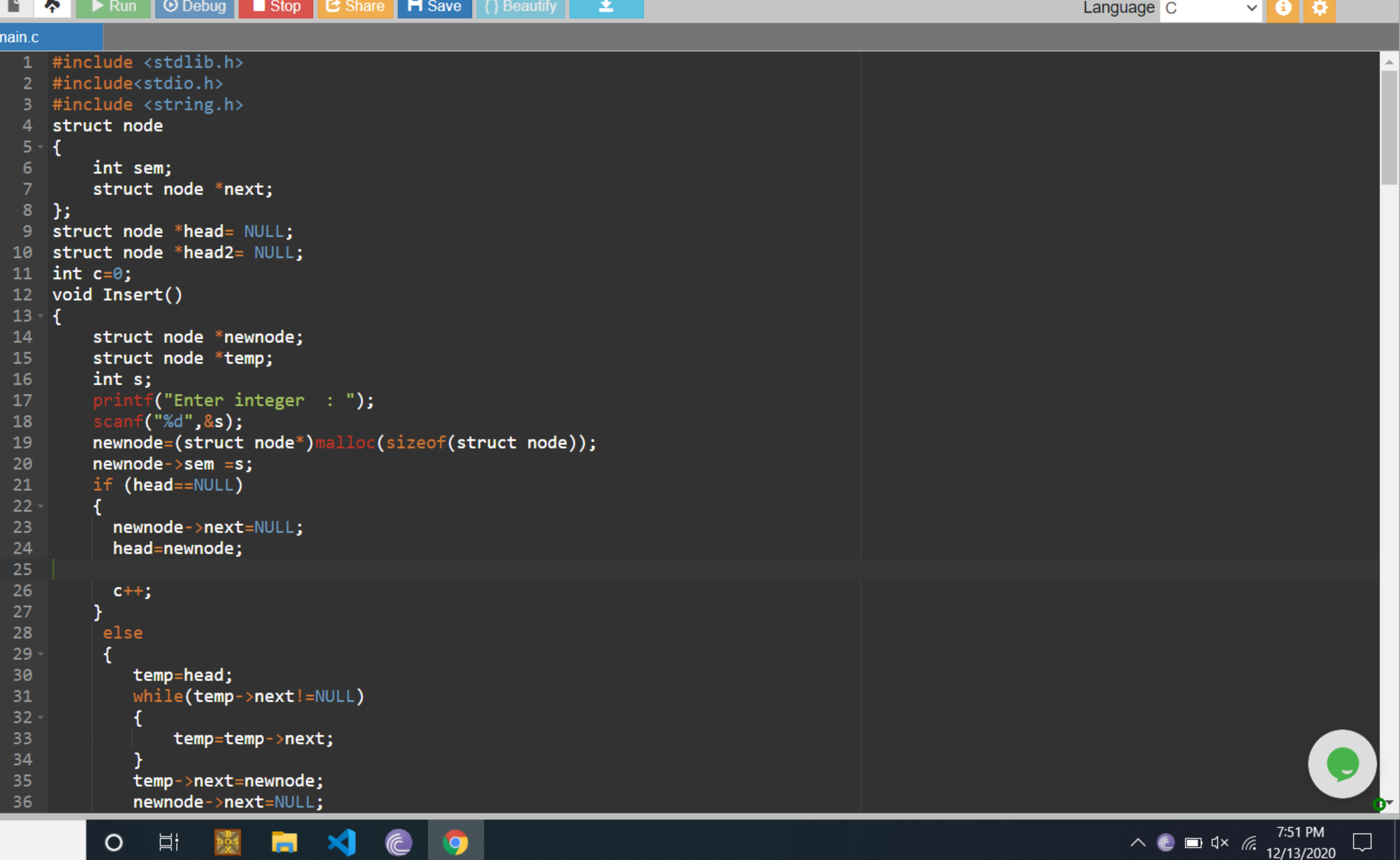
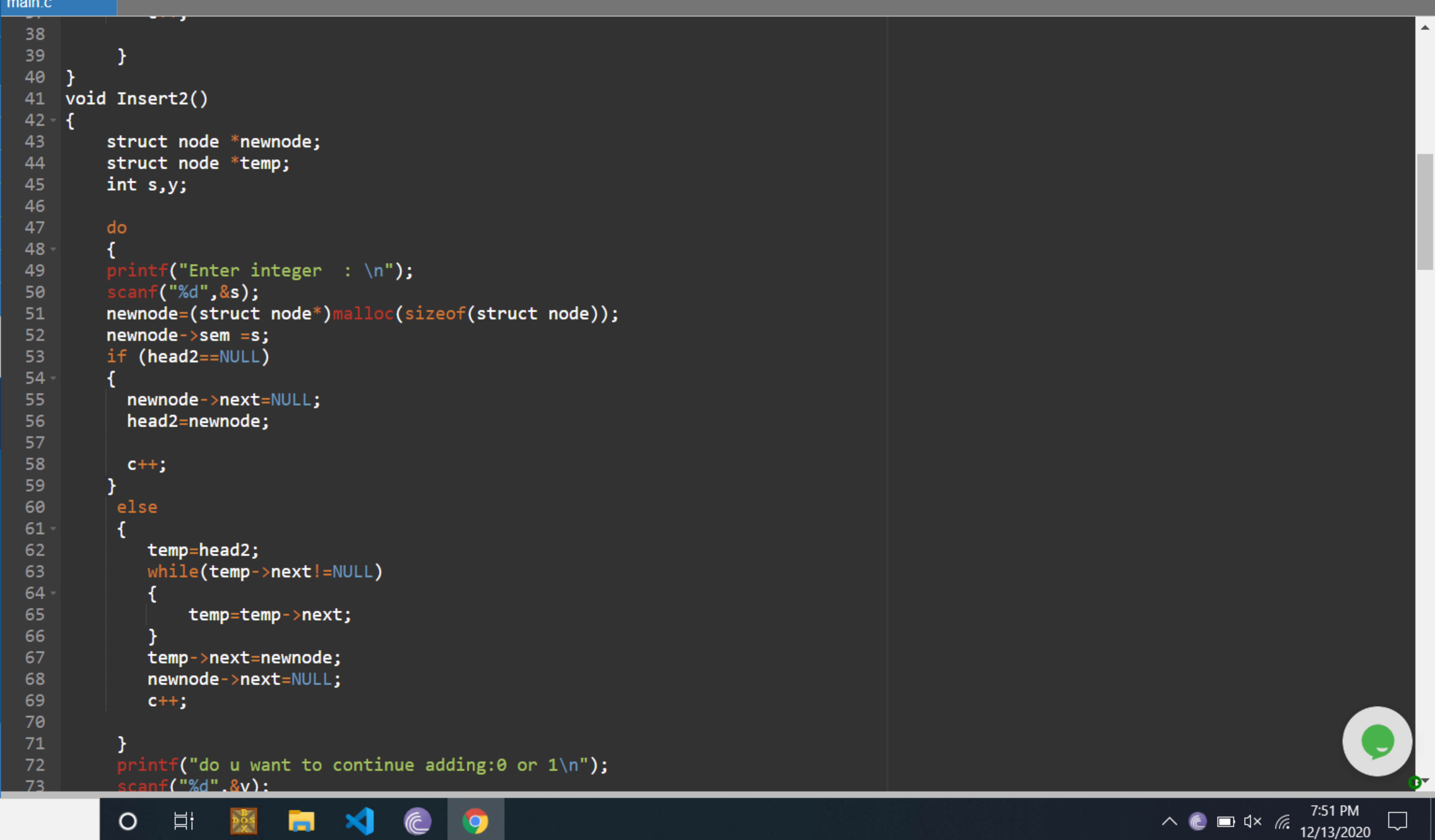
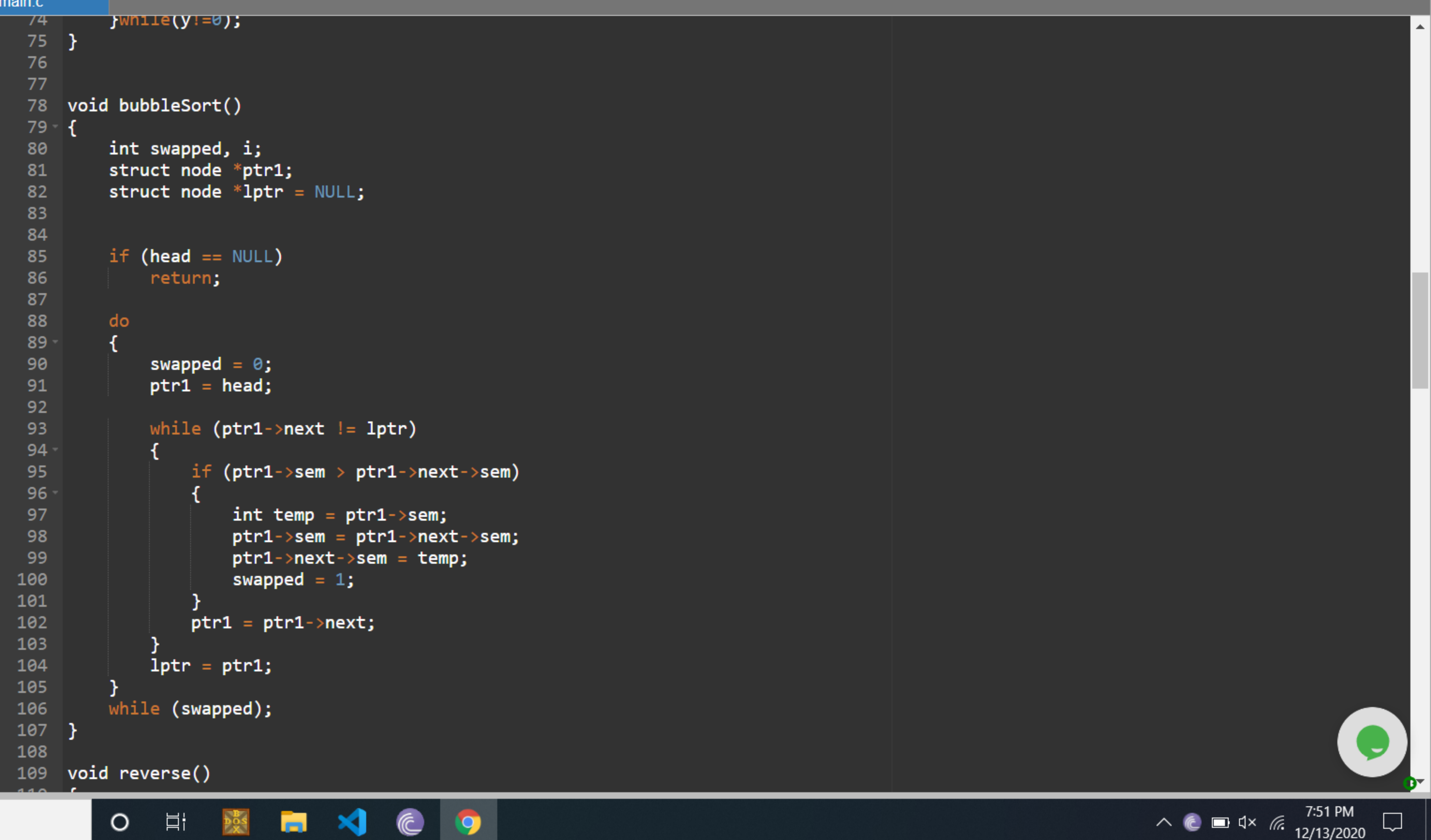


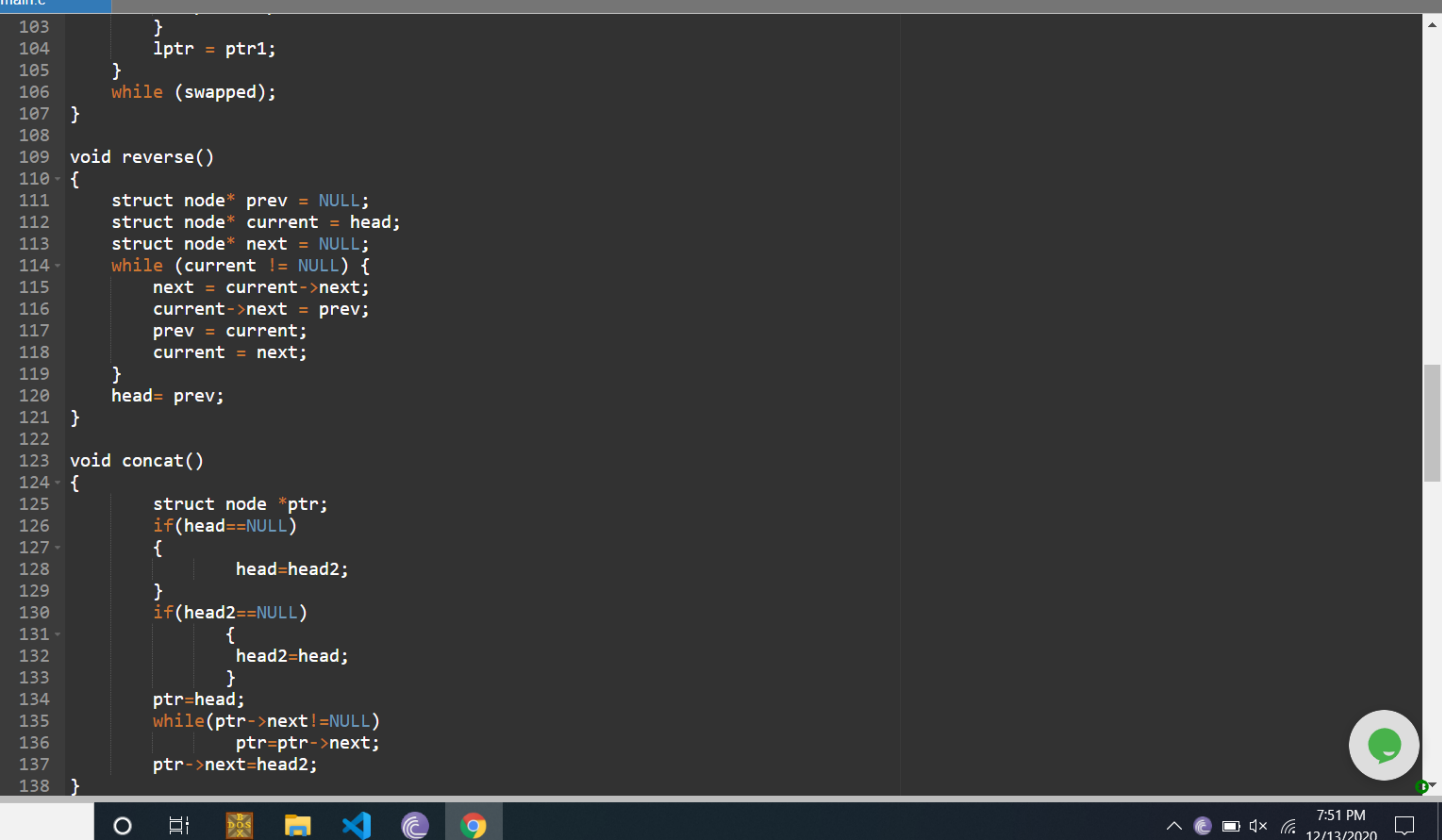
```c
#include <stdlib.h>
#include<stdio.h>
#include <string.h>
struct node
{
    int sem;
    struct node *next;
};
struct node *head= NULL;
struct node *head2= NULL;
int c=0;
void Insert()
{
    struct node *newnode;
    struct node *temp;
    int s;
