Lab Program - 10

10) Write a program
a) To construct a binary search tree.
b) To traverse the tree using all methods i.e, in-order, preorder & post-order.
c) To display the elements in the tree.

A.

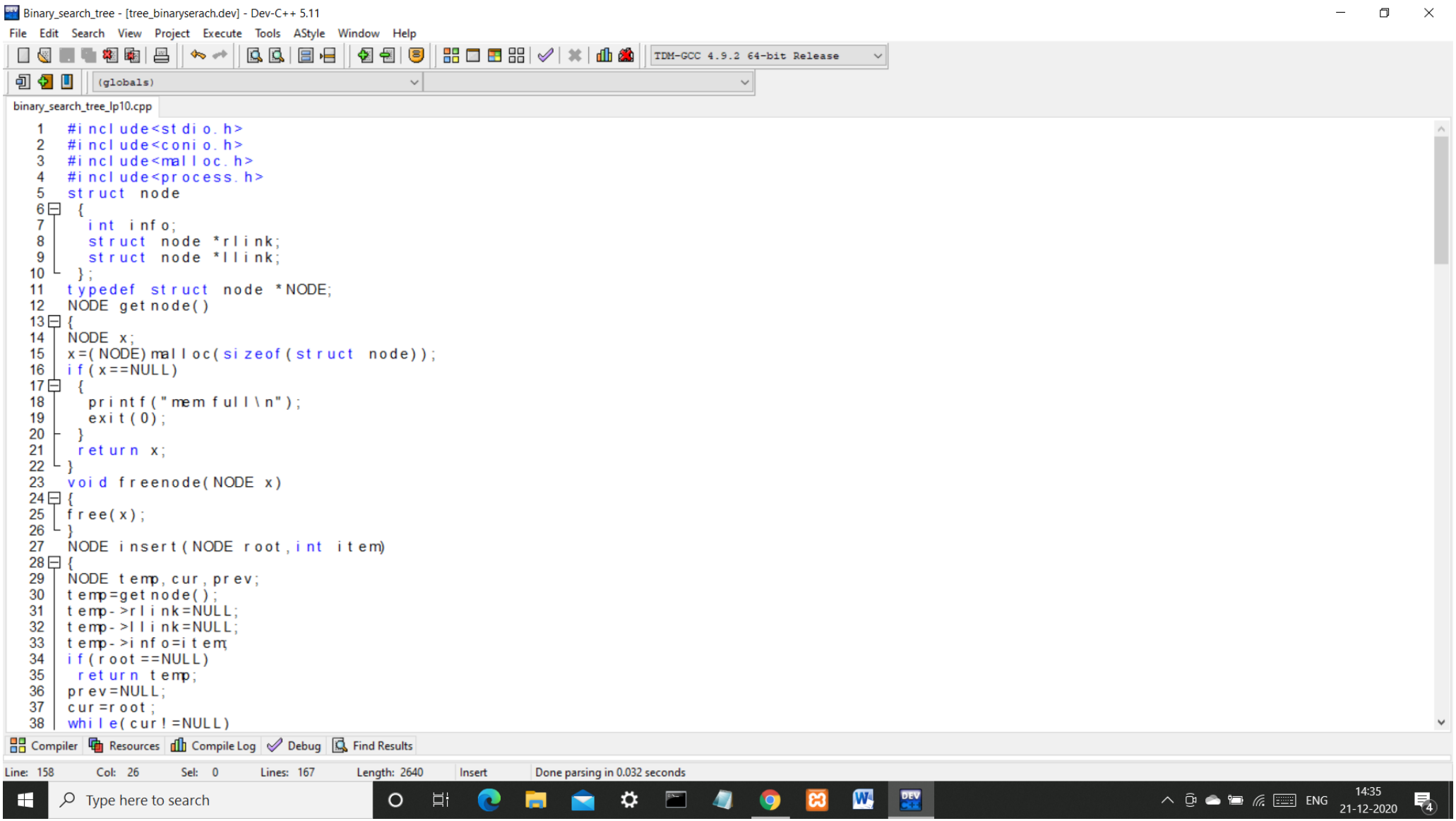
```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
struct node{
int info;
struct node *llink;
struct node *rlink;
};
typedef struct node *NODE;
NODE getnode(){
NODE x;
x=(NODE)malloc(sizeof(struct node));
if(x==NULL){
printf("Memory not available\n");
exit(0);
}
return x;
}
void freenode(NODE x){
free(x);
}
NODE insert(int item, NODE root){
NODE temp, cur, prev;
char direction[10];
int i;
temp=getnode();
```

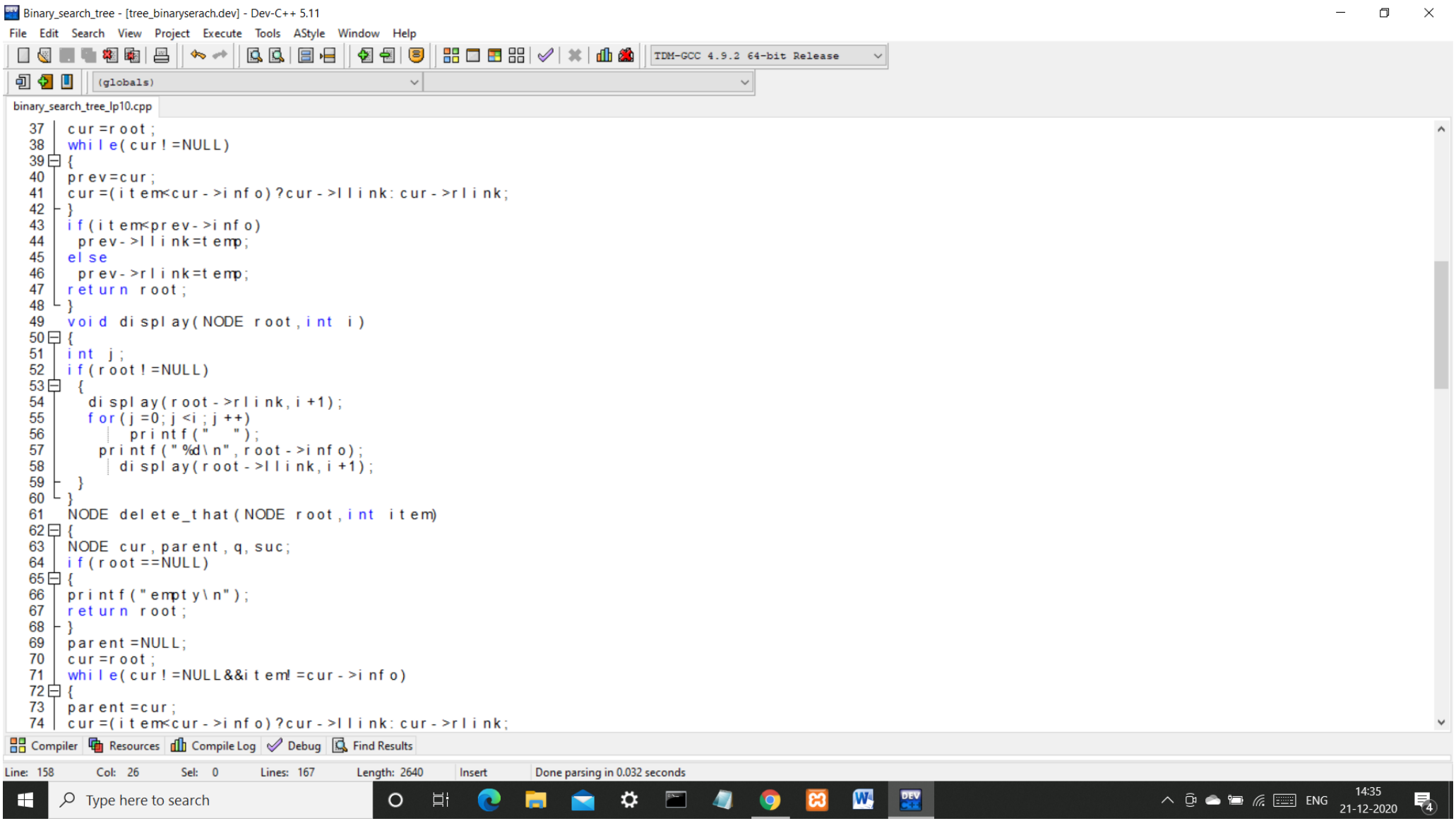
```
temp->info = item;
temp->llink = NULL;
temp->rlink = NULL;
if(root == NULL)
return temp;
printf("Give direction to insert \n");
scanf("%s", direction);
prev = NULL;
cur = root;
for(i=0; i<strlen(direction) && cur != NULL; i++){
prev = cur;
if(direction[i] == 'l');
cur = cur->llink;
else
cur = cur->rlink;
}
if(cur != NULL || i != strlen(direction)){
printf("Insertion not possible \n");
freenode(temp);
return(root);
}
if(cur == NULL){
if(direction[i-1] == 'l')
prev->llink = temp;
else
prev->rlink = temp;
}return(root);
}
void preorder(NODE root){
if(root != NULL){
printf("the item is %d \n", root->info);
preorder(root->llink);
preorder(root->rlink);
}}
```

```
void inorder(NODE root){
if(root!=NULL){
inorder(root->llink);
printf("The item is %d \n", root->info);
inorder(root->rlink);
}}

void preorder(NODE root){
if(root!=NULL){
preorder(root->llink);
preorder(root->rlink);
printf("The item is %d \n", root->info);
}}

void display(NODE root, int i){
int j;
if(root!=NULL){
display(root->rlink, i+1);
for(j=1; j<=i; j++)
printf("  ");
printf("%d \n", root->info);
display(root->llink, i+1);
}}

int main(){
NODE root = NULL;
int choice, i, item;
for(;;){
printf("1. Insert \n 2.Preorder \n 3. Inorder \n 4. Postorder \n 5.
    Display \n");
printf("Enter the choice \n");
scanf("%d \n", &choice);
switch(choice){
case 1: printf("Enter the item: \n");
        scanf("%d", &item);
        root=insert(item, root);
        break;
```

```
case 2: if(root == NULL){
          printf("Tree is empty \n");
        }
        else{
        printf("Given tree is... ");
        display(root, 1);
        printf("The preorder traversal is: \n");
        preorder(root);
        }
        break;
case 3: if(root == NULL){
          printf("Tree is empty \n");
        }
        else{
        printf("Given tree is.. ");
        display(root, 1);
        printf("The inorder transversal is \n");
        inorder(root);
        }
        break;
case 4: if(root == NULL){
          printf("Tree is empty \n");
        }
        else{
        printf("Given tree is.. \n");
        display(root, 1);
        printf("The postorder transversal is \n");
        postorder(root);
        }
        break;
case 5: display(root, 1);
        break;
default: printf("Invalid choice entered \n");
        exit(0);
```

```
    }
    return 0;
}
```


Binary_search_tree - [tree_binaryserach.dev] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
binary_search_tree_lp10.cpp
#include<stdio.h>
#include<conio.h>
#include<malloc.h>
#include<process.h>
struct node
{
int info;
struct node *rlink;
struct node *llink;
};
typedef struct node *NODE;
NODE getnode()
{
NODE x;
x=(NODE)malloc(sizeof(struct node));
if(x==NULL)
{
printf("mem full\n");
exit(0);
}
return x;
}
void freenode(NODE x)
{
free(x);
}
NODE insert(NODE root,int item)
{
NODE temp,cur,prev;
temp=getnode();
temp->rlink=NULL;
temp->llink=NULL;
temp->info=item;
if(root==NULL)
return temp;
prev=NULL;
cur=root;
while(cur!=NULL)
Compiler Resources Compile Log Debug Find Results
Line: 158 Col: 26 Sel: 0 Lines: 167 Length: 2640 Insert Done parsing in 0.032 seconds
Type here to search
ENG
14:35
21-12-2020


Binary_search_tree - [tree_binaryserach.dev] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
binary_search_tree_lp10.cpp
cur=root;
while(cur!=NULL)
{
prev=cur;
cur=(item<cur->info)?cur->llink:cur->rlink;
}
if(item<prev->info)
prev->llink=temp;
else
prev->rlink=temp;
return root;
}
void display(NODE root,int i)
{
int j;
if(root!=NULL)
{
display(root->rlink,i+1);
for(j=0;j<i;j++)
printf(" ");
printf("%d\n",root->info);
display(root->llink,i+1);
}
}
NODE delete_that(NODE root,int item)
{
NODE cur,parent,q,suc;
if(root==NULL)
{
printf("empty\n");
return root;
}
parent=NULL;
cur=root;
while(cur!=NULL&&item!=cur->info)
{
parent=cur;
cur=(item<cur->info)?cur->llink:cur->rlink;
Compiler Resources Compile Log Debug Find Results
Line: 158 Col: 26 Sel: 0 Lines: 167 Length: 2640 Insert Done parsing in 0.032 seconds
Type here to search
ENG
14:35
21-12-2020

Binary_search_tree - [tree_binaryserach.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

(globals)

binary_search_tree_lp10.cpp

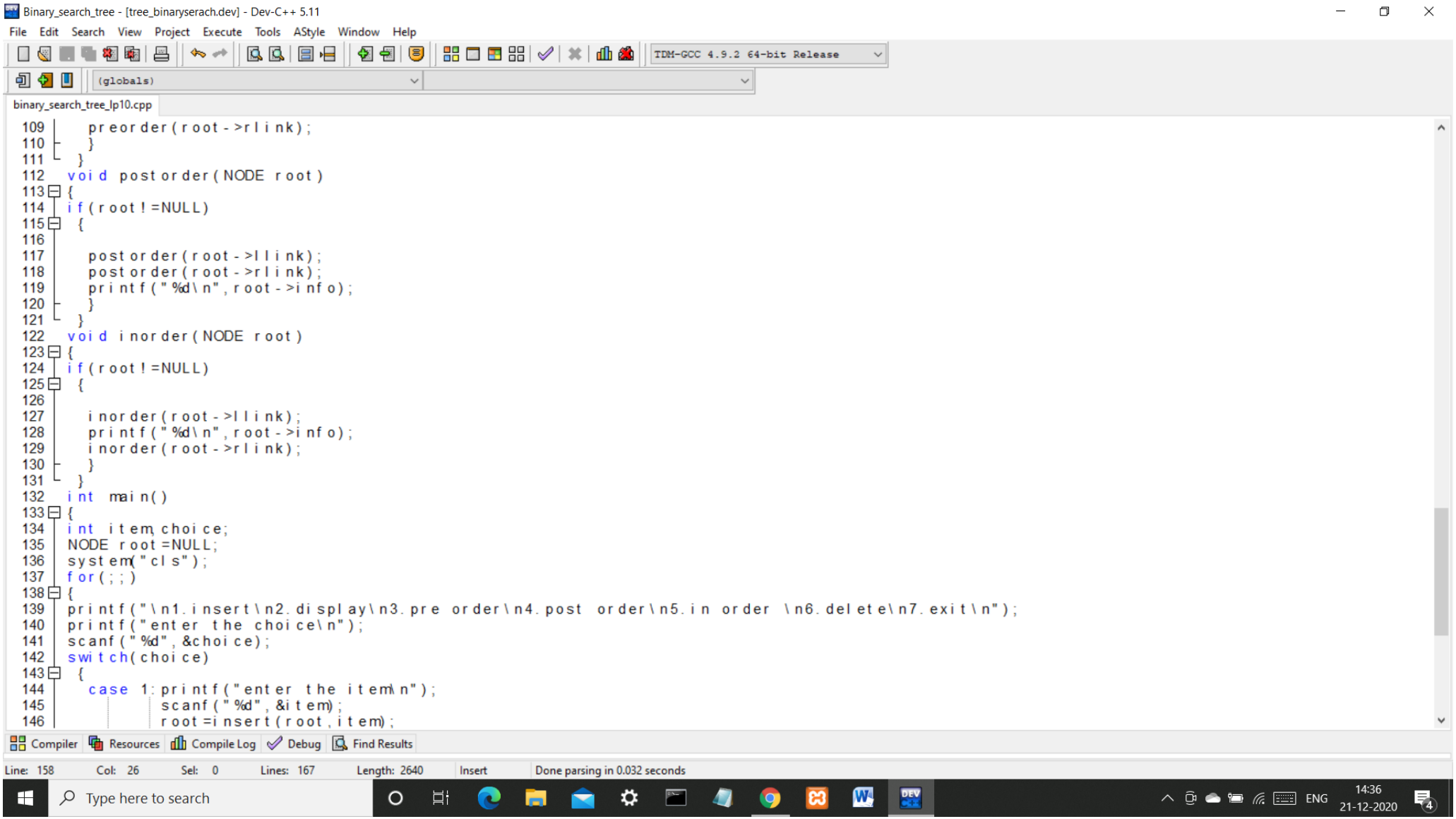
```
73  parent=cur;
74  cur=(item<cur->info)?cur->llink:cur->rlink;
75  }
76  if(cur==NULL)
77  {
78   printf("not found\n");
79   return root;
80  }
81  if(cur->llink==NULL)
82   q=cur->rlink;
83  else if(cur->rlink==NULL)
84   q=cur->llink;
85  else
86   {
87   suc=cur->rlink;
88   while(suc->llink!=NULL)
89    suc=suc->llink;
90   suc->llink=cur->llink;
91   q=cur->rlink;
92   }
93   if(parent==NULL)
94    return q;
95   if(cur==parent->llink)
96    parent->llink=q;
97   else
98    parent->rlink=q;
99   freenode(cur);
100  return root;
101  }
102
103 void preorder(NODE root)
104 {
105 if(root!=NULL)
106  {
107   printf("%d\n",root->info);
108   preorder(root->llink);
109   preorder(root->rlink);
110   }
```

