EXPERIMENT-9

Student Name: Garv Khurana **UID:** 21BCS6615
Branch: BE CSE **Section/Group:** 21AML-9/A
Semester: 3 **Date of Performance:** 2-11-22
Subject Name: Data Structures **Subject Code:** CSH-241

1. Aim/Overview of the practical:

Implement menu driven program in c for the following on binary search tree of integers, create a binary search tree for integers 6,9,5,2,8,15,24,14,7,8,5,2.

Traverse binary search tree in preorder, Inorder and postorder techinique.

2. Task to be done:

Binary Tree has 0,1 or atmost 2 nodes/branching.
Traversing in binary tree is as follows:

- Preorder – Root, Left Sub tree, Right Sub tree
- Inorder - Left Sub tree, Root, Right Sub tree
- Postorder- Left Sub tree, Right Sub tree, Root

To perform preorder, postorder and inorder traversing in binary search tree.

3. Algorithm/Flowchart:

Algorithm for preorder traversing:

```
void preorder(struct node* root){
if (root!=NULL){       printf
```

(“%d”, root->data);

preorder(root->left);

preorder(root->right);

}

}

Algorithm for postorder traversing:

```
void postOrder(NODE*node){
     if(node!=NULL){
postOrder(node->left);          postOrder(node->right);
        printf("%d\t", node->data);
   }
}
```

Algorithm for inorder traversing:

```
   void inOrder(NODE*node){
if(node!=NULL){          inOrder(node->left);          printf("%d\t", node->data);
        inOrder(node->right);
   }
}
```

4. Code and Output:

```
#include<stdio.h>
```

```
#include <stdlib.h>
struct BST
{
   int data;     struct
BST*left;     struct
BST*right;
};
typedef struct BST NODE;
NODE *node;
NODE* createtree(NODE*node,int data)
{
if(node==NULL)
   {
      NODE*temp;
temp=(NODE*)malloc(sizeof(NODE));
temp->data=data;        temp->left=temp-
>right=NULL;        return temp;
   }
   if(data<(node->data)){        node->left=createtree(node-
>left, data);
   }
```

```
    else if(data>node->data)
    {
        node-> right=createtree(node->right,data);
    }     return
node;
}
void preorder(NODE*node)
{
if(node!=NULL)
    {
        printf("%d\t",node->data);
preorder(node->left);         preorder(node-
>right);
    }
}
void    postOrder(NODE*node){
if(node!=NULL){
postOrder(node->left);
postOrder(node->right);
printf("%d\t", node->data);
    }
```

```
}

void      inOrder(NODE*node){
if(node!=NULL){
inOrder(node->left);
printf("%d\t",        node->data);
inOrder(node->right);
   }
} void
main() {
   int data, ch,i,n;
   NODE*root =NULL;
   while (1)
   {
      printf("\n****Binary Search Tree Operation****\n");
printf("\n1.Insertion in Binary search tree");
printf("\n2. preorder");        printf("\n3. inorder");
printf("\n4. postorder");        printf("\n5. Exit!!");
printf("\nEnter your choice: ");        scanf("%d",&ch);

      switch (ch)
      {
```

```
        case 1: printf("\nEnter N value:");
scanf("%d",&n);
                printf("\nEnter the values to create BST
like(6,9,5,2,8,15,24,14,7,8,5,2)\n");
for(i=0;i<n;i++)
                {
                   scanf("%d",&data);
root=createtree(root,data);
                }
break;
            case 2: printf("\nPreorder Traversal:\n");
preorder(root);              break;
            case 3: printf("\nInorder Traversal:\n");
inOrder(root);              break;
            case 4: printf("\nPostorder Traversal:\n");
postOrder(root);              break;              case 5:

            printf("Exit!!");
exit(0);              default:
        printf("\nINVALID CHOICE...TRY AGAIN!!");
break;
        }}
```

}

```
PS E:\DSA> cd "e:\DSA\" ; if ($?) { gcc bst2.c -o bst2 } ; if ($?) { .\bst2 }

****Binary Search Tree Operation****

1.Insertion in Binary search tree
2. preorder
3. inorder
4. postorder
5. Exit!!
Enter your choice: 1

Enter N value:12

Enter the values to create BST like(6,9,5,2,8,15,24,14,7,8,5,2)
6
9
5
2
8
15
24
14
7
8
5
2

****Binary Search Tree Operation****
```

```
1.Insertion in Binary search tree
2. preorder
3. inorder
4. postorder
5. Exit!!
Enter your choice: 2

Preorder Traversal:
6       5       2       9       8       7       15      14      24
****Binary Search Tree Operation****

1.Insertion in Binary search tree
2. preorder
3. inorder
4. postorder
5. Exit!!
Enter your choice: 3

Inorder Traversal:
2       5       6       7       8       9       14      15      24
****Binary Search Tree Operation****

1.Insertion in Binary search tree
2. preorder
3. inorder
4. postorder
5. Exit!!
Enter your choice: 4

Postorder Traversal:
2       5       7       8       14      24      15      9       6
****Binary Search Tree Operation****
```

```
****Binary Search Tree Operation****

1.Insertion in Binary search tree
2. preorder
3. inorder
4. postorder
5. Exit!!
Enter your choice: 5
Exit!!
PS E:\DSA>
```

Learning outcomes (What I have learnt):

- I have learnt about Data Structures.
- I have learnt about application of Data Structures.
- I have learnt about Tree.
- I have learnt about binary tree and different traversing techniques on binary search tree.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.	PERFORMANCE		12
2.	WORKSHEET		08
3.	VIVA		10