**Programs**

1. Write a program to create and traverse through Binary Tree.

Ans:

#include<stdio.h>

#include<stdlib.h>

#include<conio.h>

struct node

{

int data;

struct node \*left,\*right;

};

struct node \*create();

void preorder(struct node \*root);

void inorder(struct node \*root);

void postorder(struct node \*root);

void main()

{

struct node \*root;

int ch;

clrscr();

while(ch!=5)

{

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

printf("\n1.Create node \t2.Inorder \t3.Preorder \t4.Postorder \t5.Exit");

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

printf("\nEnter your choice:");

scanf("%d",&ch);

switch(ch)

{

case 1:root=create();

break;

case 2:inorder(root);

break;

case 3:preorder(root);

break;

case 4:postorder(root);

break;

case 5:exit(0);

default:printf("Invalid choice..");

}

}

}

struct node \*create()

{

struct node \*temp;

int num;

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

printf("Enter data (Press 0 for no node): ");

scanf("%d",&num);

if(num==0)

return(0);

temp->data=num;

printf("Enter left child of %d\n",num);

temp->left=create();

printf("Enter right child of %d\n",num);

temp->right=create();

return(temp);

}

void preorder(struct node \*root)

{

if(root!=NULL)

{

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

preorder(root->left);

preorder(root->right);

}

}

void inorder(struct node \*root)

{

if(root!=NULL)

{

inorder(root->left);

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

inorder(root->right);

}

}

void postorder(struct node \*root)

{

if(root!=NULL)

{

postorder(root->left);

postorder(root->right);

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

}

}

Output:

