

```

#include<iostream>
using namespace std;
struct BSTnode
{
    BSTnode*right;
    BSTnode*left;
    int data;
};

BSTnode*insert(BSTnode*T,int x)
{
    if(T==NULL)
    {
        T=new BSTnode;
        T->data=x;
        T->left=T->right=NULL;
        return T;
    }
    if(x>T->data)
    {
        T->right=insert(T->right,x);
        return T;
    }
    T->left=insert(T->left,x);
    return T;
}

BSTnode*create()
{
    int n,x,i;
    BSTnode*root=NULL;
    cout<<"\nEnter Number of nodes : ";
    cin>>n;
    cout<<"Enter tree value : ";
    for(i=0;i<n;i++)
    {
        cin>>x;
        root=insert(root,x);
    }
    return root;
}

void inorder(BSTnode*T)
{
    if(T!=NULL)
    {
        inorder(T->left);
        cout<<T->data<<" ";
        inorder(T->right);
    }
}

BSTnode*find_min(BSTnode*T)
{

```

```

        while(T->left!=NULL)
            T=T->left;
        return T;
    }
    BSTnode*find_max(BSTnode*T)
    {
        while(T->right!=NULL)
            T=T->right;
        return T;
    }
    BSTnode*swap(BSTnode*T)
    {
        BSTnode *p;
        if(T!=NULL)
        {
            p=T->left;
            T->left=swap(T->right);
            T->right=swap(p);
        }
        return T;
    }
    int ht(BSTnode*T)
    {
        int hl,hr;
        if(T==NULL)
            return 0;
        if(T->left==NULL && T->right==NULL)
            return 0;
        hl=ht(T->left);
        hr=ht(T->right);
        if(hl>hr)
            return (hl+1);
        return (hr+1);
    }
    BSTnode*find(BSTnode*T,int x)
    {
        while(T!=NULL)
        {
            if(x==T->data)
                return T;

            if(x>T->data)
                T=T->right;
            else
                T=T->left;
        }
        return NULL;
    }
    int main()
    {
        BSTnode*T,*p;
        int ch,x;
    }

```

```

char c;
do
{
    cout<<"\n1. Create Tree";
    cout<<"\n2. Insert node";
    cout<<"\n3. Minimum";
    cout<<"\n4. Maximum";
    cout<<"\n5. Number of nodes in the longest path";
    cout<<"\n6. Swapping";
    cout<<"\n7. Search";
    cout<<"\nEnter your choice : ";
    cin>>ch;
    switch(ch)
    {
        case 1:
            T=create();
            inorder(T);
            break;
        case 2:
            cout<<"\nEnter node : ";
            cin>>x;
            p=insert(T,x);
            inorder(T);
            break;
        case 3:
            p=find_min(T);
            cout<<"\nMinimum : "<<p->data;
            break;
        case 4:
            p=find_max(T);
            cout<<"\nMaximum : "<<p->data;
            break;
        case 5:
            x=ht(T)+1;
            cout<<"\nNumber of nodes in the longest path : "<<x;
            break;
        case 6:
            p=swap(T);
            inorder(T);
            break;
        case 7:
            cout<<"\nEnter node : ";
            cin>>x;
            p=find(T,x);
            if(p==NULL)
                cout<<"\nNode not found";
            else
                cout<<"\nNode found";
            break;

        default:
            cout<<"\nWrong choice";
    }
}

```

```

        }

        cout<<"\nDo you want to continue?Press 'y' if yes : ";
        cin>>c;
    }while(c=='y');
    return 0;
}

```

Output

```

1. Create Tree
2. Insert node
3. Minimum
4. Maximum
5. Number of nodes in the longest path
6. Swapping
Enter your choice : 1

```

```

Enter Number of nodes : 5
Enter tree value : 23
67
89
99
76
23 67 76 89 99
Do you want to continue?Press 'y' if yes : y

```

```

1. Create Tree
2. Insert node
3. Minimum
4. Maximum
5. Number of nodes in the longest path
6. Swapping
Enter your choice : 2

```

```

Enter node : 55
23 55 67 76 89 99
Do you want to continue?Press 'y' if yes : y

```

```

1. Create Tree
2. Insert node
3. Minimum
4. Maximum
5. Number of nodes in the longest path
6. Swapping
Enter your choice : 3

```

```

Minimum : 23
Do you want to continue?Press 'y' if yes : y

```

```

1. Create Tree
2. Insert node

```

3. Minimum
4. Maximum
5. Number of nodes in the longest path
6. Swapping
Enter your choice : 4

Maximum : 99
Do you want to continue?Press 'y' if yes : y

1. Create Tree
2. Insert node
3. Minimum
4. Maximum
5. Number of nodes in the longest path
6. Swapping
Enter your choice : 5

Number of nodes in the longest path : 4
Do you want to continue?Press 'y' if yes : y

1. Create Tree
2. Insert node
3. Minimum
4. Maximum
5. Number of nodes in the longest path
6. Swapping
Enter your choice : 6
99 89 76 67 55 23
Do you want to continue?Press 'y' if yes : n

1. Create Tree
2. Insert node
3. Minimum
4. Maximum
5. Number of nodes in the longest path
6. Swapping
7. Search
Enter your choice : 7

Enter node : 55

Node found
Do you want to continue?Press 'y' if yes : y

1. Create Tree
2. Insert node
3. Minimum
4. Maximum
5. Number of nodes in the longest path
6. Swapping
7. Search
Enter your choice : 7

Enter node : 98

Node not found

Do you want to continue?Press 'y' if yes : n