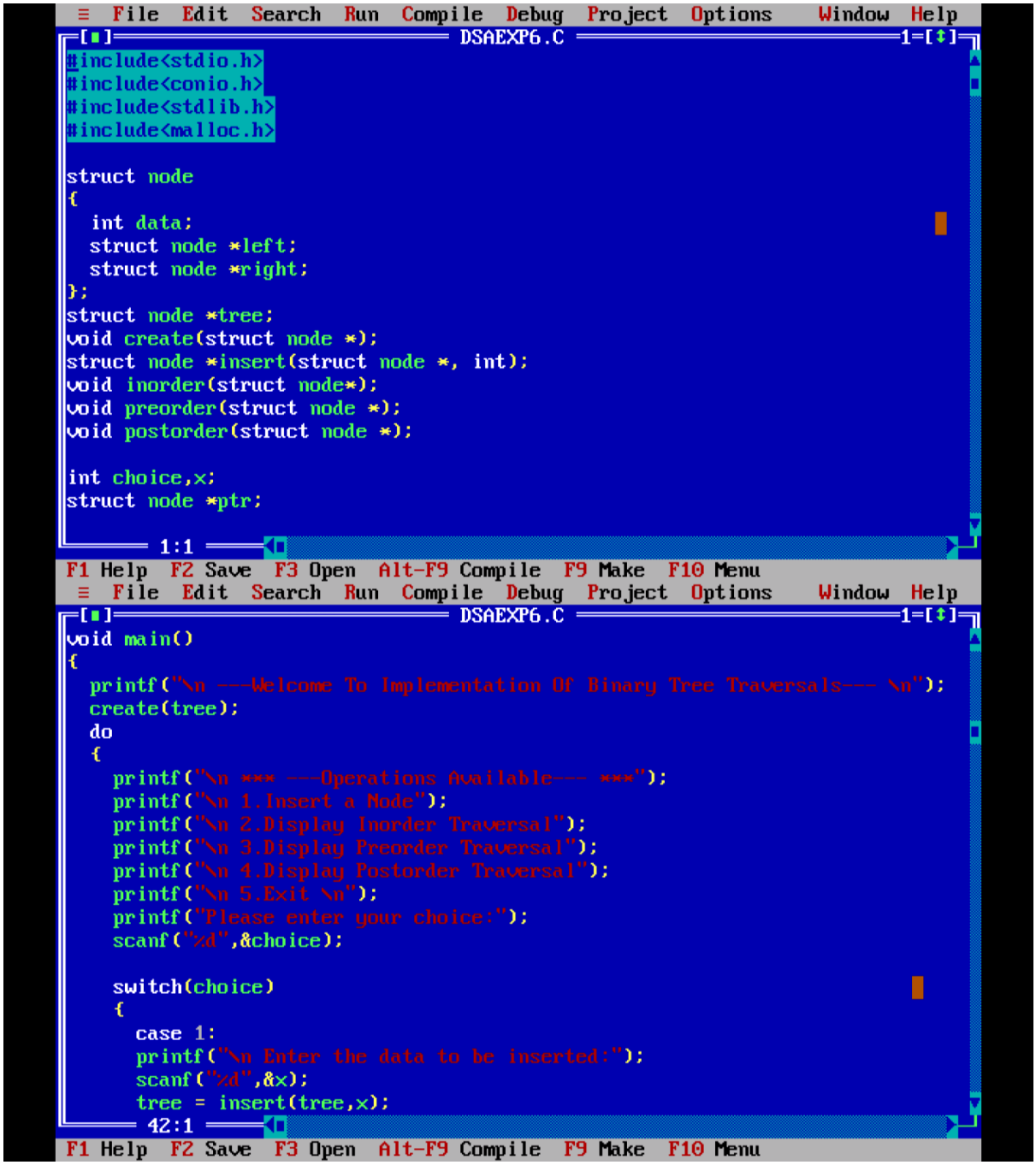
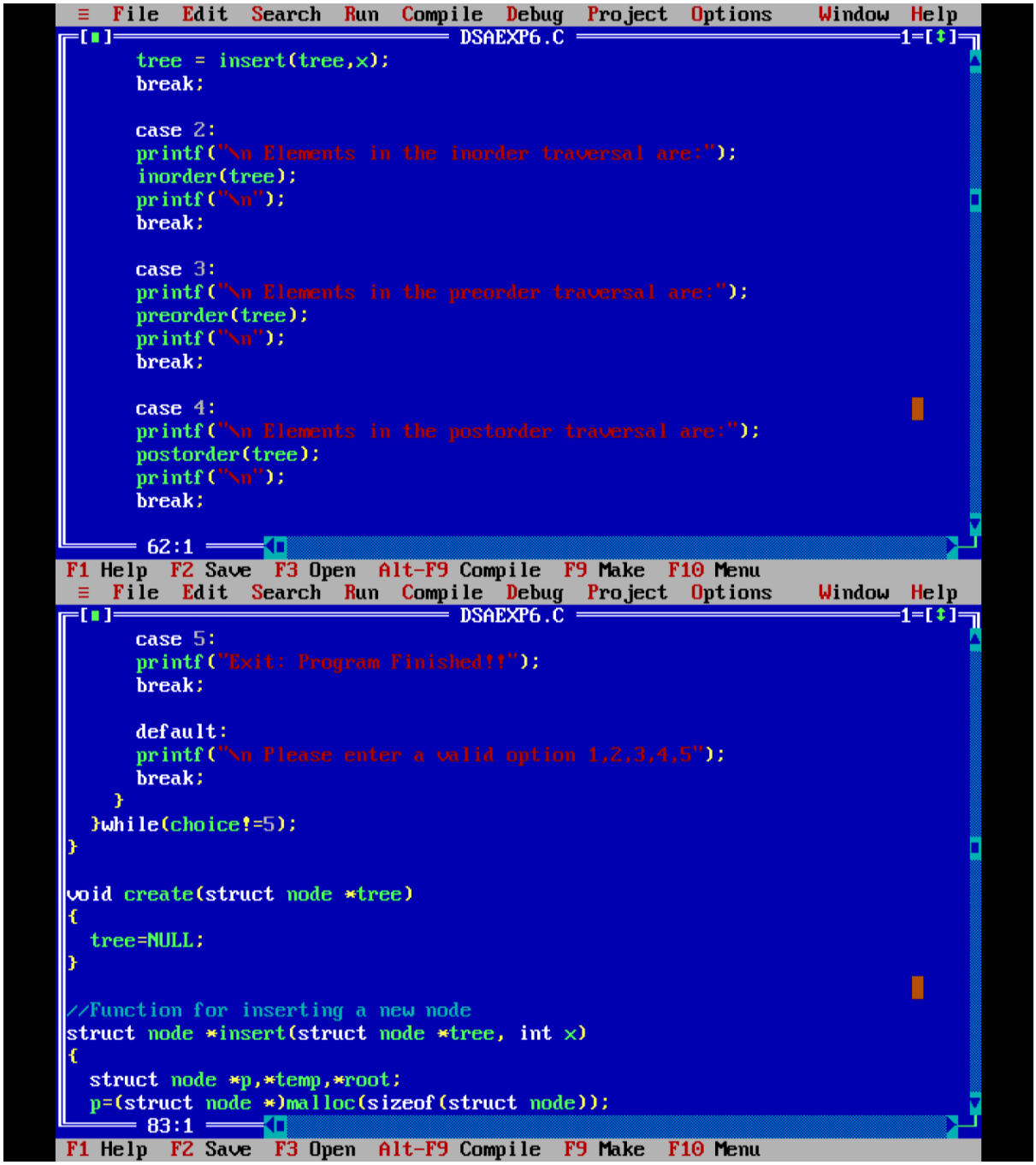
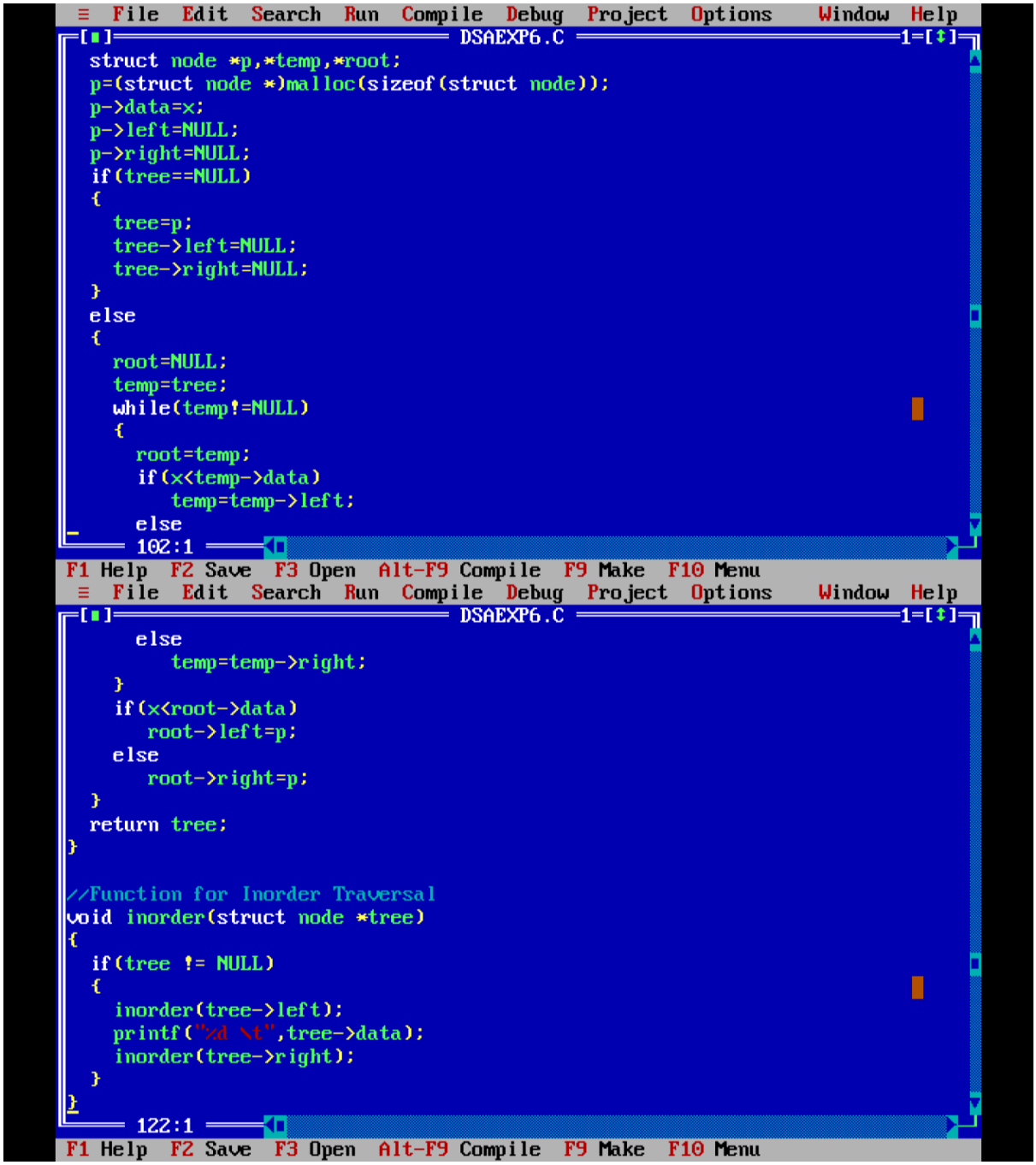
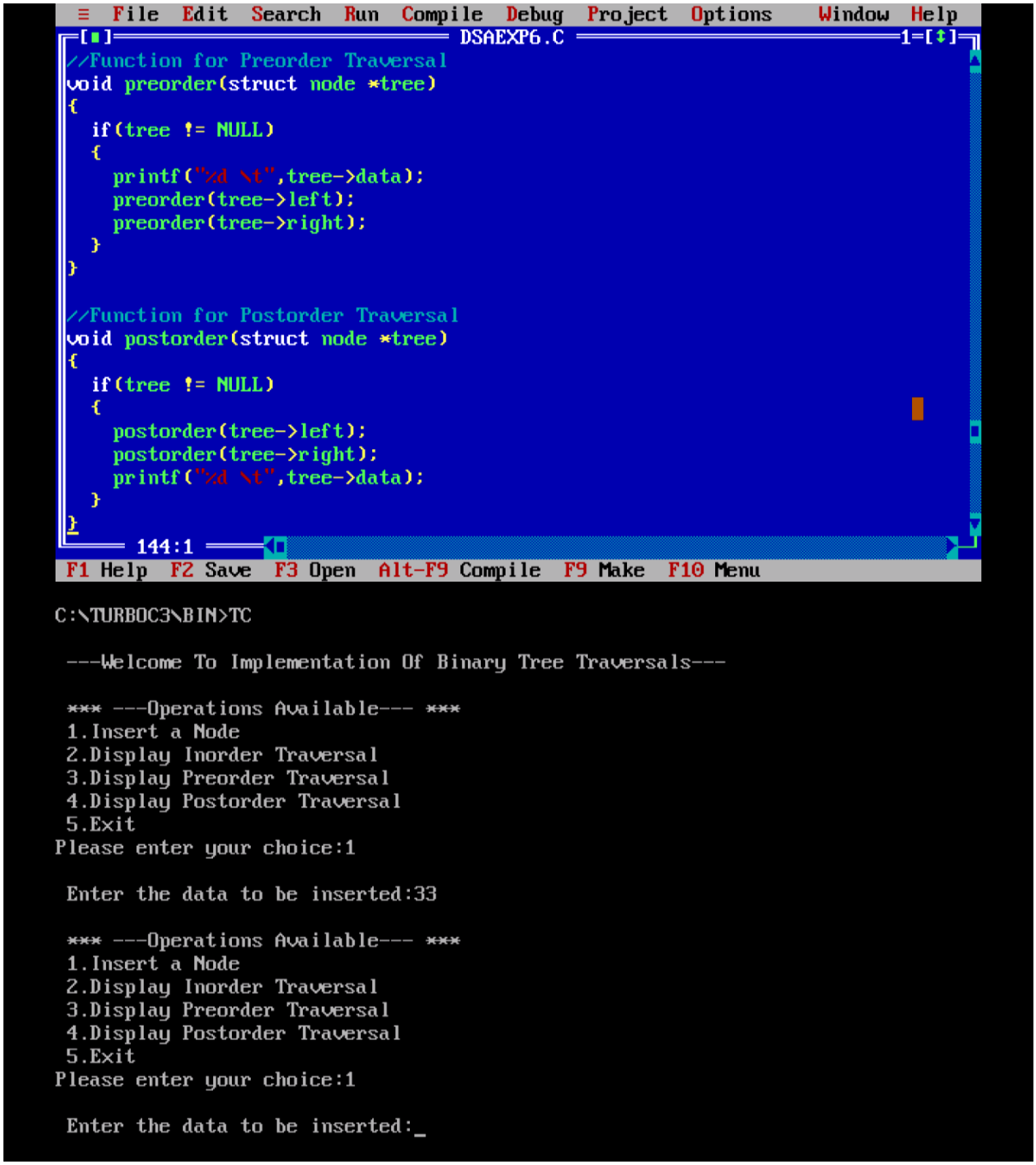

File Edit Search Run Compile Debug Project Options Window Help
DSAEXP6.C
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<malloc.h>
struct node
{
int data;
struct node *left;
struct node *right;
};
struct node *tree;
void create(struct node *);
struct node *insert(struct node *, int);
void inorder(struct node*);
void preorder(struct node *);
void postorder(struct node *);
int choice,x;
struct node *ptr;
1:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu
File Edit Search Run Compile Debug Project Options Window Help
DSAEXP6.C
void main()
{
printf("\n ---Welcome To Implementation Of Binary Tree Traversals--- \n");
create(tree);
do
{
printf("\n *** ---Operations Available--- ***");
printf("\n 1.Insert a Node");
printf("\n 2.Display Inorder Traversal");
printf("\n 3.Display Preorder Traversal");
printf("\n 4.Display Postorder Traversal");
printf("\n 5.Exit \n");
printf("Please enter your choice:");
scanf("%d",&choice);
switch(choice)
{
case 1:
printf("\n Enter the data to be inserted:");
scanf("%d",&x);
tree = insert(tree,x);
42:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu


File Edit Search Run Compile Debug Project Options Window Help
DSAEXP6.C
