







**Program Code :-**

#include <iostream>

#include <conio.h>

using namespace std;

struct tree

{

tree \*l, \*r;

int data;

} \*root = NULL, \*p = NULL, \*np = NULL, \*q;

void create()

{

int value, c = 0;

while (c < 7)

{

if (root == NULL)

{

root = new tree;

cout << "Enter the value of root node\n";

cin >> root->data;

root->r = NULL;

root->l = NULL;

}

else

{

p = root;

cout << "Enter the value of node\n";

cin >> value;

while (true)

{

if (value < p->data)

{

if (p->l == NULL)

{

p->l = new tree;

p = p->l;

p->data = value;

p->l = NULL;

p->r = NULL;

cout << "value entered in left\n"<<endl;

break;

}

else if (p->l != NULL)

{

p = p->l;

}

}

else if (value > p->data)

{

if (p->r == NULL)

{

p->r = new tree;

p = p->r;

p->data = value;

p->l = NULL;

p->r = NULL;

cout << "value entered in right\n"<<endl;

break;

}

else if (p->r != NULL)

{

p = p->r;

}

}

}

}

c++;

}

}

void inorder(tree \*p)

{

if (p != NULL)

{

inorder(p->l);

cout << p->data << endl;

inorder(p->r);

}

}

void preorder(tree \*p)

{

if (p != NULL)

{

cout << p->data << endl;

preorder(p->l);

preorder(p->r);

}

}

void postorder(tree \*p)

{

if (p != NULL)

{

postorder(p->l);

postorder(p->r);

cout << p->data << endl;

}

}

int main()

{

create();

cout << "printing traversal in inorder\n";

inorder(root);

cout << "printing traversal in preorder\n";

preorder(root);

cout << "printing traversal in postorder\n";

postorder(root);

getch();

}

**Program Output :-**

Text

Description automatically generated

Text

Description automatically generated