

# Bahria University,

## Karachi Campus



### LAB EXPERIMENT NO. \_12\_ LIST OF TASKS

TASK NO	OBJECTIVE
Task 1	Create a data structure to store a binary search. Implement a method to insert and search an element in BST.
Task 2	Create a data structure to store a binary search. Implement a method to insert and traverse in BST (inorder, postorder, preorder).
Task 3	N/A
Task 4	N/A
Task 5	N/A
Task 6	N/A
Task 7	N/A
Task 8	N/A

**Submitted On:**

\_\_17/06/2020\_\_  
(Date: DD/MM/YY)

**Task No. 1:** Create a data structure to store a binary search. Implement a method to insert and search an element in BST.

### Coding:

```

Q1.cpp Q2.cpp
1  #include<iostream>
2  using namespace std;
3
4  struct Node
5  {
6      char data;
7      Node *left,*right;
8  };
9
10 Node *Insert(Node *root,char data)
11 {
12     if(root==NULL){
13         root=new Node();
14         root->data=data;
15         root->left=root->right=NULL;
16     }
17     else if(data<=root->data){
18         root->left=Insert(root->left,data);
19     }
20     else{
21         root->right=Insert(root->right,data);
22     }
23     return root;
24 }
25
26 void searchData(Node *root,char n)
27 {
28     if(n==root->data){
29         cout<<"Element found!"<<endl;
30     }
31     else if(n<root->data){
32         searchData(root->left,n);
33     }
34     else {
35         searchData(root->right,n);
36     }
37 }
38
39 int main()
40 {
41     Node *root=NULL;
42     root=Insert(root,'Q');
43     root=Insert(root,'A');
44     root=Insert(root,'S');
45     root=Insert(root,'I');
46     root=Insert(root,'M');
47     root=Insert(root,'H');
48     root=Insert(root,'A');
49     root=Insert(root,'S');
50     root=Insert(root,'S');
51     root=Insert(root,'A');

```

```

52     root=Insert(root,'N');
53     char letter;
54     cout<<"Enter character to search: ";
55     cin>>letter;
56     searchData(root,letter);
57 }

```

## Output:

```

E:\4th semester\Data Strcture and Algorithms\12 trees\Q1.exe
Enter character to search: Q
Element found!
-----
Process exited after 3.519 seconds with return value 0
Press any key to continue . . .

```

**Task No. 2:** Create a data structure to store a binary search. Implement a method to insert and traverse in BST (inorder, postorder, preorder).

## Coding:

```

Q1.cpp Q2.cpp
1  #include<iostream>
2  using namespace std;
3
4  struct Node
5  {
6      char data;
7      struct Node *left;
8      struct Node *right;
9  };
10
11 Node* Insert(Node *root, char data)
12 {
13     if (root == NULL) {
14         root = new Node();
15         root->data = data;
16         root->left = root->right = NULL;}
17     else if (data <= root->data)
18         root->left = Insert(root->left, data);
19     else
20         root->right = Insert(root->right, data);
21     return root;}
22
23 void Preorder(struct Node *root)
24 {
25     if (root == NULL) return;
26     cout<<root->data<<" ";
27     Preorder(root->left);
28     Preorder(root->right);
29 }
30
31 void Inorder(Node *root)
32 {
33     if (root == NULL) return;

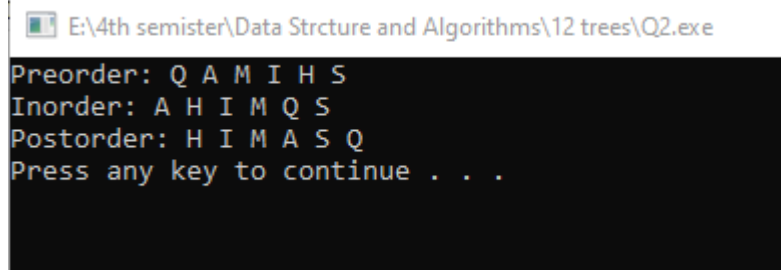
```

```

34     Inorder(root->left);
35     cout<<root->data<<" ";
36     Inorder(root->right);
37 }
38
39 void Postorder(Node *root)
40 {
41     if (root == NULL) return;
42     Postorder(root->left);
43     Postorder(root->right);
44     cout<<root->data<<" ";
45 }
46
47 int main()
48 {
49     Node* root = NULL;
50     root = Insert(root, 'Q'); root = Insert(root, 'A');
51     root = Insert(root, 'S'); root = Insert(root, 'M');
52     root = Insert(root, 'I'); root = Insert(root, 'H');
53     cout << "Preorder: ";
54     Preorder(root);
55     cout << "\n";
56     cout << "Inorder: ";
57     Inorder(root);
58     cout << "\n";
59     cout << "Postorder: ";
60     Postorder(root);
61     cout << "\n";
62     system("pause");
63 }

```

## Output:



```

E:\4th semister\Data Strcture and Algorithms\12 trees\Q2.exe
Preorder: Q A M I H S
Inorder: A H I M Q S
Postorder: H I M A S Q
Press any key to continue . . .

```