1=[↕]
tree = insert(tree,x);
break;
case 2:
printf("\n Elements in the inorder traversal are:");
inorder(tree);
printf("\n");
break;
case 3:
printf("\n Elements in the preorder traversal are:");
preorder(tree);
printf("\n");
break;
case 4:
printf("\n Elements in the postorder traversal are:");
postorder(tree);
printf("\n");
break;
62:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu
File Edit Search Run Compile Debug Project Options Window Help
DSAEXP6.C
1=[↕]
case 5:
printf("Exit: Program Finished!!");
break;
default:
printf("\n Please enter a valid option 1,2,3,4,5");
break;
}
}while(choice!=5);
}
void create(struct node *tree)
{
tree=NULL;
}
//Function for inserting a new node
struct node *insert(struct node *tree, int x)
{
struct node *p,*temp,*root;
p=(struct node *)malloc(sizeof(struct node));
83:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu


File Edit Search Run Compile Debug Project Options Window Help
DSAEXP6.C
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
102:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu
File Edit Search Run Compile Debug Project Options Window Help
DSAEXP6.C
else
temp=temp->right;
}
if(x<root->data)
root->left=p;
else
root->right=p;
}
return tree;
}
//Function for Inorder Traversal
void inorder(struct node *tree)
{
if(tree != NULL)
{
inorder(tree->left);
