

EXPERIMENT 8

Binary Search Tree

Program:

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>

struct node
{
    int data; struct
node
    *right,*left;
}*root=NULL;

struct node* createNode(int x)
{
    struct    node*    newNode;
    newNode=(struct
node*)malloc(sizeof(struct    node));
    if(newNode==NULL)
```

```

{
    printf("Cannot Create Node For %d
data\n",x);
}
else
{
    newNode->data=x;
newNode->right=NULL;
newNode->left=NULL;

}
return newNode;
}

```

```

struct node* insert(struct node* r,struct
node* t)
{
    if(r==NULL || r->data==t->data)
    {
        r=t;
return r;
    }
    else if(t->data<r->data)
    {
        r->left=insert(r->left,t);

```

```
    }  
    else  
    {  
        r->right=insert(r->right,t);  
    }  
    return r;  
}
```

```
void inorder(struct node* r)  
{  
    if(r==NULL)  
    {  
        return;  
    }  
    else  
    {  
        inorder(r->left);  
        printf("%d\t",r->data);  
        inorder(r->right);  
    }  
}
```

```
void preorder(struct node* r)
```

```
{
if(r==NULL)
{
return;
}
else
{
printf("%d\t",r->data);
preorder(r->left);
preorder(r->right);
}
}
```

```
void postorder(struct node* r)
{
if(r==NULL)
{
return;
}
else
{
postorder(r->left); postorder(r->right);
printf("%d\t",r->data);
}
```

```
}  
}
```

```
void main()  
{  
    int data,ch;  
    struct node  
    *temp;  
    clrscr(); printf("Abhay Gori\t SE-04\t  
Roll No.11\n\n"); printf("Enter Your  
Elements For Binary Search Tree\n\n\n");  
    printf("Enter Root Element\n");  
    scanf("%d",&data);  
    root=createNode(data); printf("Enter  
Your remaining Elements For BST\n");  
    printf("Enter Value '-0' to stop creating  
new Nodes\n"); while(data!=-0)  
    {  
        printf("Enter Your  
Element\n");  
        scanf("%d",&data);  
        if(data==0)  
        {
```

```

        break;
    }
    temp=createNode(data);
    root=insert(root,temp);
}
printf("                *****
OPERATIONS  *****  \n");
printf("1. Insert  an  Element\n");
printf("2. Inorder Traversal\n");
printf("3. Pre-Order
Traversal\n"); printf("4.
Post Order
Traversal\n");
printf("Enter Your
Choice\n");
scanf("%d",&ch);
switch(ch)
{
    case 1:printf("Enter Your Element to be
inserted\n");
        scanf("%d",&data);
        temp=createNode(data);
        root=insert(root,temp);

```

```
printf("Updated BST is:\n\n");  
inorder(root);  
    break;  
case 2:inorder(root);  
    break;  
case 3:preorder(root);  
    break;  
case 4:postorder(root);  
    break;  
default:printf("Wrong Choice Nigga\n");  
}  
getch();  
}
```

Output:

```

BST

Binary Search Tree By Aayush Joshi SE4-14

Enter Your Elements For Binary Search Tree

Enter Root Element
50
Enter Your remaining Elements For BST
Enter Value '-0' to stop creating new Nodes
Enter Your Element
30
Enter Your Element
60
Enter Your Element
70
Enter Your Element
80
Enter Your Element
-0

***** OPERATIONS *****
1. Insert an Element
2. Inorder Traversal
3. Pre-Order Traversal
4. Post Order Traversal
5. Exit
Enter Your Choice
1
Enter Your Element to be inserted
90
Updated BST is:
30      50      60      70      80      90
***** OPERATIONS *****
1. Insert an Element
2. Inorder Traversal
3. Pre-Order Traversal
4. Post Order Traversal
5. Exit
Enter Your Choice
2
30      50      60      70      80      90
***** OPERATIONS *****
***** OPERATIONS *****
1. Insert an Element
2. Inorder Traversal
3. Pre-Order Traversal
4. Post Order Traversal
5. Exit
Enter Your Choice
3
50      30      60      70      80      90
***** OPERATIONS *****
1. Insert an Element
2. Inorder Traversal
3. Pre-Order Traversal
4. Post Order Traversal
5. Exit
Enter Your Choice
4
30      90      80      70      60      50
***** OPERATIONS *****
1. Insert an Element
2. Inorder Traversal
3. Pre-Order Traversal
4. Post Order Traversal
5. Exit
Enter Your Choice
5
Press any key to continue . . .
```