```
#include<iostream>
#include<stack>
using namespace std;
struct Node
{
int data;
Node * left;
Node * right;
Node(int  data)
{
this->data=data;
left=right=NULL;
}
};
void inorder(Node* root)
{
stack<Node*>s;
Node*curr=root;
while(curr!=NULL||!s.empty())
{
while(curr!=NULL)
{
s.push(curr);
curr=curr->left;
}
curr=s.top();
s.pop();
```

```
cout<<curr->data<<" ";
curr=curr->right;
}
}
void preorder(Node* root)
{
if(root==NULL)
return;
stack<Node*>stack;
stack.push(root);
while(!stack.empty())
{
Node*curr=stack.top();
stack.pop();
cout<<curr->data<<" ";
if(curr->right)
{
stack.push(curr->right);
}
if(curr->left)
{
stack.push(curr->left);
}
}
}
void postorder(Node* root)
{
```

```
stack<Node*>s;
stack<Node*>vals;
Node *curr=root;
Node *temp=root;
s.push(curr);
while(!s.empty())
{
curr=s.top();
s.pop();
vals.push(curr);
if(curr->left)
{
s.push(curr->left);
}
if(curr->right)
{
s.push(curr->right);
}
}
while(!vals.empty())
{
cout<<vals.top()->data<<" ";
vals.pop();
}
}
int main()
{
```

```
Node *root=new Node(1);
root->left=new Node(2);
root->right=new Node(3);
root->left->left=new Node(4);
root->left->right=new Node(5);
cout<<"\nPreorder:";
preorder(root);
cout<<"\nInorder:";
inorder(root);
cout<<"\nPostorder:";
postorder(root);
cout<<"\n";
return 0;
}
```

```
Preorder:1 2 4 5 3
Inorder:4 2 5 1 3
Postorder:4 5 2 3 1

--------------------------------
Process exited after 0.05506 seconds with return value 0
Press any key to continue . . .
```