Exp no. 6

input

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


struct node
{
int data;
struct node *left;
struct node *right;
};

struct node *tree;
void create (struct node *,int);
void inorder(struct node *);
void preorder(struct node *);
void postorder(struct node *);

void main()

{
int choice,x;
struct node *ptr;
create(tree);
do
{
printf("\n 1.insert a node");
printf("\n2. display inorder traversal");

printf("\n3. display preorder traversal");

printf("\n4. display postorder traversal");
```

```
printf("\n5.exit");
printf("enter your choice\n");
scanf("%d",&choice);

switch(choice)
{
case 1:

printf("\n enter the dta inserted");
scanf("%d",&x);
tree = insert(tree,x);
break;
case 2:

printf("\n elements in inorder traversal are\n");
inorder(tree);
break;
case 3:
printf("\n elements in preorder traversal are\n");
preorder(tree);
break;
case 4:
printf("\n elements in postorder traversal are\n");
postorder(tree);
break;
case 5:
printf("\n exit the program");
break;
default:
printf("\n error \n");

break;
}
}while(choice != 5);
}

void create(struct node *tree)
```

```
{
tree= NULL;
}

struct node *insert(struct node *tree,int x)
{
struct node *p,*temp,*root;
p=(struct node *)malloc(sizeof(struct node));
p->data = x;
p->left = NULL;
p->right = NULL;
if (tree== NULL)
{
tree =p;
tree -> left=NULL;
tree -> right=NULL;
}
else
{
root= NULL;
temp = tree;
while(temp!= NULL)
{
root = temp;
if(x< temp->data)
temp= temp-> left;
else
temp = temp->right;
}
if(x< root->data)
root->left= p;
else
root->right = p;
}
return tree;
}
```

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

void preorder(struct node *tree)
{
if(tree!= NULL)
{
printf("%d\t",tree->data);
preorder(tree-> left);
preorder(tree-> right);
}
}

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

```
Operations available are :
1. Insert a node
2. Display inorder traversal
3. Display preorder traversal
4. Display postorder traversal
5. Exit

Enter your choice:2

Elements in the inorder traversal are:2        5       7       8       9

Operations available are :
1. Insert a node
2. Display inorder traversal
3. Display preorder traversal
4. Display postorder traversal
5. Exit

Enter your choice:3

Elements in the preorder traversal are:5       2       7       9       8

Operations available are :
1. Insert a node
2. Display inorder traversal
3. Display preorder traversal
4. Display postorder traversal
5. Exit

Enter your choice:4

Elements in the postorder traversal are:2      8       9       7       5
```