**PROGRAM FOR BINERY TREE:**

#include<iostream>

#include<cstdlib>

using namespace std;

class BT

{

private:

struct node

{

int data;

node \*left;

node \*right;

}\*root1,\*root2;

int pre1[20],in1[20],in2[20],post2[20];

public:

node \*create(int in1[],int pre1[],int inStrt,int inEnd)

{

static int preindex=0;

if(inStrt>inEnd)

{

return NULL;

}

node \*temp;

temp=new node;

temp->data=pre1[preindex++];

temp->left=NULL;

temp->right=NULL;

if(inStrt==inEnd)

{

return temp;

}

int inIndex=search(in,1inStrt,inEnd,temp->data);

temp->left=create(in1,pre1,inStrt,inIndex-1);

temp->right=create(in1,pre1,inIndex+1,inEnd);

return temp;

}

node \*create1(int in2[],int post2[],int inStrt,int inEnd)

{

static int postindex=inEnd;

if(inStrt>inEnd)

{

return NULL;

}

node \*temp;

temp=new node;

temp->data=post2[postindex--];

temp->left=NULL;

temp->right=NULL;

if(inStrt==inEnd)

{

return temp;

}

int inIndex=search(in2,inStrt,inEnd,temp->data);

temp->right=create1(in2,post2,inIndex+1,inEnd);

temp->left=create1(in2,post2,inStrt,inIndex-1);

return temp;

}

int search(int arr[],int strt,int end,int value);

void printPreorder(node\* node);

void printInorder(node\* node);

void printPostorder(node\* node);

void construction\_BT();

};

int BT::search(int arr[],int strt,int end,int value)

{

int i;

for(i=strt;i<=end;i++)

{

if(arr[i]==value)

return i;

}

return 0;

}

void BT::printPreorder(node\* node)

{

if(node==NULL)

return;

cout<<" "<<node->data;

printPreorder(node->left);

printPreorder(node->right);

}

void BT::printInorder(node\* node)

{

if (node==NULL)

return;

printInorder(node->left);

cout<<" "<<node->data;

printInorder(node->right);

}

void BT::printPostorder(node\* node)

{

if(node==NULL)

return;

printPostorder(node->left);

printPostorder(node->right);

cout<<" "<<node->data;

}

void BT::construction\_BT()

{

int i,n,ch;

do

{

cout<<"\n-------------------------\n";

cout<<"Menu\n1.pre and in\n2.post and in\n3.Exit\n";

cout<<"---------------------------\n";

cout<<"Enter your choice\n";

cin>>ch;

switch(ch)

{

case 1:

cout <<"Enter the number of nodes in tree\n";

cin>>n;

cout<<"Enter the preorder values\n";

for(i=0;i<n;i++)

{

cin>>pre1[i];

}

cout<<"\nEnter the inorder values\n";

for(i=0;i<n;i++)

{

cin>>in1[i];

}

root1=create(in1,pre1,0,n-1);

cout<<"\nPreorder traversal of the constructed tree is\n";

printPreorder(root1);

cout<<"\nInorder traversal of the constructed tree is\n";

printInorder(root1);

cout<<"\nPostorder traversal of the constructed tree is\n";

printPostorder(root1);

cout<<"\n";

break;

case 2:

cout <<"Enter the number of nodes in tree\n";

cin>>n;

cout<<"Enter the postorder values\n";

for(i=0;i<n;i++)

{

cin>>post2[i];

}

cout<<"\nEnter the inorder values\n";

for(i=0;i<n;i++)

{

cin>>in2[i];

}

root2=create1(in2,post2,0,n-1);

cout<<"\nPreorder traversal of the constructed tree is\n";

printPreorder(root2);

cout<<"\nInorder traversal of the constructed tree is\n";

printInorder(root2);

cout<<"\nPostorder traversal of the constructed tree is\n";

printPostorder(root2);

cout<<"\n";

break;

case 3:

exit(0);

break;

}

}while(ch!=3);

}

int main()

{

BT p;

p.construction\_BT();

return 0;

}

**OUTPUT:**

-------------------------

Menu

1.pre and in

2.post and in

3.Exit

---------------------------

Enter your choice

1

Enter the number of nodes in tree

6

Enter the preorder values

5

2

1

4

7

6

Enter the inorder values

1

2

4

5

6

7

Preorder traversal of the constructed tree is

5 2 1 4 7 6

Inorder traversal of the constructed tree is

1 2 4 5 6 7

Postorder traversal of the constructed tree is

1 4 2 6 7 5

-------------------------

Menu

1.pre and in

2.post and in

3.Exit

---------------------------

Enter your choice

2

Enter the number of nodes in tree

6

Enter the postorder values

1

4

2

6

7

5

Enter the inorder values

1

2

4

5

6

7

Preorder traversal of the constructed tree is

5 2 1 4 7 6

Inorder traversal of the constructed tree is

1 2 4 5 6 7

Postorder traversal of the constructed tree is

1 4 2 6 7 5

-------------------------

Menu

1.pre and in

2.post and in

3.Exit

---------------------------

Enter your choice

3