

## Experiment 4

### Program Code:

```
#include<iostream>

using namespace std;

class node {

    public:

        int data;

        node *left;

        node *right;

};

class bst {

    public:

        node *root;

        bst() {

            root=NULL;

        }

        void create();

        void insert();

        void postorder(node*);

        void inorder(node *);

        void preorder(node *);

        void search(int key);

        void minimum();

        int height(node*);

};

void bst::minimum() {

    node *temp;

    int min;

    temp=root;

    while(temp->left!=NULL) {

        min=temp->data;
```

```

        temp=temp->left;

        if(temp->data<min) {

            min=temp->data;

        }

        else {

            temp=temp->left;

        }

    }

    cout<<"minimum no. is:"<<min;

}

int bst::height(node *root) {

    if(root==NULL) {

        return 0;

    }

    else {

        if(height(root->right)>height(root->left)) {

            return (1+height(root->right));

        }

        else {

            return (1+height(root->left));

        }

    }

}

void bst::create() {

    node *curr,*temp;

    int ans=1;

    do {

        cout<<"Enter data:";

        curr=new node;

        cin>>curr->data;

        curr->left=curr->right=NULL;

```

```

        if(root==NULL) {

            root=curr;

        }

        else {

            temp=root;

            while(1) {

                if(curr->data<=temp->data) {

                    if(temp->left==NULL) {

                        temp->left=curr;

                        break;

                    }

                    else {

                        temp=temp->left;

                    }

                }

                else {

                    if(temp->right==NULL) {

                        temp->right=curr;

                        break;

                    }

                    else {

                        temp=temp->right;

                    }

                }

            }

        }

        cout<<"Do you want to continue:\n1.Yes\n2.No\n";

        cin>>ans;

    }while(ans==1);

}

void bst::inorder(node *root) {

```

```

        if(root!=NULL) {

            inorder(root->left);

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

            inorder(root->right);

        }

    }

void bst::preorder(node *root) {

    if(root!=NULL) {

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

        preorder(root->left);

        preorder(root->right);

    }

}

void bst::postorder(node *root) {

    if(root!=NULL) {

        postorder(root->left);

        postorder(root->right);

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

    }

}

void bst::insert() {

    node *curr,*temp;

    int ans=1;

    cout<<"Enter data:";

    curr=new node;

    cin>>curr->data;

    curr->left=curr->right=NULL;

    if(root==NULL) {

        root=curr;

    }

    else {

```

```

temp=root;
while(1) {
    if(curr->data<=temp->data) {
        if(temp->left==NULL) {
            temp->left=curr;
            break;
        }
        else {
            temp=temp->left;
        }
    }
    else {
        if(temp->right==NULL) {
            temp->right=curr;
            break;
        }
        else {
            temp=temp->right;
        }
    }
}

}

void bst::search(int key) {
    node *curr;
    curr=root;
    while(curr!=NULL) {
        if(curr->data==key) {
            cout<<"Found";
            break;
        }
    }
}

```

```

        else {
            if(key<curr->data) {
                curr=curr->left;
            }
            else {
                curr=curr->right;
            }
        }
    }

    if(curr==NULL) {
        cout<<"Not found";
    }
}

int main() {
    bst b;

    int key,ch;

    do {
        cout<<"\n1.Create\n2.Insert\n3.Inorder\n4.Preorder\n5.Postorder\n6.Search\n7.Minimum\n8.Height\nPress 0 to exit\n";

        cout<<"Enter your choice:";

        cin>>ch;

        switch(ch) {
            case 1:
                b.create();

                break;

            case 2:
                b.insert();

                break;

            case 3:
                cout<<"Inorder traversal is: ";

                b.inorder(b.root);

```

```

        break;
    case 4:
        cout<<"Preorder traversal is: ";
        b.preorder(b.root);
        break;
    case 5:
        cout<<"Postorder traversal is: ";
        b.postorder(b.root);
        break;
    case 6:
        cout<<"\nEnter key:";
        cin>>key;
        b.search(key);
        break;
    case 7:
        b.minimum();
        break;
    case 8:
        cout<<"Height of tree: "<<b.height(b.root);
        break;
    }
} while(ch!=0);
return 0;
}

```

### **Output:**

- 1.Create
- 2.Insert
- 3.Inorder
- 4.Preorder
- 5.Postorder
- 6.Search

7.Minimum

8.Height

Press 0 to exit

Enter your choice: 1

Enter data:5

Do you want to continue:

1.Yes

2.No

1

Enter data:2

Do you want to continue:

1.Yes

2.No

1

Enter data:7

Do you want to continue:

1.Yes

2.No

1

Enter data:3

Do you want to continue:

1.Yes

2.No

1

Enter data:8

Do you want to continue:

1.Yes

2.No

1

Enter data:4

Do you want to continue:



1.Yes

2.No

1

Enter data:6

Do you want to continue:

1.Yes

2.No

2

1.Create

2.Insert

3.Inorder

4.Preorder

5.Postorder

6.Search

7.Minimum

8.Height

Press 0 to exit

Enter your choice:3

Inorder traversal is: 2 3 4 5 6 7 8

1.Create

2.Insert

3.Inorder

4.Preorder

5.Postorder

6.Search

7.Minimum

8.Height

Press 0 to exit

Enter your choice:4

Preorder traversal is: 5 2 3 4 7 6 8

1.Create

2.Insert

3.Inorder

4.Preorder

5.Postorder

6.Search

7.Minimum

8.Height

Press 0 to exit

Enter your choice:5

Postorder traversal is: 4 3 2 6 8 7 5

1.Create

2.Insert

3.Inorder

4.Preorder

5.Postorder

6.Search

7.Minimum

8.Height

Press 0 to exit

Enter your choice:6

Enter key:7

Found

1.Create

2.Insert

3.Inorder

4.Preorder

5.Postorder

6.Search

7.Minimum

8.Height

Press 0 to exit

Enter your choice:7

minimum no. is:2

1.Create

2.Insert

3.Inorder

4.Preorder

5.Postorder

6.Search

7.Minimum

8.Height

Press 0 to exit

Enter your choice:8

Height of tree: 4

1.Create

2.Insert

3.Inorder

4.Preorder

5.Postorder

6.Search

7.Minimum

8.Height

Press 0 to exit

Enter your choice:0

-----

Process exited after 105.9 seconds with return value 0

Press any key to continue . . .