    printf("Enter integer  : ");
    scanf("%d",&s);
    newnode=(struct node*)malloc(sizeof(struct node));
    newnode->sem =s;
    if (head==NULL)
    {
      newnode->next=NULL;
      head=newnode;

      c++;
    }
     else
     {
        temp=head;
        while(temp->next!=NULL)
        {
            temp=temp->next;
        }
        temp->next=newnode;
        newnode->next=NULL;
```



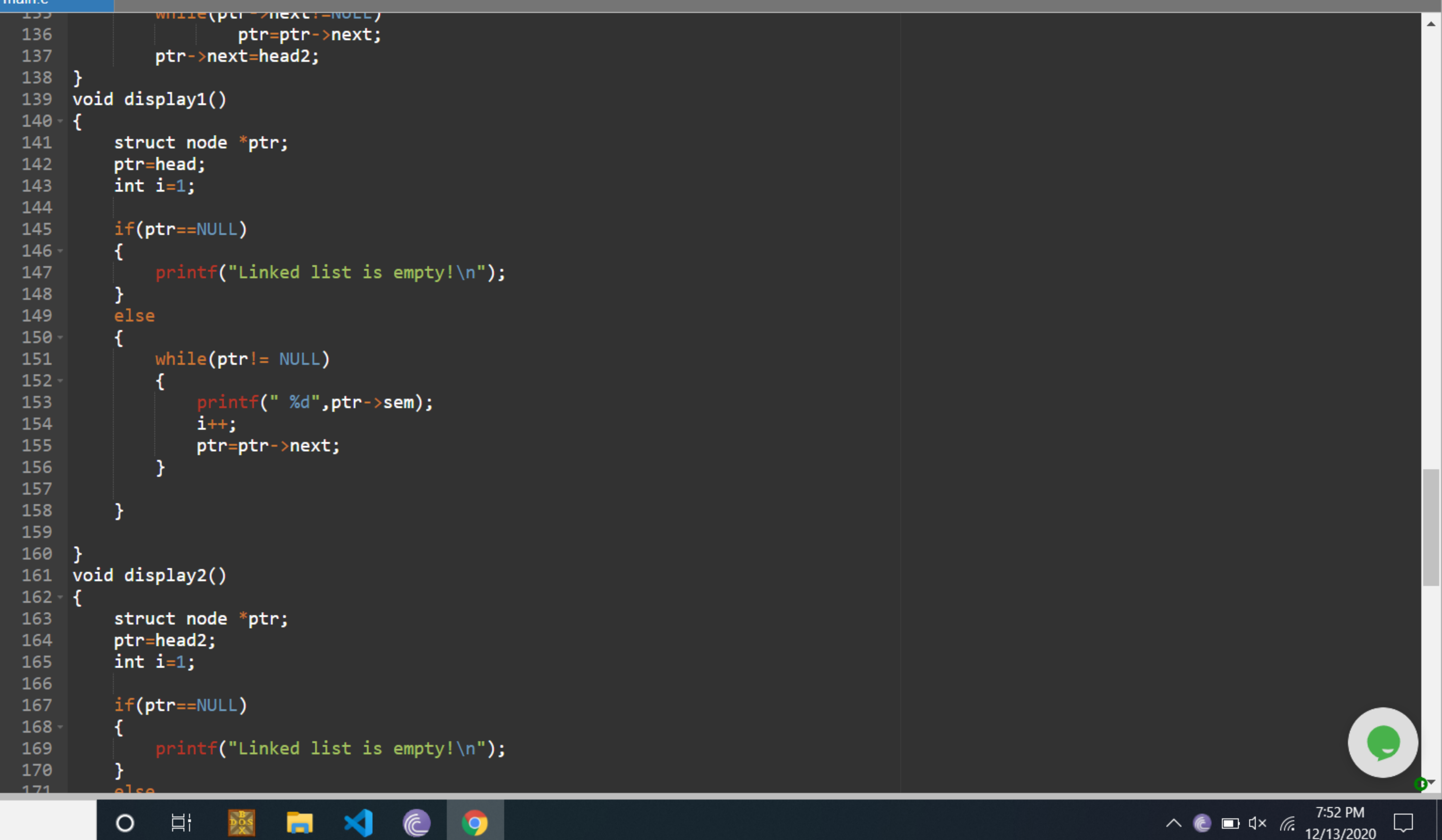
