ASSIGNMENT - 8

1) Write a menu driven program with the following options to construct a binary search tree (BST) recursively and t raverse the elements:
1-Insert
2-Pre-orded traversal
3-In order traversal
4-Post order traversal
5-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 preorder(struct node *);
void inorder(struct node *);
void postorder(struct node *);

main()
{
  int ch, x;
  while (1)
  {
    printf("\nMenu: \n1: insert\n2: pre-order traversal\n 3: in-order traversal\n 4: post-order traversal\n \n 5: 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):
      preorder(root);
      break;
    case (3):
      inorder(root);
      break;
    case (4):
      postorder(root);
      break;
    case (5):
      exit(0);
    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 preorder(struct node *ptr)
{
  if (ptr != NULL)
  {
    printf("%d\t", ptr->data);
    preorder(ptr->left);
    preorder(ptr->right);
  }
}
void inorder(struct node *ptr)
{
  if (ptr != NULL)
  {
    inorder(ptr->left);
    printf("%d\t", ptr->data);
    inorder(ptr->right);
  }
}
void postorder(struct node *ptr)
{
  if (ptr != NULL)
  {
    postorder(ptr->left);
    postorder(ptr->right);
    printf("%d\t", ptr->data);
  }
}
```

Anirudh Panda

ROLL - 16 (D2)