

```

/*Design, Develop and Implement
ent 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 info;
    struct BST *llink;
    struct BST *rlink;
};
typedef struct BST *NODE;
NODE getnode()
{
    NODE x;
    x=(NODE)malloc(sizeof(struct BST));
    if(x==NULL)
    {
        printf("Out of Memory\n");
        exit(0);
    }
    return x;
}

NODE create(int item, NODE root)
{
    NODE temp, cur, prev;
    temp = getnode();
    temp->info=item;
    temp->llink = NULL;
    temp->rlink = NULL;
    if(root == NULL)
        return temp;
    prev=NULL;
    cur=root;
    while( cur!= NULL )
    {
        prev=cur;
        if(item==cur->info)
        {
            printf("Duplicate items not allowed\n");
            free(temp);
            return root;
        }
        if(item < cur->info)
            cur=cur->llink;
        else
            cur=cur->rlink;
    }
    if(item < prev->info)
        prev->llink=temp;
    else
        prev->rlink=temp;
    return root;
}

void search(NODE root)
{
    int key;
    NODE cur;
    if(root == NULL)
    {
        printf("\nBST is empty.");
        return;
    }
    printf("\nEnter Element to be searched: ");
    scanf("%d", &key);
    cur = root;
    while (cur != NULL)
    {
        if (cur->info == key)

```

```

{
    printf("\nKey element is present in BST");
    return;
}
if (key < cur->info)
cur = cur->llink;
else
cur = cur->rlink;
}
printf("\nKey element is not found in the BST");
}

void inorder(NODE root)
{
if(root != NULL)
{
inorder(root->llink);
printf("%d ", root->info);
inorder(root->rlink);
}
}

void preorder(NODE root)
{
if (root != NULL)
{
printf("%d ", root->info);
preorder(root->llink);
preorder(root->rlink);
}
}

void postorder(NODE root)
{
if (root != NULL)
{
postorder(root->llink);
postorder(root->rlink);
printf("%d ", root->info);
}
}

void main()
{
int ch, key,n, item,i;
NODE root, temp;
clrscr();
root=NULL;
for(;;)
{
printf("\n----BST MENU---\n");
printf("\n 1.Create a BST\n 2.BST Traversals\n 3.Search\n 4.Exit \n");
printf("\nEnter your choice: ");
scanf("%d", &ch);
switch(ch)
{
case 1: printf("\nEnter the number of elements: ");
scanf("%d", &n);
printf("Enter elements\n");
for(i=1;i<=n;i++)
{
scanf("%d", &item);
root= create(item, root);
}
break;
case 2: if (root == NULL)
printf("\nTree Is Not Created");
else
{
printf("\nThe Preorder display :");
preorder(root);
printf("\nThe Inorder display :");
inorder(root);
printf("\nThe Postorder display :");
postorder(root);
}
}
}

```

```
        }
        break;
    case 3: search(root);
        break;
    case 4: exit(0);
    }
}
}
```

/* OUTPUT

----BST MENU----

- 1.Create a BST
- 2.BST Traversals
- 3.Search
- 4.Exit

Enter your choice: 1

Enter the number of elements: 12

Enter elements

6
9
5
2
8
15
24
14
7
8

Duplicate items not allowed

5
Duplicate items not allowed
2
Duplicate items not allowed

----BST MENU----

- 1.Create a BST
- 2.BST Traversals
- 3.Search
- 4.Exit

Enter your choice: 2

The Preorder display :6 5 2 9 7 8 15 14 24
The Inorder display :2 5 6 7 8 9 14 15 24
The Postorder display :2 5 8 7 14 24 15 9 6

----BST MENU----

- 1.Create a BST
- 2.BST Traversals
- 3.Search
- 4.Exit

Enter your choice: 3

Enter Element to be searched: 14

Key element is present in BST

----BST MENU----

- 1.Create a BST
- 2.BST Traversals
- 3.Search
- 4.Exit

Enter your choice: 3

Enter Element to be searched: 99

Key element is not found in the BST

----BST MENU----

- 1.Create a BST
- 2.BST Traversals
- 3.Search
- 4.Exit

Enter your choice: 4 */