Date __/__/__ Lab Program 10

Q) Write a program
a) To construct a Binary Search Tree
b) To traverse the tree using all the methods i.e., in order, preorder and postorder
c) To display the elements in the tree.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
struct node
{
    int info;
    struct node *rlink;
    struct node *llink;
};
typedef struct node *NODE;
NODE getnode()
{
    NODE x;
    x = (NODE) malloc (sizeof (struct node));
    if (x == NULL)
    {
        printf ("mem full\n");
        exit(0);
    }
    return x;
}
void freenode( NODE x)
{
    free (x);
}
```

Scanned with CamScanner

Date __/__/____

```
NODE insert ( NODE root, int item)
{
    NODE temp, cur, prev;
    temp = getnode ();
    temp -> rlink = NULL;
    temp -> llink = NULL;
    temp -> info = item;
    if ( root == NULL)
    return temp;
    prev = NULL;
    cur = root;
    while ( cur != NULL)
    {
      prev = cur;
      cur = (item < cur -> info) ? cur -> llink :
                                   cur -> rlink;
    }
  if (item < prev -> info)
  prev -> llink = temp;
  else
   prev -> rlink = temp;
   return root;
}
void display ( NODE root , int i)
{
  int j;
  if ( root != NULL)
  {
     display ( root -> rlink ; i+1);
     for ( j = 0; j < i; j++)
     printf ( "  ");
```

Scanned with CamScanner

```
        printf("%d\n", root->info);
        display(root->llink, i+1);
    }
}
void preorder(NODE root)
{
    if(root != NULL)
    {
        printf("%d\n", root->info);
        preorder(root->llink);
        preorder(root->rlink);
    }
}
void postorder(NODE root)
{
    if(root != NULL)
    {
        postorder(root->llink);
        postorder(root->rlink);
        printf("%d\n", root->info);
    }
}
void inorder(NODE root)
{
    if(root != NULL)
    {
        inorder(root->llink);
        printf("%d\n", root->info);
        inorder(root->rlink);
    }
}
```

Scanned with CamScanner

Date ___/___/_____

```
void main ()
{
  int item, choice;
  NODE root = NULL;
  for (;;)
  {
    printf ("\n 1. insert \n 2. preorder \n 3. postorder
          \n 4. inorder \n 5. display \n 6. exit \n");
  printf (" enter the choice \n");
  scanf (" %d", &choice);
  printf ("------\n");
  switch (choice)
  {
    case 1: printf (" enter the item \n");
          scanf (" %d", &item);
          root = insert (root, item);
          break;
    case 2: preorder (root);
          break;
    case 3: postorder (root);
          break;
    Case 4: inorder (root);
          break;
    case 5: display (root, 0);
          break;
    default: exit (0);
          break;
  }
  }
}
```

Scanned with CamScanner