printf("%d \t",tree->data);
inorder(tree->right);
}
}
122:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu


File Edit Search Run Compile Debug Project Options Window Help
DSAEXP6.C
//Function for Preorder Traversal
void preorder(struct node *tree)
{
if(tree != NULL)
{
printf("%d \t",tree->data);
preorder(tree->left);
preorder(tree->right);
}
}
//Function for Postorder Traversal
void postorder(struct node *tree)
{
if(tree != NULL)
{
postorder(tree->left);
postorder(tree->right);
printf("%d \t",tree->data);
}
}
144:1
F1 Help F2 Save F3 Open Alt-F9 Compile F9 Make F10 Menu
C:\TURBOC3\BIN>TC
---Welcome To Implementation Of Binary Tree Traversals---
*** ---Operations Available--- ***
1.Insert a Node
2.Display Inorder Traversal
3.Display Preorder Traversal
4.Display Postorder Traversal
5.Exit
Please enter your choice:1
Enter the data to be inserted:33
*** ---Operations Available--- ***
1.Insert a Node
2.Display Inorder Traversal
3.Display Preorder Traversal
4.Display Postorder Traversal
5.Exit
Please enter your choice:1
Enter the data to be inserted:_

```
 1.Insert a Node
 2.Display Inorder Traversal
 3.Display Preorder Traversal
 4.Display Postorder Traversal
 5.Exit
Please enter your choice:1

 Enter the data to be inserted:56

 *** ---Operations Available--- ***
 1.Insert a Node
 2.Display Inorder Traversal
