**EXPERIMENT – 09**

**AIM:** To implement Binary Search Tree ADT and operation (Insert, Traverse) using Linked List.

**Program:**

#include<stdio.h>

#include<stdlib.h>

struct node

{

int data;

struct node\* left;

struct node\* right;

};

struct node\* root = NULL;

void insert(int d)

{

struct node\* t,\* parent;

t = (struct node\*)malloc(sizeof(struct node));

t->data = d;

t->left = NULL;

t->right = NULL;

parent = root;

if(root == NULL)

{

root = t;

}

else

{

struct node\* current;

current = root;

while(current)

{

parent = current;

if(t->data > current->data)

{

current = current->right;

}

else

{

current = current->left;

}

}

if(t->data > parent->data)

{

parent->right = t;

}

else

{

parent->left = t;

}

}

printf("\n%d was inserted successfully!\n", d);

}

void inorder(struct node\* t)

{

if (t != NULL) {

inorder(t->left);

printf("\t%d\t", t->data);

inorder(t->right);

}

}

void preorder( struct node \*t)

{

if (t != NULL)

{

printf("\t%d\t", t->data);

preorder(t->left);

preorder(t->right);

}

}

void postorder(struct node \*t)

{

if (t != NULL)

{

postorder(t->left);

postorder(t->right);

printf("\t%d\t", t->data);

}

}

int main()

{

int choice, d;

while(1)

{

printf("\n ---Main Menu---\n");

printf("\n1. Insert\n2. Preorder Traverse\n3. Inorder Traverse\n4. Postorder Traverse\n5. Exit\n");

printf("Enter your choice: ");

scanf("%d", &choice);

switch(choice)

{

case 1: printf("\nEnter element to be inserted: ");

scanf("%d", &d);

insert(d);

printf("\n");

break;

case 2: printf("\nElements in Preorder:");

preorder(root);

printf("\n");

break;

case 3: printf("\nElements in Inorder:");

inorder(root);

printf("\n");

break;

case 4: printf("\nElements in Postorder:");

postorder(root);

printf("\n");

break;

case 5: exit(0);

default: printf("Invalid Choice");

}

}

return 0;

}

**OUTPUT:**



