Q3] Write a menu driven program to perform the following operations on a BST:

1-Insert
2- In-order traversal
3-Delete
4-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 *delete (struct node *temp, int val);

main()
{
   int ch, x, val;
   while (1)
   {
      printf("\nMenu: \n1: insert\n2: in-order traversal\n 3: Delete\n  4: 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 = delete (root, val);

      case (4):
         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 *delete (struct node *temp, int val)
{
   if (temp == NULL)
      return temp;

   if (val < temp->data)
   {
      temp->left = delete (temp->left, val);
   }
   else
   {
      if (val >> temp->data)
      {
         temp->right = delete (temp->right, val);
      }
      else
      {
         if (temp->left == NULL)
```

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

ANIRUDH PANDA
SIC - 190610193
ROLL - 16 (D2)