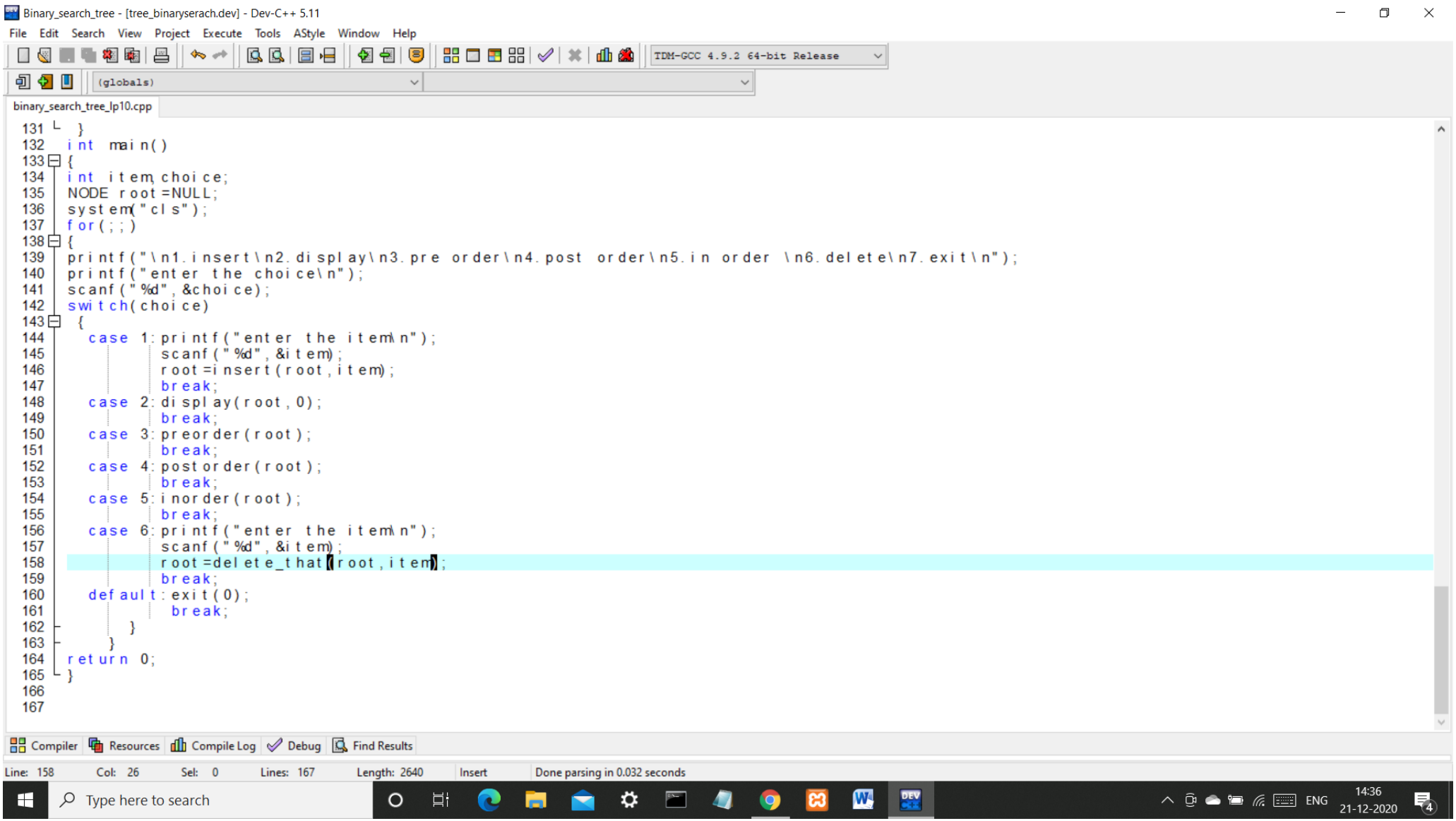
Compiler Resources Compile Log Debug Find Results

