15/02/24

Week-8

```
#include<stdio.h>
#include <stdlib.h>
struct node
{
  int value;
  struct node *left;
  struct node *right;
} node;

struct node *create()
{
   struct node *temp;
   temp=(struct node*) malloc(sizeof(struct
                     node));
   printf("Enter data: ");
   scanf("%d", &temp->value);
   temp->left=temp->right=NULL;
   return temp;
}
void insert(struct node *root, struct node
      *temp)
 {
   if(temp->value<root->value)
   {
     if(root->left!=NULL)
         insert(root->left, temp);
     else
         root->left=temp;
   }
   if(temp->value>root->value)
  {
     if (root -> right != NULL)
          insert(root -> right, temp);
     else
          root -> right = temp;
  }
 }
void inorder(struct node *root)
{
    if (root != NULL)
    {
        inorder(root -> left);
        printf("%d \t", root -> value);
        inorder(root -> right);
    }
 }
void postorder(struct node *root)
{
    if (root != NULL)
    {
        postorder(root -> left);
        postorder( ~~rig~~ root -> right);
        printf("%d \t", root -> value);
    }
 }
void preorder(struct node *root)
{
    if(root != NULL)
    {
        printf("%d \t", root -> value);
        preorder(root -> left);
        preorder(root -> right);
```

```
      }
    }
int main()
{
   int choice;
   struct node *root=NULL;
   while(1)
   {
        printf("\n 1.Insert the element \n
              2.Inorder traversal \n 3.Preorder
              traversal \n 4.Post order traversal
               \n 5.exit");
        printf("Enter choice \n");
        scanf("%d", &choice);
        switch(choice)
        {
        case 1:
             if (root==NULL)
                  root=create();
             else
                 insert(root, create());
             break;
        case 2:
             printf("\n Inorder traversal \n");
             inorder(root);
             break;
        case 3:
             printf(" \n Preorder traversal \n");
             preorder(root);
             break;
        case 4:
             printf("\n Postorder traversal \n");
```

```
            postorder(root);
            break;
        case 5:
            exit(0);
        default:
            printf("Invalid choice");
            break;
        }
    }
    return 0;
}
```

output

```
1. Insert the element
2. In-order traversal
3. Pre-order traversal
4. Post-order traversal
5. exit.
Enter choice : 1
Enter data : 50

Enter choice : 1
Enter data : 60

Enter choice : 1
Enter data : 70

Enter choice : 1
Enter data : ~~20~~90

Enter choice : 1
```

15/2/24

Enter data:10.

Enter choice:2.

Inorder traversal: 10, 50, 60, 70, 90

~~Preorder traversal: 50.~~

Enter choice:3

Preorder traversal:50,10,60,70,90.

Enter choice:4

Postorder traversal:10, 90,70,60,50.