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 In-order, Pre-order 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 value;
        struct node *ltree,*rtree;
};

typedef struct node *NODE;

NODE getnode(){
        NODE x;
        x=(NODE)malloc(sizeof(struct node));
        x->ltree=x->rtree=NULL;
        return x;
}

NODE create(int item,NODE root){
        NODE temp,cur,prev;
        temp=getnode();
        temp->value=item;
        if(root==NULL){
                root=temp;
                return root;
        }
        prev=NULL;
        cur=root;
        while (cur!=NULL){
                prev=cur;
                if(temp->value==cur->value)
                        return root;
                cur=(temp->value<cur->value)?cur->ltree:cur->rtree;
        }
        if(temp->value<prev->value)
                prev->ltree=temp;
        else if(temp->value>prev->value)
                prev->rtree=temp;
        return root;
}

void in(NODE IN){
        if(IN!=NULL){
                in(IN->ltree);
                printf("%d\t",IN->value);
                in(IN->rtree);
        }
}

void pre(NODE PRE){
        if(PRE!=NULL){
                printf("%d\t",PRE->value);
                pre(PRE->ltree);
                pre(PRE->rtree);
        }
}

void post(NODE POST){
        if(POST!=NULL){
                post(POST->ltree);
                post(POST->rtree);
                printf("%d\t",POST->value);
        }
}

void search(NODE root){
        int item;
        NODE cur;
        printf("Enter the element to be searched : ");
        scanf("%d",&item);
        if(root==NULL)
                printf("tree is empty\n");
        cur=root;
        while(cur!=NULL){
                if(item==cur->value){
                        printf("Found key %d in tree\n",cur->value);
                        return ;
                }
                if(item<cur->value)
                        cur=cur->ltree;
                else
                        cur=cur->rtree;
        }
        printf("Key not found\n");
}

int main(){
        int choice,item,n,i;
        NODE root=NULL;
        printf( \
                "\n1. Create BST of N Integers \
                 \n2. BST Traversal \
                 \n3. Binary Search \
                 \n4. Exit"
        );
        while(1){
                printf("\n> ");
                scanf("%d",&choice);
                switch(choice){
                        case 1:
                                printf("\nEnter number elements : ");
                                scanf("%d",&n);
                                printf("Enter the item(s) to be inserted : \n");

                                for(i=0;i<n;i++){
                                        scanf("%d",&item);
                                        root=create(item,root);
                                }

                                break;
                        case 2:

                                if(root==NULL)
                                        printf("Tree is empty\n");
                                else{

                                        printf("\n\nPREORDER traversal\n");
                                        pre(root);
                                        printf("\n\nINORDER traversal\n");
                                        in(root);
                                        printf("\n\nPOSTORDER traversal\n");
                                        post(root);
                                }
                                break;
                        case 3:
                                search(root);
                                break;
                        case 4:
                                exit(0);
                        default:
                                printf("\nInvalid Choice !\n");
                }
        }
}
```