**Experiment No 6**

Name:- Anuj Rajendra Mane

ROll No:- 65

Div:-A

Subject:- Data Structures

Title: Implementation of binary search Trees.

**Problem Statements:**

**1) Write a program to implement Binary Search Tree with following operations**

**a. Create Node**

**b. Insert node**

**c. Traverse Tree**

**In-order Traversal, Pre-order Traversal and Post-order Traversal**

#include <stdio.h>

#include <stdlib.h>

struct node

{

int data;

struct node \*left;

struct node \*right;

};

struct node \*createNode(int val)

{

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

n->data = val;

n->right = NULL;

n->left = NULL;

return n;

}

struct node \*insert(struct node \*root, int val)

{

if (root == NULL)

return createNode(val);

if (val < root->data)

root->left = insert(root->left, val);

if (val > root->data)

root->right = insert(root->right, val);

return root;

}

void inorder(struct node \*root)

{

if (root == NULL)

return;

inorder(root->left);

printf("%d -> ", root->data);

inorder(root->right);

}

void preorder(struct node \*root)

{

if (root == NULL)

{

return;

}

printf("%d ->", root->data);

preorder(root->left);

preorder(root->right);

}

void postorder(struct node \*root)

{

if (root == NULL)

{

return;

}

postorder(root->left);

postorder(root->right);

printf("%d ->", root->data);

}

int main()

{

struct node \*root = NULL;

root = insert(root,100);

insert(root, 120);

insert(root, 50);

insert(root, 20);

insert(root, 12);

insert(root, 70);

insert(root, 200);

insert(root, 110);

int ch;

while(1)

{

printf("\n\nEnter 1: For Inorder\t2:For Preorder\t3:For Postorder:");

scanf("%d",&ch);

switch(ch)

{

case 1:

inorder(root);

break;

case 2:

preorder(root);

break;

case 3:

postorder(root);

break;

case 4:

exit(0);

break;

default:

printf("Invalid Choice\n");

}

}

}