```
/*10. Design, Develop and Implement 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 Node
{
    int data;
    struct Node* left;
    struct Node* right;
};
typedef struct Node* NODEPTR;

NODEPTR root = NULL;

NODEPTR getnode(int ele)
{
    NODEPTR temp;
    temp = (NODEPTR)malloc(sizeof(struct Node));
    temp->data = ele;
    temp->left = NULL;
    temp->right = NULL;
    return temp;
}

NODEPTR create(NODEPTR trace,int ele)
{
    if(trace == NULL)
    {
        NODEPTR temp;
        temp = getnode(ele);
        trace = temp;
    }
    else
    {
        if(trace->data == ele)
            return trace;
        else if(trace->data < ele)
            trace->right = create(trace->right, ele);
        else if(trace->data > ele)
            trace->left = create(trace->left, ele);
    }
    return trace;
}
void inorder(NODEPTR temp)
{
    if(temp != NULL)
    {
        inorder(temp->left);
        printf("%d\t",temp->data);
        inorder(temp->right);
```

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

void search(int key)
{
    NODEPTR temp = root;
    while (temp!= NULL)
        {
            if(temp->data == key)
            {
                printf(" The key element's search is SUCCESSFUL!\n");
                return;
            }
            else if(temp->data > key)
                temp = temp->left;
            else
                temp = temp->right;
        }
        printf("The key element's search is UNSUCCESSFUL!");
}
void main()
{
    int  n, i,ele, key;
    char ch;
    while (1)
      {
            printf("\n\n\n********** Binary Search Tree (BST) Operation
Menu**********\n\n");
            printf("1. Create a BST of N Integers\n");
            printf("2. Traverse the BST in InOrder, PreOrder and
PostOrder\n");
            printf("3. Search the BST for a given element (KEY)\n");
            printf("4. Exit\n");
            printf("Enter your choice:\n");
            scanf("%d", &ch);
            switch (ch)
            {
             case 1: printf("Enter the value of N\n");
                     scanf("%d", &n);
                     printf("Enter %d Nodes\n",n);
```

```
                    for(i = 0;i < n;i++)
                    {
                        scanf("%d", &ele);
                        root = create(root,ele);
                    }
                    break;
            case 2: printf("\nThe INORDER traversal for given BST is\n");
                    inorder(root);
                    printf("\nThe PREORDER traversal for given BST
is\n");
                    preorder(root);
                    printf("\nThe POSTORDER traversal for given BST
is\n");
                    postorder(root);
                    break;
            case 3: printf("Enter the Key value to search\n");
                    scanf("%d", &key);
                    search(key);
                    break;
            case 4: exit(0);
            default: printf("Enter a Valid Choice\n");
                    break;
            }
     }
}
```