 3.Display Preorder Traversal
 4.Display Postorder Traversal
 5.Exit
Please enter your choice:55

 Please enter a valid option 1,2,3,4,5
 *** ---Operations Available--- ***
 1.Insert a Node
 2.Display Inorder Traversal
 3.Display Preorder Traversal
 4.Display Postorder Traversal
 5.Exit
Please enter your choice:_
 1.Insert a Node
 2.Display Inorder Traversal
 3.Display Preorder Traversal
 4.Display Postorder Traversal
 5.Exit
Please enter your choice:55

 Please enter a valid option 1,2,3,4,5
 *** ---Operations Available--- ***
 1.Insert a Node
 2.Display Inorder Traversal
 3.Display Preorder Traversal
 4.Display Postorder Traversal
 5.Exit
Please enter your choice:1

 Enter the data to be inserted:56

 *** ---Operations Available--- ***
 1.Insert a Node
 2.Display Inorder Traversal
 3.Display Preorder Traversal
 4.Display Postorder Traversal
 5.Exit
Please enter your choice:
```

```
 2.Display Inorder Traversal
 3.Display Preorder Traversal
 4.Display Postorder Traversal
 5.Exit
Please enter your choice:1

 Enter the data to be inserted:56

 *** ---Operations Available--- ***
 1.Insert a Node
 2.Display Inorder Traversal
 3.Display Preorder Traversal
 4.Display Postorder Traversal
 5.Exit
Please enter your choice:1

 Enter the data to be inserted:78

 *** ---Operations Available--- ***
 1.Insert a Node
 2.Display Inorder Traversal
