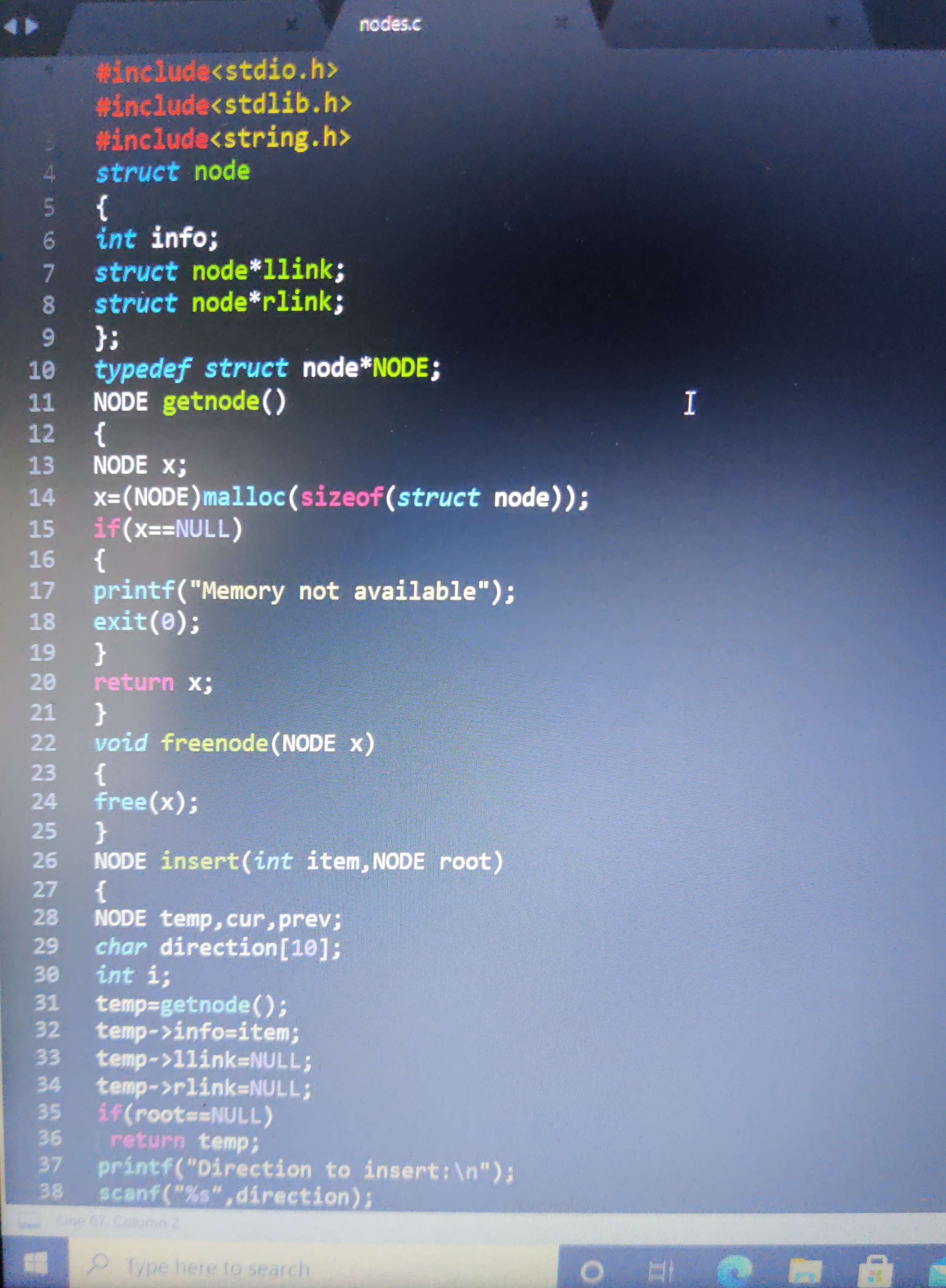
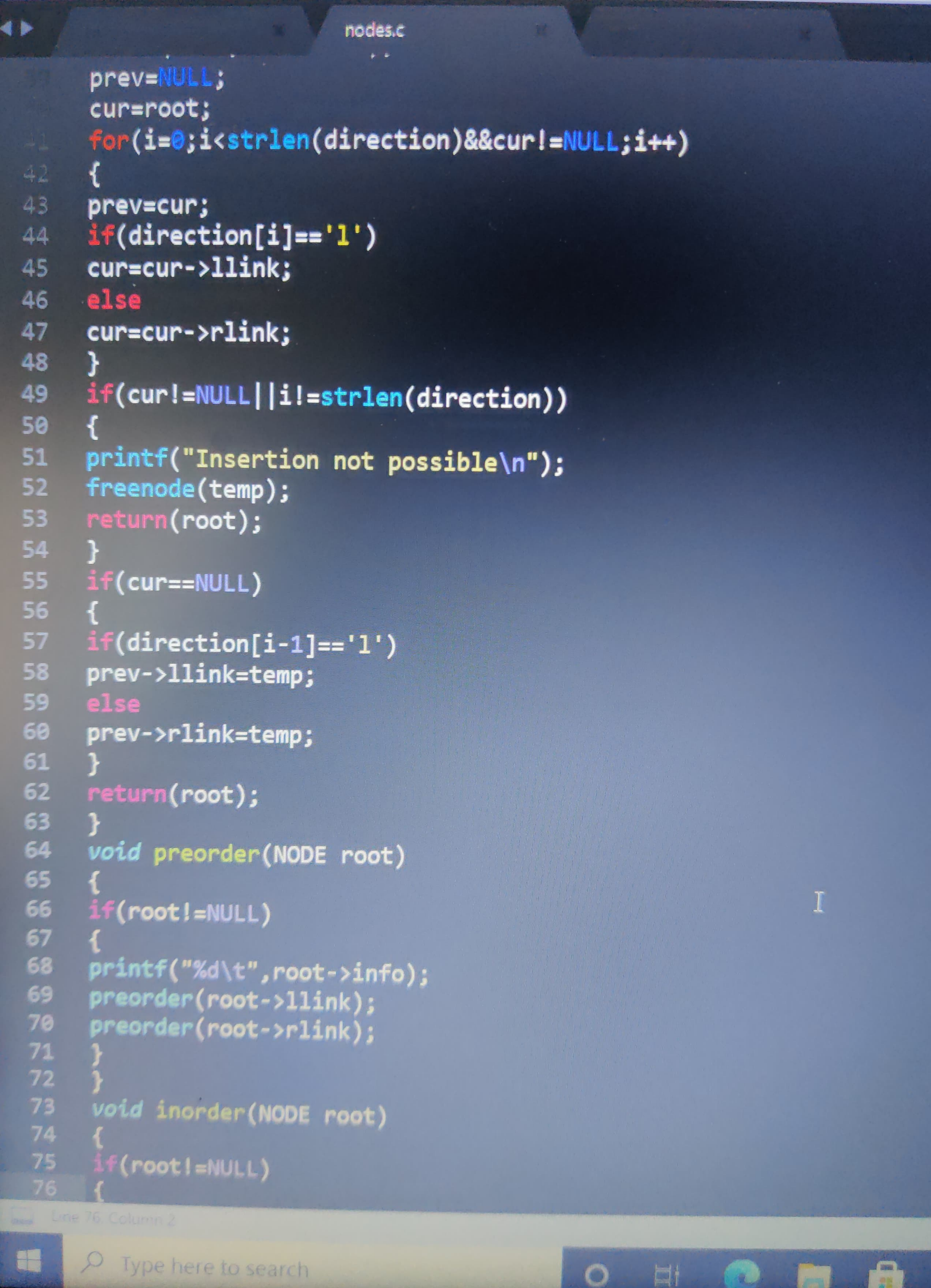
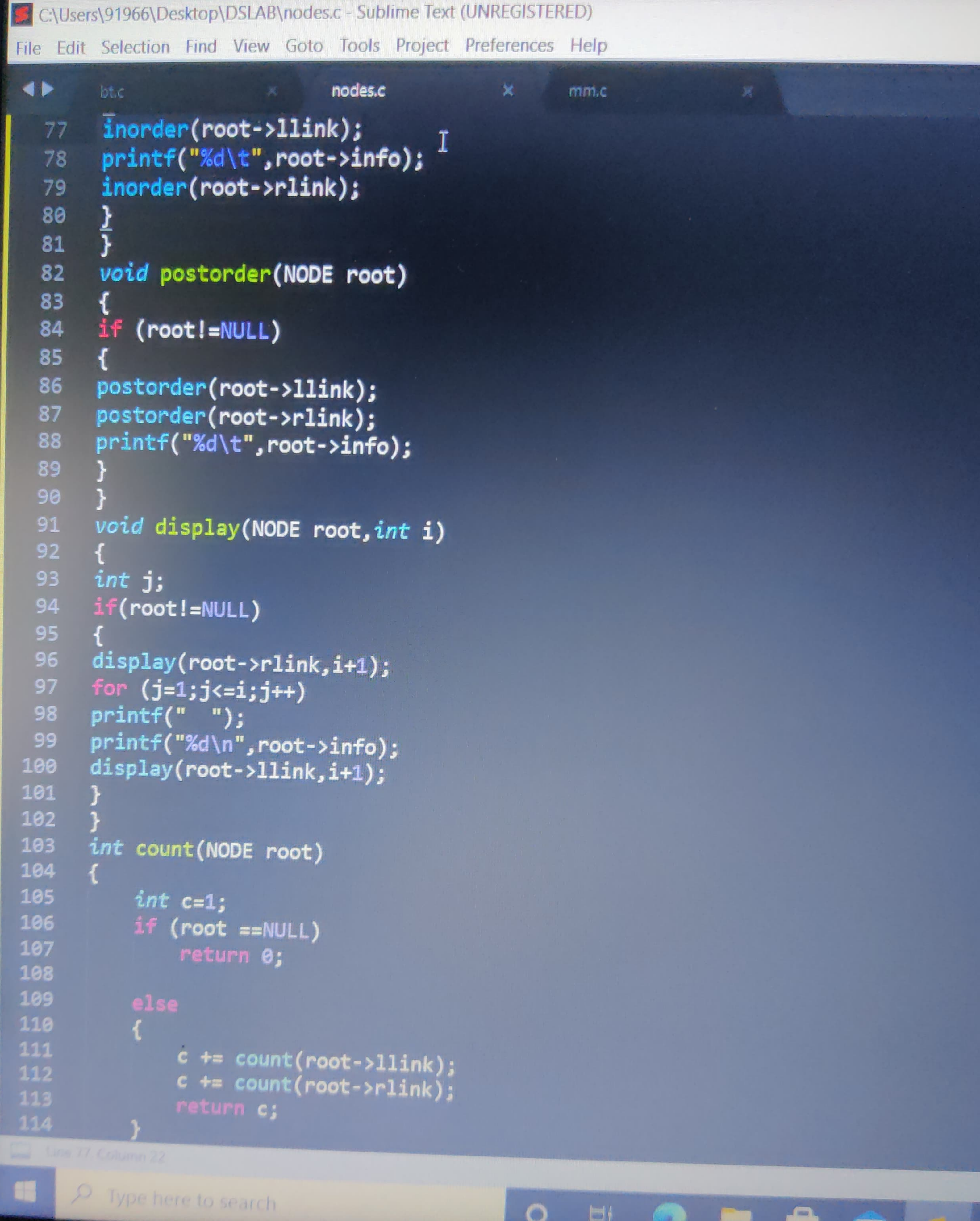


```c
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
struct node
{
int info;
struct node*llink;
struct node*rlink;
};
typedef struct node*NODE;
NODE getnode()
{
NODE x;
x=(NODE)malloc(sizeof(struct node));
if(x==NULL)
{
printf("Memory not available");
exit(0);
}
return x;
}
void freenode(NODE 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("Direction to insert:\n");
scanf("%s",direction);
```

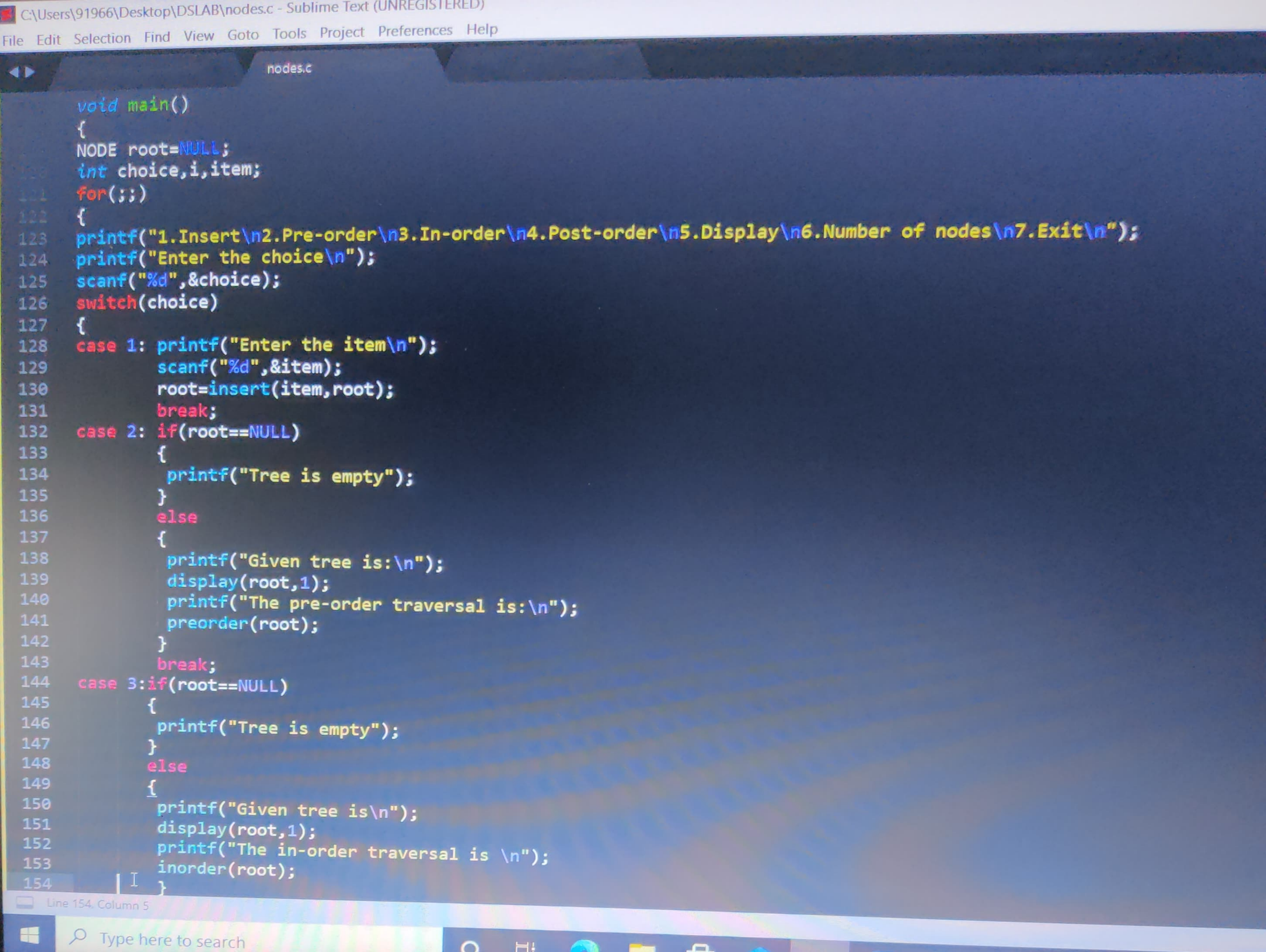


```c
prev=NULL;
cur=root;
for(i=0;i<strlen(direction)&&cur!=NULL;i++)
{
prev=cur;
if(direction[i]=='l')
cur=cur->llink;
else
cur=cur->rlink;
}
if(cur!=NULL||i!=strlen(direction))
{
printf("Insertion not possible\n");
freenode(temp);
return(root);
}
if(cur==NULL)
{
if(direction[i-1]=='l')
prev->llink=temp;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

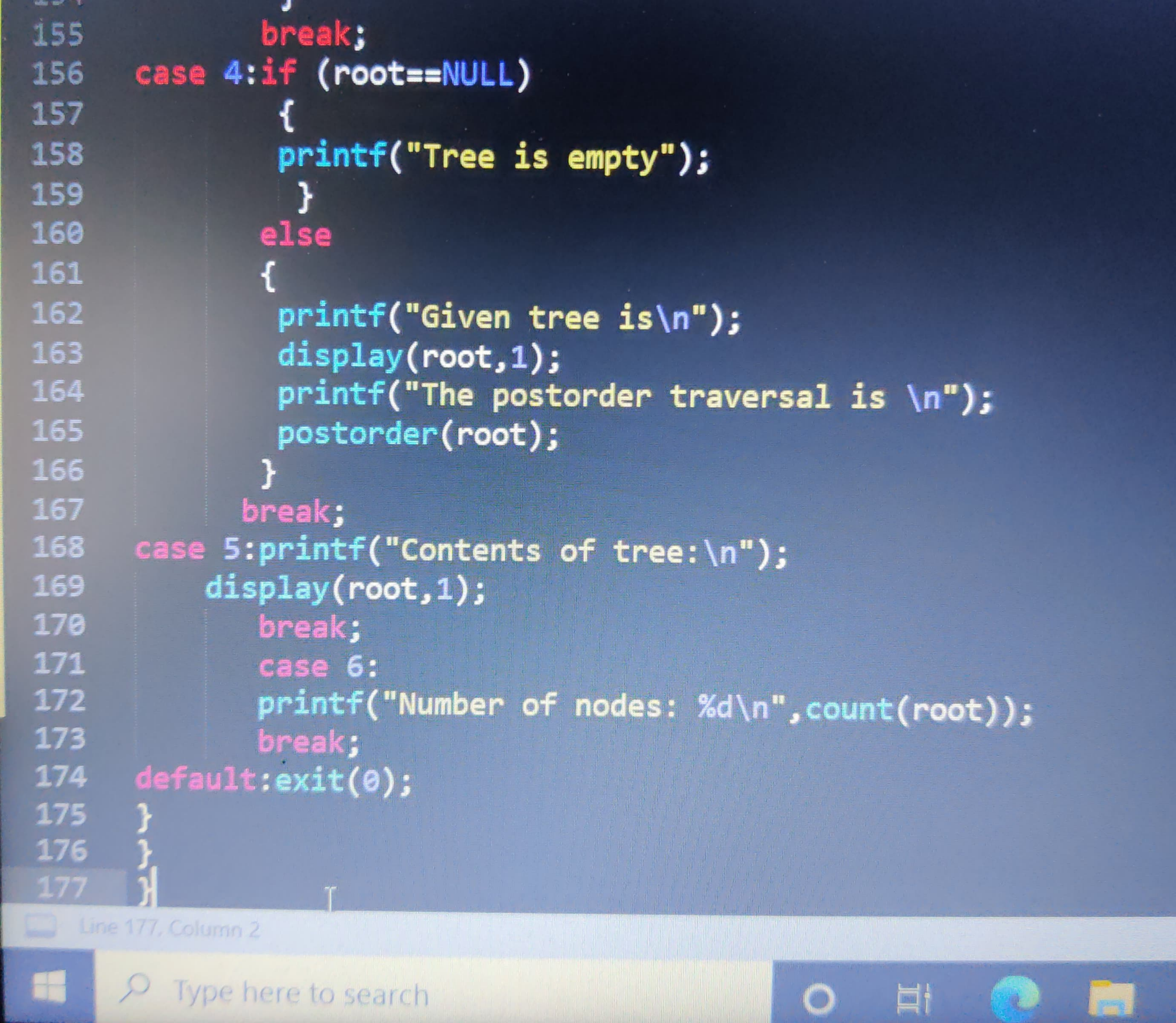

C:\Users\91966\Desktop\DSLAB\nodes.c - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
bt.c
nodes.c
mm.c
Line 77, Column 22
Type here to search

```
inorder(root->llink);
printf("%d\t",root->info);
inorder(root->rlink);
}
}
void postorder(NODE root)
{
if (root!=NULL)
{
postorder(root->llink);
postorder(root->rlink);
printf("%d\t",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 count(NODE root)
{
    int c=1;
    if (root ==NULL)
        return 0;

    else
    {
        c += count(root->llink);
        c += count(root->rlink);
        return c;
    }
```


C:\Users\91966\Desktop\DSLAB\nodes.c - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
nodes.c
Line 154, Column 5
Type here to search

```c
void main()
{
NODE root=NULL;
int choice,i,item;
for(;;)
{
printf("1.Insert\n2.Pre-order\n3.In-order\n4.Post-order\n5.Display\n6.Number of nodes\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(item,root);
        break;
case 2: if(root==NULL)
        {
         printf("Tree is empty");
        }
        else
        {
         printf("Given tree is:\n");
         display(root,1);
         printf("The pre-order traversal is:\n");
         preorder(root);
        }
        break;
case 3:if(root==NULL)
       {
        printf("Tree is empty");
       }
       else
       {
        printf("Given tree is\n");
        display(root,1);
        printf("The in-order traversal is \n");
        inorder(root);
        }
```



```c
            break;
    case 4:if (root==NULL)
             {
             printf("Tree is empty");
              }
            else
            {
             printf("Given tree is\n");
             display(root,1);
             printf("The postorder traversal is \n");
             postorder(root);
            }
           break;
    case 5:printf("Contents of tree:\n");
        display(root,1);
            break;
            case 6:
            printf("Number of nodes: %d\n",count(root));
            break;
    default:exit(0);
    }
    }
    }
```