```
        c++;

     }
}
void Insert2()
{
    struct node *newnode;
    struct node *temp;
    int s,y;

    do
    {
    printf("Enter integer  : \n");
    scanf("%d",&s);
    newnode=(struct node*)malloc(sizeof(struct node));
    newnode->sem =s;
    if (head2==NULL)
    {
      newnode->next=NULL;
      head2=newnode;

      c++;
    }
     else
     {
        temp=head2;
        while(temp->next!=NULL)
        {
            temp=temp->next;
        }
        temp->next=newnode;
        newnode->next=NULL;
        c++;

     }
     printf("do u want to continue adding:0 or 1\n");
     scanf("%d",&y);
```



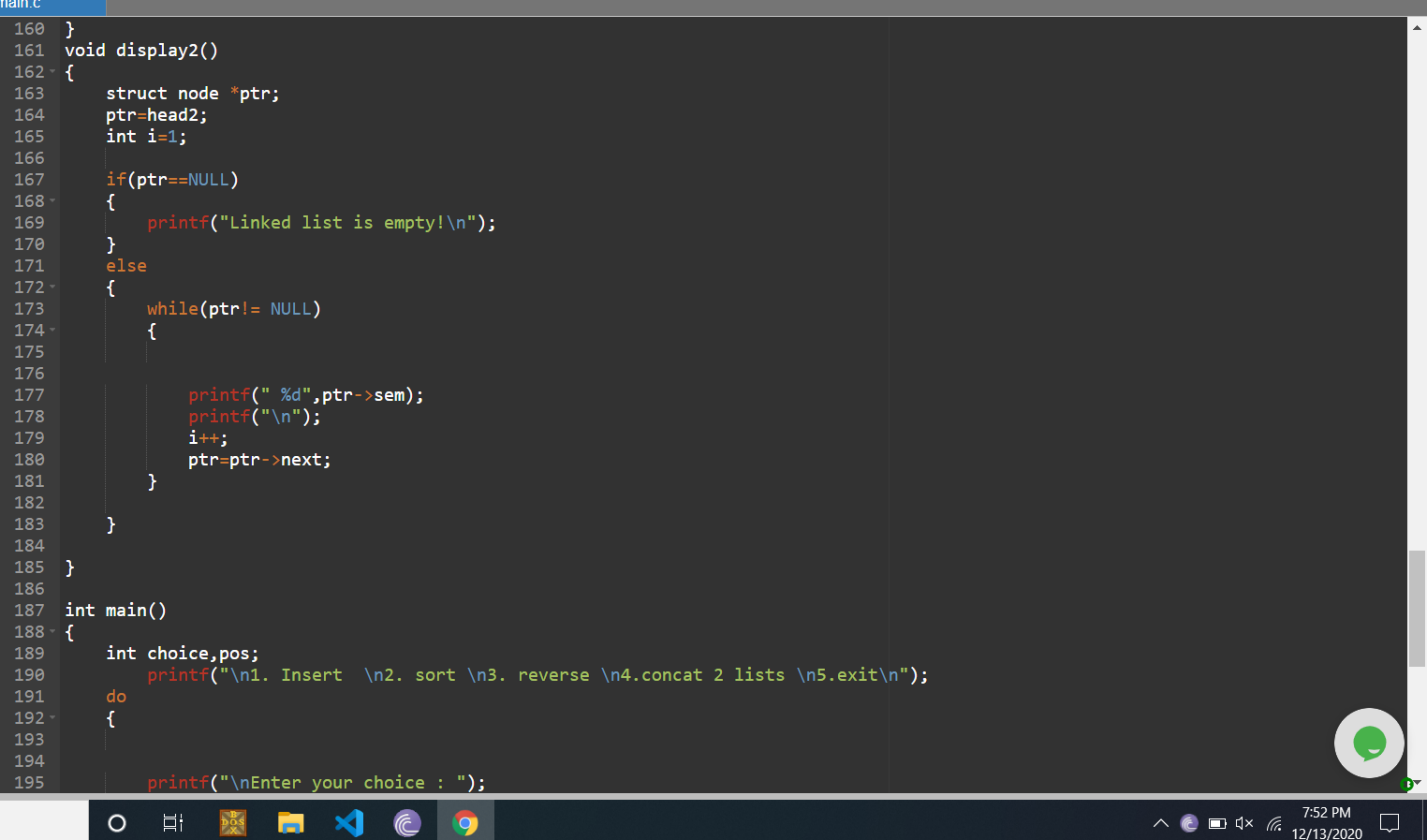
```
    }while(y!=0);
}


void bubbleSort()
{
    int swapped, i;
    struct node *ptr1;
    struct node *lptr = NULL;


    if (head == NULL)
        return;

    do
    {
        swapped = 0;
        ptr1 = head;

        while (ptr1->next != lptr)
        {
            if (ptr1->sem > ptr1->next->sem)
            {
                int temp = ptr1->sem;
                ptr1->sem = ptr1->next->sem;
                ptr1->next->sem = temp;
                swapped = 1;
            }
            ptr1 = ptr1->next;
        }
        lptr = ptr1;
    }
    while (swapped);
}

void reverse()
{
```



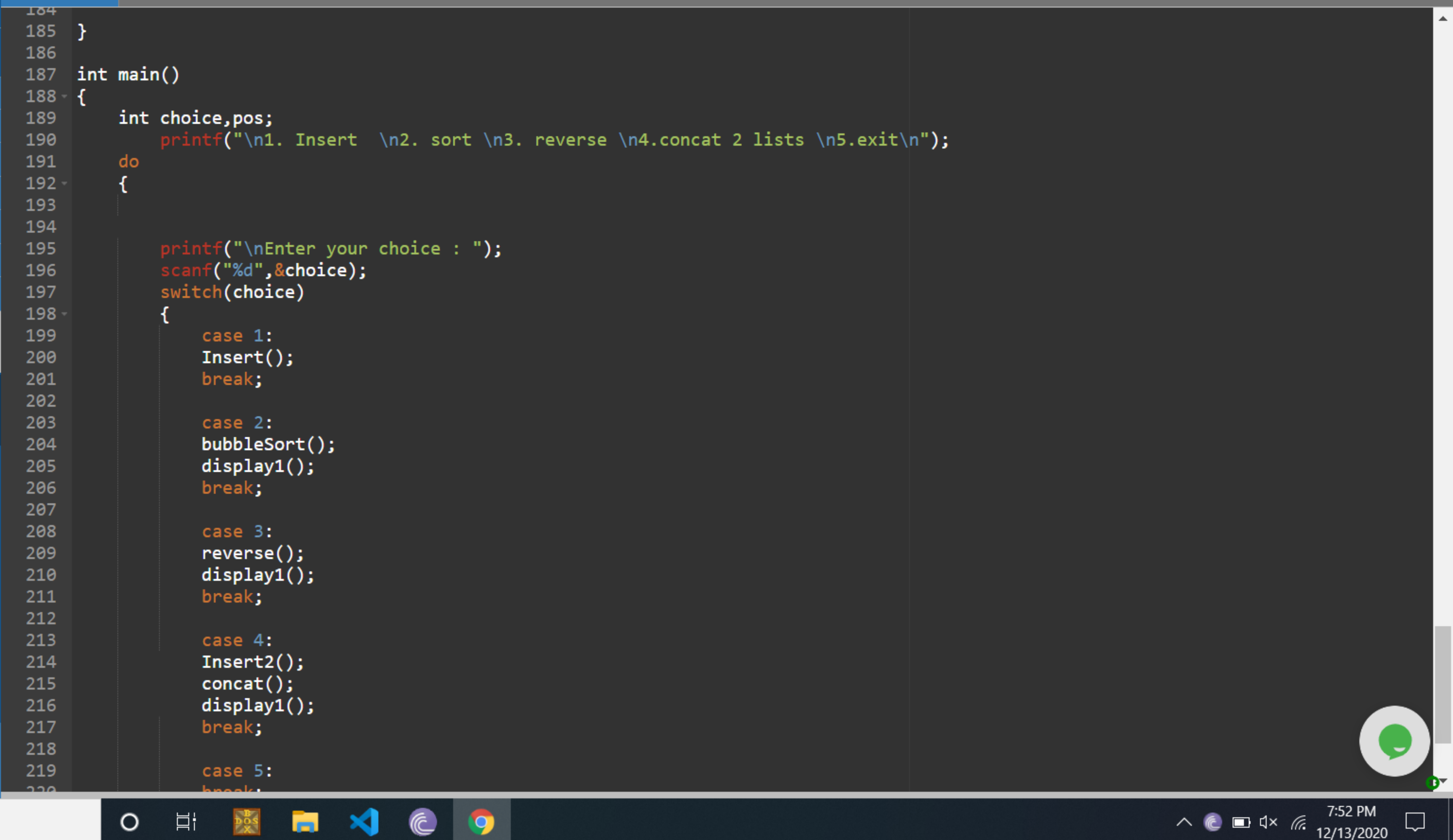
```c
        }
        lptr = ptr1;
    }
    while (swapped);
}

void reverse()
{
    struct node* prev = NULL;
    struct node* current = head;
    struct node* next = NULL;
    while (current != NULL) {
        next = current->next;
        current->next = prev;
        prev = current;
        current = next;
    }
    head= prev;
}

void concat()
{
        struct node *ptr;
