root->r==NULL)

{

root->rv=-1;

}

else

{

root->rv=root->r->rv;

root->r->rv=-1;

}

}

}

return root;

}

void preorder(Node\* root)

{

if(root==NULL)

{

return;

}

if(root->rv==-1&&root->lv==-1)

{

return;

}

if(root->rv==-1)

{

cout<<root->lv<<":"<<"- ";

}

else

{

cout<<root->lv<<":"<<root->rv<<" ";

}

preorder(root->l);

preorder(root->m);

preorder(root->r);

}

int main()

{

Node\* root=NULL;

int val;

cout<<"Enter the values to be inserted or -99 to exit\n";

while(1)

{

cin>>val;

if(val== -99)

{

break;

}

root=insert(root,val);

}

cout<<"Preorder traversal for the 2-3 tree is\n";

preorder(root);

cout<<"\nEnter the node to deleted\n";

cin>>val;

root=deleteNode(root,val);

cout<<"Preorder traversal after deletion is\n";

preorder(root);

return 0;

}

OUTPUT:

