

Program 10

Develop a menu driven Program in C for the following operations on Binary Search Tree (BST) of Integers .

- a. Create a BST of N Integers: 6, 9, 5, 2, 8, 15, 24, 14, 7, 8, 5, 2
- b. Traverse the BST in Inorder, Preorder and Post Order
- c. Search the BST for a given element (KEY) and report the appropriate message
- d. Exit

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

```

if (data < (root->data))
{
    root->left = createtree(root->left, data);
}
else if (data > root->data)
{
    root -> right = createtree(root->right, data);
}
return root;
}

```

```

NODE search(int key ,NODE root)
{
    if(root == NULL)
        printf("\nElement not found");
    else if(key < root->data)
    {
        root->left=search(key,root->left);
    }
    else if(key > root->data)
    {
        root->right=search(key,root->right);
    }
    else
        printf("\nElement found is: %d", root->data);
    return root;
}

```

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

void preorder(NODE root)
{
    if(root != NULL)
    {
        printf("%d\t", root->data);
        preorder(root->left);
        preorder(root->right);
    }
}

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

```

void main()
{
    int data, ch, i, n, key;
    NODE *root=NULL;
    while (1)
    {
        printf("\n1.Insertion \n2.Inorder\n3.Preorder\n4.Postorder\n5.search\n6.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("\nInorder Traversal: \n");
                    inorder(root);
                    break;
            case 3: printf("\nPreorder Traversal: \n");
                    preorder(root);
                    break;
            case 4: printf("\nPostorder Traversal: \n");
                    postorder(root);
                    break;

```

```

case 5: printf("enetr the key element to search\n");
        scanf("%d",&key);
        search(key,root);
        break;
default:exit(0);
}
}
}

```

1.Insertion in Binary Search Tree

2.Inorder

3.Preorder

4.Postorder

5.search

6.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

1.Insertion in Binary Search Tree

- 2.Inorder
- 3.Preorder
- 4.Postorder
- 5.search
- 6.Exit

Enter your choice: 2

Inorder Traversal:

2 5 6 7 8 9 14 15 24

1.Insertion in Binary Search Tree

- 2.Inorder
- 3.Preorder
- 4.Postorder
- 5.search
- 6.Exit

Enter your choice: 3

Preorder Traversal:

6 5 2 9 8 7 15 14 24

1.Insertion in Binary Search Tree

- 2.Inorder
- 3.Preorder
- 4.Postorder
- 5.search
- 6.Exit

Enter your choice: 4

Postorder Traversal:

2 5 7 8 14 24 15 9 6

1.Insertion in Binary Search Tree

- 2.Inorder
- 3.Preorder
- 4.Postorder

5.search

6.Exit

Enter your choice: 5

enetr the key element to search

24

Element found is: 24

1.Insertion in Binary Search Tree

2.Inorder

3.Preorder

4.Postorder

5.search

6.Exit

Enter your choice: