

Green University of Bangladesh Department of Computer Science and Engineering (CSE)

Faculty of Sciences and Engineering
Semester: (Summer , Year:2022), B.Sc. in CSE (Day)
LAB REPORT NO 6

Course Title: Structured Programming Lab

Course Code: CSE 106 Section:PC-213DA

Lab Experiment Name: Linear Search & Binary Search

Student Details

Name	ID
MD Dulal Hossain	213902116

Lab Date :17-08-2022 Submission Date :23-08-2022

Course Teacher's Name :Farhana Akther Sunny

[For Teachers use only:Don't Write Anything inside this box]

<u>Lab Report Status</u>	
Marks:	Signature:
Comments:	Date:

Algorithm

Main Function

```
if (node == NULL )
return new_node(value);

else
{
   if (value < node->data)
    node->left = insert(node->left, value);
   else if (value > node->data)
     node->right = insert(node->right, value);
}
return node;
```

Inorder traversal

First, visit all the nodes in the left subtree

Then the root node

Visit all the nodes in the right subtree

Preorder traversal

Visit root node

Visit all the nodes in the left subtree

Visit all the nodes in the right subtree

Postorder traversal

Visit all the nodes in the left subtree

Visit all the nodes in the right subtree

Visit the root node

Source Code

```
#include <stdio.h>
#include <stdlib.h>
struct node
{
int data;
struct node *left, *right;
};
struct node *new_node(int n)
{ struct node *temp;
 temp = (struct node *)malloc(sizeof(struct node));
 temp->data = n;
 temp->left = temp->right = NULL;
 return temp;
struct node *insert(struct node *node, int value)
 if (node == NULL)
 return new_node(value);
 else
  if (value < node->data)
  node->left = insert(node->left, value);
  else if (value > node->data)
    node->right = insert(node->right, value);
 }
 return node;
}
void inorder(struct node *root)
 if(root != NULL)
    inorder(root->left);
    printf("%d ->",root->data);
    inorder(root->right);
```

```
}
}
void preorder(struct node *root)
 if (root != NULL)
    printf("%d -> ", root->data);
    preorder(root->left);
    preorder(root->right);
  }
}
void postorder(struct node *root)
{
 if (root != NULL)
 {
        postorder(root->left);
        postorder(root->right);
        printf("%d ->", root->data);
 }
}
int main()
        struct node *root = NULL;
        int n,i;
        printf("\n Enter the Egede Number : ");
        scanf("%d",&n);
        int a[n];
        for (i = 0; i < n; i++)
          {
        printf("User Input Number %d : ",i +1);
        scanf("%d", &a[i]);
   }
        for (i = 0; i < n; i++)
                root = insert(root, a[i]);
        while (1)
        int n;
        printf("\n 1. Preorder ");
        printf("\n 2. Inorder ");
        printf("\n 3. Postorder ");
        printf("\n 4. Exit ");
```

```
printf("\nEnter You Want To Choice: ");
        scanf("%d", &n);
                switch (n)
                {
                case 1:
                printf("\n\n");
                printf("\nPreorder Traversal : ");
                preorder(root);
                break;
                case 2:
                printf("\n\n");
                printf("\nInorder Traversal : ");
                inorder(root);
                break;
                case 3:
                printf("\n\n");
                printf("\nPostorder Traversal :");
                postorder(root);
                break;
                case 0:
                exit(0);
                break;
                default:
                printf("\nWrrong Choice Try Again ");
                break;
                }
        }
        return 0;
}
```

Output

```
13
        return temp;
14 L
                                                            D:\Coding in C\CSE 106\Lab report 6.exe
15
      struct node *insert(struct node *node, int value)
16 🗔 🧃
17
       if (node == NULL )
                                                             Enter the Egede Number: 4
18
       return new node(value);
                                                            User Input Number 1 : 12
19
                                                            User Input Number 2 : 3
20
       else
                                                            User Input Number 3 : 4
21 -
                                                            User Input Number 4 : 2
         if (value < node->data)
22
23
         node->left = insert(node->left, value);
                                                             1. Preorder
         else if (value > node->data)
24
                                                             2. Inorder
25
             node->right = insert(node->right, value);
                                                             3. Postorder
26
                                                            4. Exit
27
        return node;
                                                            Enter You Want To Choice: 1
28
29
     void inorder(struct node *root)
30
31 🖵 {
32
       if(root != NULL)
                                                           Preorder Traversal : 12 -> 3 -> 2 ->
33 —
                                                            1. Preorder
34
             inorder(root->left);
                                                            2. Inorder
35
             printf("%d ->",root->data);
                                                            3. Postorder
36
             inorder(root->right);
                                                            4. Exit
37
                                                            Enter You Want To Choice: 2
38 L
39
     void preorder(struct node *root)
40 🖵 {
41
        if (root != NULL)
                                                            Inorder Traversal : 2 ->3 ->4 ->12 -
42 -
                                                            1. Preorder
43
             printf("%d -> ", root->data);
44
             preorder(root->left);
                                                            2. Inorder
45
                                                             3. Postorder
             preorder(root->right);
46
                                                            4. Exit
47 L
                                                            Enter You Want To Choice: 3
     void postorder(struct node *root)
48
49 🗏 {
51
       if (root != NULL)
                                                            Postorder Traversal :2 ->4 ->3 ->12
52 -
                                                            1. Preorder
             postorder(root->left);
53
                                                             2. Inorder
54
             postorder(root->right);
                                                            3. Postorder
55
             printf("%d ->", root->data);
                                                            4. Exit
56
                                                            Enter You Want To Choice: 4
57
58
```

59

int main()