Q2] Write a menu driven program to perform the following operations on a BST:

1-Insert
2- In-order traversal
3- Search an element
4- Find minimum
5- Find maximum
6- Exit

```
#include <stdio.h>
#include <stdlib.h>
#include <malloc.h>

struct node
{
   int data;
   struct node *left;
   struct node *right;
} *root = NULL;

struct node *insert(struct node *, int);
void inorder(struct node *);
struct node *search(struct node *, int);
struct node *min(struct node *temp);
struct node *max(struct node *temp);

main()
{
   int ch, x, val;
   while (1)
   {
      printf("\nMenu: \n1: insert\n2: in-order traversal\n 3: Search\n 4:  Minimum\n \n 5: Maximum\n 6: exit\n");
      printf("\n Enter your choice");
      scanf("%d", &ch);
      switch (ch)
      {
      case (1):
         printf("enter the data to insert:");
         scanf("%d", &x);
         root = insert(root, x);
         break;
      case (2):
         inorder(root);
         break;
      case (3):
         printf("Enter element to be searched");
         scanf("%d", &val);
         root = search(root, val);

      case (4):
```

```
            min(root);
            break;
        case (5):
            max(root);
            break;
        case (6):
            exit(0);
            break;
        default:
            printf("Invalid option");
        }
    }
}

struct node *insert(struct node *temp, int ele)
{
    if (temp == NULL)
    {
        temp = (struct node *)malloc(sizeof(struct node));
        temp->data = ele;
        temp->left = NULL;
        temp->right = NULL;
    }
    else
    {
        if (ele < temp->data)
            temp->left = insert(temp->left, ele);
        else
        {
            if (ele > temp->data)
                temp->right = insert(temp->right, ele);
        }
    }
    return temp;
}

void inorder(struct node *p)
{
    if (p != NULL)
    {
        inorder(p->left);
        printf("%d \t", p->data);
        inorder(p->right);
    }
}

struct node *search(struct node *temp, int val)
{
    struct node *p;
    p = temp;
    if (p != NULL && p->data != val)
    {
```

```
        if (val < p->data)
        {
           p = p->left;
        }

        else
        {
           if (val > p->data)
              p = p->right;
        }
    }
    if (p == NULL)
    {
       printf("Element not found");
    }
    else
    {
       return p;
    }
}

struct node *min(struct node *temp)
{
    if (temp == NULL)
       return NULL;
    if (temp->left == NULL)
    {
       return temp;
    }
    else
    {
       return (min(temp->left));
    }
}

struct node *max(struct node *temp)
{
    if (temp == NULL)
       return NULL;
    if (temp->right == NULL)
    {
       return temp;
    }
    else
    {
       return (max(temp->right));
    }
}
```

NAME = ANIRUDH PANDA
ROLL = 16 (D2)

SIC = 190610193