AVL Tree (Prajwal Horti)

1.Insert

Input: 100,50,89,36,15,72,20,25 (in this order)

Output: Print the tree

```
Select the Operation that you want to Perform?
1.Insert Node
2.Search Node(Iterative)
3.Search Node(Recursive)
4.Delete Node
5.Print AVL Tree
6.Height of Tree
7.Clear Screen
0.Exit Program
5
AVL Tree(Horizontal View):

                        100

              89

                        72

     50

                        36

                                  25

              20

                        15

Level Order BFS: 50 20 89 15 36 72 100 25
Postorder Traversal: 15 25 36 20 72 100 89 50
Preorder Traversal: 50 20 15 36 25 89 72 100
Inorder Traversal: 15 20 25 36 50 72 89 100
```

2.Search Node (Iterative) and 3. Search Node (Recursive)

Input: 36 and 22 respectively

Output:

```
Select the Operation that you want to Perform?
1.Insert Node
2.Search Node(Iterative)
3.Search Node(Recursive)
4.Delete Node
5.Print AVL Tree
6.Height of Tree
7.Clear Screen
0.Exit Program
2
Enter the value of the Node to Search in AVL Tree: 36
Value Found

Select the Operation that you want to Perform?
1.Insert Node
2.Search Node(Iterative)
3.Search Node(Recursive)
4.Delete Node
5.Print AVL Tree
6.Height of Tree
7.Clear Screen
0.Exit Program
3
Enter the value of the Node to Search in AVL Tree: 22
Value Not Found
```

4.Delete Node

Input: Node to Delete = 20

Output:

Before Deletion

```
1.Insert Node
2.Search Node(Iterative)
3.Search Node(Recursive)
4.Delete Node
5.Print AVL Tree
6.Height of Tree
7.Clear Screen
0.Exit Program
5
AVL Tree(Horizontal View):

                        100

              89

                        72

     50

                        36

                                  25

              20

                        15

Level Order BFS: 50 20 89 15 36 72 100 25
Postorder Traversal: 15 25 36 20 72 100 89 50
Preorder Traversal: 50 20 15 36 25 89 72 100
Inorder Traversal: 15 20 25 36 50 72 89 100
```

After Deletion

```
Enter the value of the Node to Delete in AVL Tree: 20
Value Deleted

Select the Operation that you want to Perform?
1.Insert Node
2.Search Node(Iterative)
3.Search Node(Recursive)
4.Delete Node
5.Print AVL Tree
6.Height of Tree
7.Clear Screen
0.Exit Program
5
AVL Tree(Horizontal View):

                        100

              89

                        72

     50

                        36

              25

                        15

Level Order BFS: 50 25 89 15 36 72 100
Postorder Traversal: 15 36 25 72 100 89 50
Preorder Traversal: 50 25 15 36 89 72 100
Inorder Traversal: 15 25 36 50 72 89 100
```

5.Print the Tree

Output:

```
Select the Operation that you want to Perform?
1.Insert Node
2.Search Node(Iterative)
3.Search Node(Recursive)
4.Delete Node
5.Print AVL Tree
6.Height of Tree
7.Clear Screen
0.Exit Program
5
AVL Tree(Horizontal View):

                        100

              89

                        72

     50

                        36

                                  25

              20

                        15

Level Order BFS: 50 20 89 15 36 72 100 25
Postorder Traversal: 15 25 36 20 72 100 89 50
Preorder Traversal: 50 20 15 36 25 89 72 100
Inorder Traversal: 15 20 25 36 50 72 89 100
```

6.Height of Tree

Output:

```
Select the Operation that you want to Perform?
1.Insert Node
2.Search Node(Iterative)
3.Search Node(Recursive)
4.Delete Node
5.Print AVL Tree
6.Height of Tree
7.Clear Screen
0.Exit Program
6
Tree Height: 3
```

7.Clear Screen

Output:

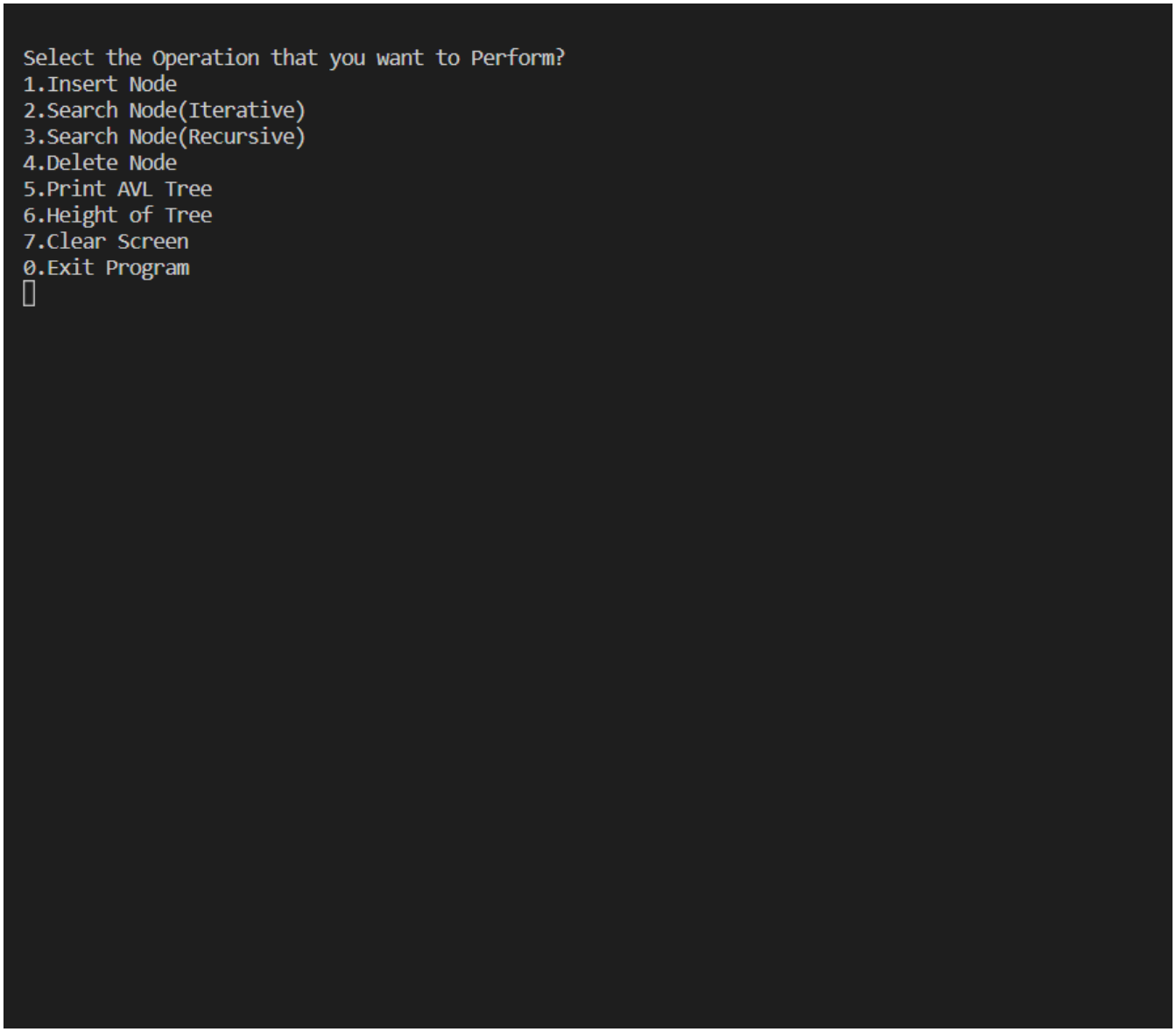
Before Clear Screen

```
Select the Operation that you want to Perform?
1.Insert Node
2.Search Node(Iterative)
3.Search Node(Recursive)
4.Delete Node
5.Print AVL Tree
6.Height of Tree
7.Clear Screen
0.Exit Program
1
Enter the vaue of the Node to insert in AVL Tree: 50
Value inserted !


Select the Operation that you want to Perform?
1.Insert Node
2.Search Node(Iterative)
3.Search Node(Recursive)
4.Delete Node
5.Print AVL Tree
6.Height of Tree
7.Clear Screen
0.Exit Program
1
Enter the vaue of the Node to insert in AVL Tree: 56
Value inserted !


Select the Operation that you want to Perform?
1.Insert Node
2.Search Node(Iterative)
3.Search Node(Recursive)
4.Delete Node
5.Print AVL Tree
6.Height of Tree
7.Clear Screen
0.Exit Program
```

After Clear Screen



0.Exit Program

Input: 0

Output: