/* Binary Tree & its traversals */

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
struct bt
{
	struct bt *left;
	int key;
	struct bt *right;
}*root=NULL,*new1,*temp;

void create();
void insert(struct bt *,struct bt *);
void inorder(struct bt*);
void preorder(struct bt*);
void postorder(struct bt*);

void create()
{
	int ele;
	new1=(struct bt*)malloc(sizeof(struct bt));
	new1->left=NULL;
	new1->right=NULL;
	printf("enter data ");
	scanf("%d",&ele);
	new1->key=ele;
	if(root==NULL)
		root=new1;
	else
		insert(root,new1);
}
void insert(struct bt *root,struct bt *new1)
{
	char ch;
	printf("enter l toinsert left child or r for right child\n");
```

```
        fflush(stdin);
        scanf("%c",&ch);
        switch(ch)
        {
                case 'l':
                 if(root->left==NULL)
                        root->left=new1;
                 else
                        insert(root->left,new1);
                        break;
                case 'r':if(root->right==NULL)
                        root->right=new1;
                        else
                        insert(root->right,new1);
                        break;

        }
}
void inorder(struct bt *temp)
{
        if(temp!=NULL)
        {
                inorder(temp->left);
                printf(" %d",temp->key);
                inorder(temp->right);
        }
}

void preorder(struct bt *temp)
{
        if(temp!=NULL)
        {
                printf(" %d",temp->key);
                preorder(temp->left);
                preorder(temp->right);
```

```
        }
}

void postorder(struct bt *temp)
{

        if(temp!=NULL)
        {
                postorder(temp->left);
                postorder(temp->right);
                printf(" %d",temp->key);
        }
}

void main()
{
        int choice,ele;
        while(1)
        {
                clrscr();
                printf("\t \t  binary Tree");
                printf("\n 1.create &  insert");
                printf("\n 2.ioorder traversal");
                printf("\n 3.preorder traversal");
                printf("\n 4.post order traversal");
                printf("\n 5 exit");
                printf("\nenter choic");
                scanf("%d",&choice);
                switch(choice)
                {
                        case 1:create();
                                break;
                        case 2:inorder(root);
                                getch();
                                break;
```

```
                    case 3:preorder(root);
                           getch();
                           break;
                    case 4:postorder(root);
                           getch();
                           break;

                    case 5:exit(0);
                }
        }
}
```