```
#include<iostream>
using namespace std;

class Node
{
   public:
   int data;
   Node* left;
   Node* right;
};

Node* create(int data)
{
   Node* newnode=new Node;
   newnode->data=data;
   newnode->left=NULL;
   newnode->right=NULL;
   return newnode;
}

Node* insert(Node* root,int data)
{
   if(root==NULL)
   {   root=new Node;
       root->data=data;
       root->left=NULL;
       root->right=NULL;
   }
   else
   {
       if(data > root->data)
```

```
        root->right=insert(root->right,data);
        if(data < root->data)
        root->left=insert(root->left,data);
    }
    return root;
}

void inorder(Node* root)
{
    if(root==NULL)
    return;
    inorder(root->left);
    cout<<root->data<<" ";
    inorder(root->right);
}

void preorder(Node* root)
{
    if(root==NULL)
        return;
    cout<<root->data<<" ";
    preorder(root->left);
    preorder(root->right);
}

void postorder(Node* root)
{
    if(root==NULL)
        return;
    postorder(root->left);
    preorder(root->right);
```

```
    cout<<root->data<<" ";
}

void display(Node* root)
{
    cout<<"Inorder: ";
    inorder(root);
    cout<<endl;
    cout<<"Preorder: ";
    preorder(root);
    cout<<endl;
    cout<<"Postorder: ";
    postorder(root);
    cout<<endl;
}

int search(Node* root,int& data)
{
    if(!root)
        return 0;
    if(root->data==data)
        return 1;
    else if(root->data>data)
        return search(root->left,data);
    else if(root->data<data)
        return search(root->right,data);
    else
        return 0;
}

int min_value(Node* root)
```

```
{
   Node* temp=root;
   while(temp->left!=NULL)
   {
      temp=temp->left;
   }
   return temp->data;
}

void swap_bst(Node* root)
{
   if(root==NULL)
      return;
   Node *temp;
   swap_bst(root->left);
   swap_bst(root->right);
   temp=root->left;
   root->left=root->right;
   root->right=temp;
}

int depth(Node* root)
{
   if(root==NULL)
      return 0;
   return max((depth(root->left)),(depth(root->right)))+1;
}

int main()
{
   cout<<"Enter how many values do you want to insert in BST:"<<endl;
```

```
int n;
cin>>n;
cout<<"Enter values:"<<endl;
Node* root=NULL;
for(int i=0;i<n;i++)
{
   int value;
   cin>>value;
   root=insert(root,value);
}
display(root);

while(1)
{
   cout<<"1-Insert new node.\n2-Find number of nodes in longest path.\n3-Minimum data value in BST.\n4-Swapping of left and right pointer of the BST.\n5-Search a value.\n6-Display.\n7-Exit.
"<<endl;
   int c;
   cout<<"Enter your choice"<<endl;
   cin>>c;
   if(c==1)
   {
      int value;
      cout<<"Enter the value to be inserted in bst:"<<endl;
      cin>>value;
      root=insert(root,value);
   }

   else if(c==2)
   {
      int node=depth(root);
      cout<<"Longest depth in BST:"<<node<<endl;
```

```
}

else if(c==3) {
   int min=min_value(root);
   cout<<"Minimumm value in BST is:"<<min<<endl;
}

else if(c==4) {
   cout<<"After swapping:"<<endl;
   swap_bst(root);
   display(root);
}

else if(c==5) {
   int data;
   cout<<"Enter the value to be searched:"<<endl;
   cin>>data;
   int flag=search(root,data);
   if(flag==1)
      cout<<"Value is present in BST."<<endl;
   else if(flag==0)
      cout<<"Value is not present!!!"<<endl;
}

else if(c==6)
   display(root);

else if(c==7)
{
   cout<<"End of program."<<endl;
   break;
```

```
    }

    else
        cout<<"Wrong choice!!!"<<endl;

}
return 0;
}
```

OUTPUT:-

```
Enter how many values do you want to insert in BST:
5
Enter values:
23
56
78
10
20
Inorder: 10 20 23 56 78
Preorder: 23 10 20 56 78
Postorder: 20 10 56 78 23
1-Insert new node.
2-Find number of nodes in longest path.
3-Minimum data value in BST.
4-Swapping of left and right pointer of the BST.
5-Search a value.
6-Display.
7-Exit.
Enter your choice
1
Enter the value to be inserted in bst:
```

7

1-Insert new node.

2-Find number of nodes in longest path.

3-Minimum data value in BST.

4-Swapping of left and right pointer of the BST.

5-Search a value.

6-Display.

7-Exit.

Enter your choice

2

Longest depth in BST:3

1-Insert new node.

2-Find number of nodes in longest path.

3-Minimum data value in BST.

4-Swapping of left and right pointer of the BST.

5-Search a value.

6-Display.

7-Exit.

Enter your choice

3

Minimumm value in BST is:7

1-Insert new node.

2-Find number of nodes in longest path.

3-Minimum data value in BST.

4-Swapping of left and right pointer of the BST.

5-Search a value.

6-Display.

7-Exit.

Enter your choice

5

Enter the value to be searched:

10

Value is present in BST.

1-Insert new node.

2-Find number of nodes in longest path.

3-Minimum data value in BST.

4-Swapping of left and right pointer of the BST.

5-Search a value.

6-Display.

7-Exit.

Enter your choice

6

Inorder: 7 10 20 23 56 78

Preorder: 23 10 7 20 56 78

Postorder: 7 20 10 56 78 23

1-Insert new node.

2-Find number of nodes in longest path.

3-Minimum data value in BST.

4-Swapping of left and right pointer of the BST.

5-Search a value.

6-Display.

7-Exit.

Enter your choice

4

After swapping:

Inorder: 78 56 23 20 10 7

Preorder: 23 56 78 10 20 7

Postorder: 78 56 10 20 7 23

1-Insert new node.

2-Find number of nodes in longest path.

3-Minimum data value in BST.

4-Swapping of left and right pointer of the BST.

5-Search a value.

6-Display.

7-Exit.

Enter your choice

7

End of program.

PS C:\Users\siraj\OneDrive\Desktop\vscode>