 3.Display Preorder Traversal
 4.Display Postorder Traversal
 5.Exit
Please enter your choice:
```

```
 2.Display Inorder Traversal
 3.Display Preorder Traversal
 4.Display Postorder Traversal
 5.Exit
Please enter your choice:2

 Elements in the inorder traversal are:33      56      56      78

 *** ---Operations Available--- ***
 1.Insert a Node
 2.Display Inorder Traversal
 3.Display Preorder Traversal
 4.Display Postorder Traversal
 5.Exit
Please enter your choice:3

 Elements in the preorder traversal are:33     56      56      78

 *** ---Operations Available--- ***
 1.Insert a Node
 2.Display Inorder Traversal
 3.Display Preorder Traversal
 4.Display Postorder Traversal
 5.Exit
Please enter your choice:
 2.Display Inorder Traversal
 3.Display Preorder Traversal
 4.Display Postorder Traversal
 5.Exit
Please enter your choice:3

 Elements in the preorder traversal are:33     56      56      78

 *** ---Operations Available--- ***
 1.Insert a Node
 2.Display Inorder Traversal
 3.Display Preorder Traversal
 4.Display Postorder Traversal
 5.Exit
Please enter your choice:4

 Elements in the postorder traversal are:78    56      56      33

 *** ---Operations Available--- ***
 1.Insert a Node
 2.Display Inorder Traversal
 3.Display Preorder Traversal
 4.Display Postorder Traversal
 5.Exit
Please enter your choice:_
```