Program 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<conio.h>
#include<stdlib.h>
#define MAX 10

struct node
{
int info;
struct node *llink;
struct node *rlink;
};
typedef struct node* NODE;


NODE  getnode()
{
        NODE  x;
        x=(NODE) malloc (sizeof(struct node));
        if(x==NULL)
        {
                printf("out of memory\n");
                exit(0);
        }
        return x;
}

/*****************************************/

NODE insert(int item, NODE root)
{
        NODE temp,cur,prev;
        int i;
        temp=getnode();
        temp->info=item;
        temp->llink=temp->rlink=NULL;
        if(root==NULL)
        {
                root=temp;
                return root;
        }
        else
```

```
	{
		prev=NULL;
		cur=root;
		while(cur!=NULL)
		{
			prev=cur;
			cur=(temp->info<cur->info)?cur->llink:cur->rlink;
		}

		if(temp->info<prev->info)
			prev->llink=temp;
		else
			prev->rlink=temp;
		return root;
	}
}



/*********************************************/

void pre(NODE PRE)
{
	if(PRE!=NULL)
	{
		printf("%d\t",PRE->info);
		pre(PRE->llink);
		pre(PRE->rlink);
	}
}


void in(NODE IN)
{
	if(IN!=NULL)
	{
		in(IN->llink);
		printf("%d\t",IN->info);
		in(IN->rlink);
	}
}

void post(NODE POST)
{
	if(POST!=NULL)
	{
		post(POST->llink);
		post(POST->rlink);
		printf("%d\t",POST->info);
	}
```

```
}

void Traversal(NODE root)
{
        NODE IN,PRE,POST;
        IN=root;
        PRE=root;
        POST=root;
        if(root == NULL)
        {
                printf("tree is empty\n");
                return;
        }
        printf("preorder traversal....\n");
        pre(PRE);
        printf("\nInorder traversal....\n");
        in(IN);
        printf("\n Post order traversal....\n");
        post(POST);

}

 search(NODE root)
{
        int item,i=0;
        NODE cur;
        printf("enter the elemenet to be serached\n");
        scanf("%d", &item);
        if(root == NULL)
        {
                printf("tree is empty\n");
                return;
        }
        cur=root;

        while(cur!=NULL)
        {
                if(item ==cur->info)
                {
                        i++;
                        printf("found key %d in tree\n",cur->info);

                }

                if(item <cur->info)

                        cur=cur->llink;

                else
                        cur=cur->rlink;
```

```
        }
        if(i==0)
        printf("key not found\n");
        return ;
}


void main()
{
        int choice,item;
        NODE root=NULL;

        while(1)
        {
                printf("\n\n\n\t1.create BST .\t2.Traversal..\t3.search....\t4.delete
.....\t5.Exit...");
                printf("\n\n\n\tEnter Your Choice: ");
                scanf("%d",&choice);
                switch(choice)
                {
                        case 1:
                                                printf("enter the item to be inserted\n");
                                                scanf("%d",&item);
                                                root=insert(item,root);
                                                break;
                        case   2: Traversal(root);break;

                        case 3: search(root);break;

                          case 4: exit(0);
                        default: printf("\n\n\n\tEnter proper Choice....");
                }
        }

}
```