Line: 158 Col: 26 Sel: 0 Lines: 167 Length: 2640 Insert Done parsing in 0.032 seconds

Type here to search

ENG 14:36 21-12-2020


Binary_search_tree - [tree_binaryserach.dev] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
binary_search_tree_lp10.cpp
preorder(root->rlink);
}
}
void postorder(NODE root)
{
if(root!=NULL)
{
postorder(root->llink);
postorder(root->rlink);
printf("%d\n",root->info);
}
}
void inorder(NODE root)
{
if(root!=NULL)
{
inorder(root->llink);
printf("%d\n",root->info);
inorder(root->rlink);
}
}
int main()
{
int item,choice;
NODE root=NULL;
system("cls");
for(;;)
{
printf("\n1.insert\n2.display\n3.pre order\n4.post order\n5.in order \n6.delete\n7.exit\n");
printf("enter the choice\n");
scanf("%d",&choice);
switch(choice)
{
case 1:printf("enter the item\n");
scanf("%d",&item);
root=insert(root,item);
Compiler Resources Compile Log Debug Find Results
Line: 158 Col: 26 Sel: 0 Lines: 167 Length: 2640 Insert Done parsing in 0.032 seconds
Type here to search
ENG
14:36
21-12-2020


Binary_search_tree - [tree_binaryserach.dev] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
TDM-GCC 4.9.2 64-bit Release
(globals)
binary_search_tree_lp10.cpp
}
int main()
{
int item,choice;
NODE root=NULL;
system("cls");
for(;;)
{
printf("\n1.insert\n2.display\n3.pre order\n4.post order\n5.in order \n6.delete\n7.exit\n");
printf("enter the choice\n");
scanf("%d",&choice);
switch(choice)
{
case 1:printf("enter the item\n");
scanf("%d",&item);
root=insert(root,item);
break;
case 2:display(root,0);
break;
case 3:preorder(root);
break;
case 4:postorder(root);
break;
case 5:inorder(root);
break;
case 6:printf("enter the item\n");
scanf("%d",&item);
root=delete_that(root,item);
break;
default:exit(0);
break;
}
}
return 0;
}
Compiler
Resources
Compile Log
Debug
Find Results
Line: 158 Col: 26 Sel: 0 Lines: 167 Length: 2640 Insert Done parsing in 0.032 seconds
Type here to search
ENG
14:36
21-12-2020

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\tree binary search\tree_binaryserach.exe

```
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
Enter the choice:
1
Enter the item:
1
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
Enter the choice:
1
Enter the item:
2
Give direction to insert..
l
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
Enter the choice:
1
Enter the item:
3
Give direction to insert..
r
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
Enter the choice:
1
Enter the item:
4
Give direction to insert..
ll
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
Enter the choice:
1
Enter the item:
```

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\tree binary search\tree_binaryserach.exe

```
3.Inorder
4.Postorder
5.Display
Enter the choice:
1
Enter the item:
5
Give direction to insert..
lr
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
Enter the choice:
1
Enter the item:
6
Give direction to insert..
rl
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
Enter the choice:
1
Enter the item:
7
Give direction to insert..
rr
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
Enter the choice:
5
      7
    3
      6
  1
      5
    2
      4
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
```

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\tree binary search\tree_binaryserach.exe

```
2.Preorder
3.Inorder
4.Postorder
5.Display
Enter the choice:
2
Given tree is..      7
    3
      6
  1
      5
    2
      4
The preorder traversal is:
the item is 1
the item is 2
the item is 4
the item is 5
the item is 3
the item is 6
the item is 7
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
Enter the choice:
3
Given tree is..      7
    3
      6
  1
      5
    2
      4
The inorder traversal is
The item is4
The item is2
The item is5
The item is1
The item is6
The item is3
The item is7
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
Enter the choice:
4
```

C:\Users\sohan\Desktop\C Programs\Data Structures Lab\tree binary search\tree_binaryserach.exe

```
3.Inorder
4.Postorder
5.Display
Enter the choice:
4
Given tree is..      7
    3
      6
  1
      5
    2
      4
The postorder traversal is
The item is4
The item is5
The item is2
The item is6
The item is7
The item is3
The item is1
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
Enter the choice:
5
      7
    3
      6
  1
      5
    2
      4
1.Insert
2.Preorder
3.Inorder
4.Postorder
5.Display
Enter the choice:
6
Invalid choice entered.

--------------------------------
Process exited after 226.4 seconds with return value 0
Press any key to continue . . .
```