        if(head==NULL)
        {
                head=head2;
        }
        if(head2==NULL)
               {
                head2=head;
               }
        ptr=head;
        while(ptr->next!=NULL)
                ptr=ptr->next;
        ptr->next=head2;
}
```



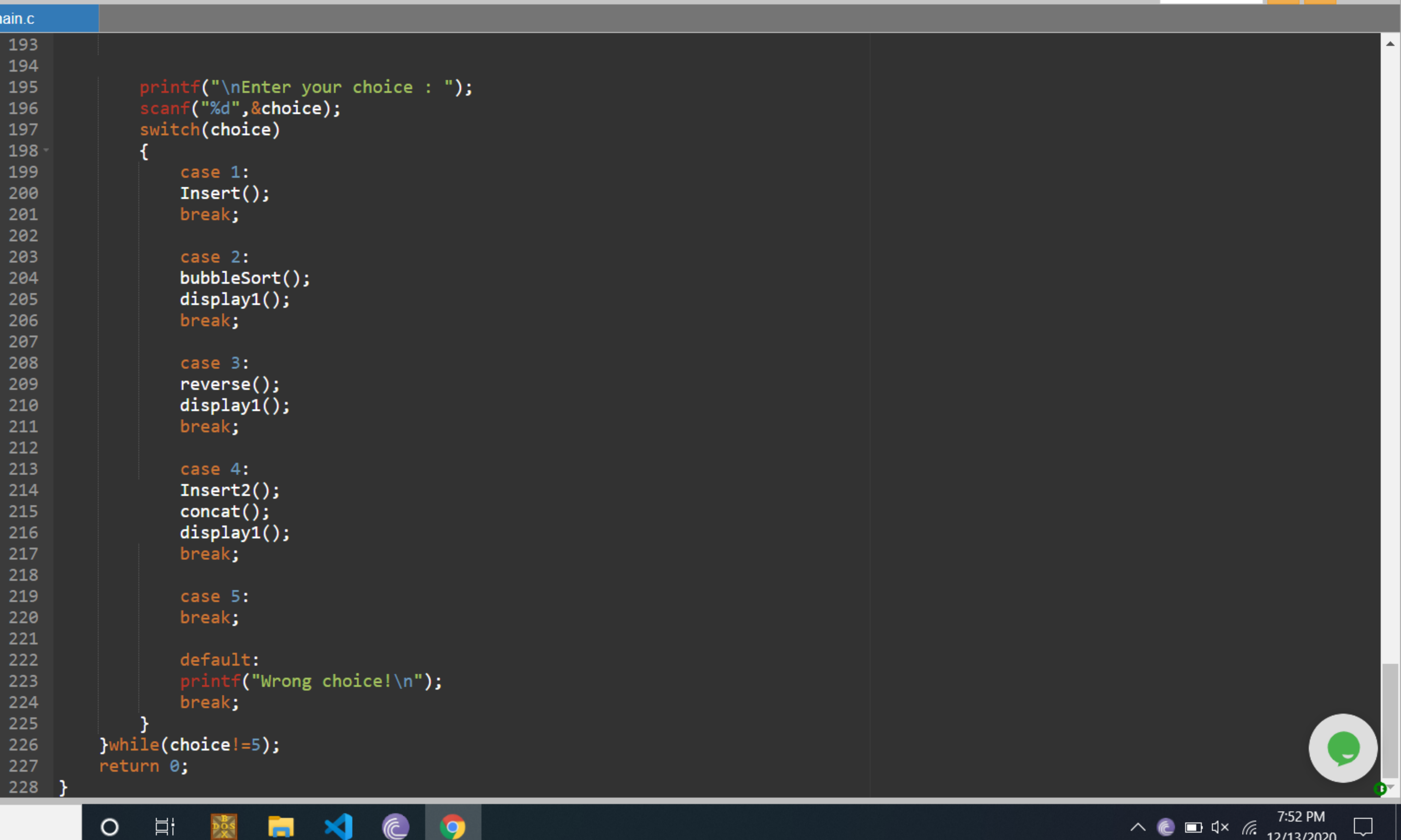
```
        while(ptr->next!=NULL)
                ptr=ptr->next;
        ptr->next=head2;
}
void display1()
{
    struct node *ptr;
    ptr=head;
    int i=1;

    if(ptr==NULL)
    {
        printf("Linked list is empty!\n");
    }
    else
    {
        while(ptr!= NULL)
        {
            printf(" %d",ptr->sem);
            i++;
            ptr=ptr->next;
        }

    }

}
void display2()
{
    struct node *ptr;
    ptr=head2;
    int i=1;

    if(ptr==NULL)
    {
        printf("Linked list is empty!\n");
    }
    else
```



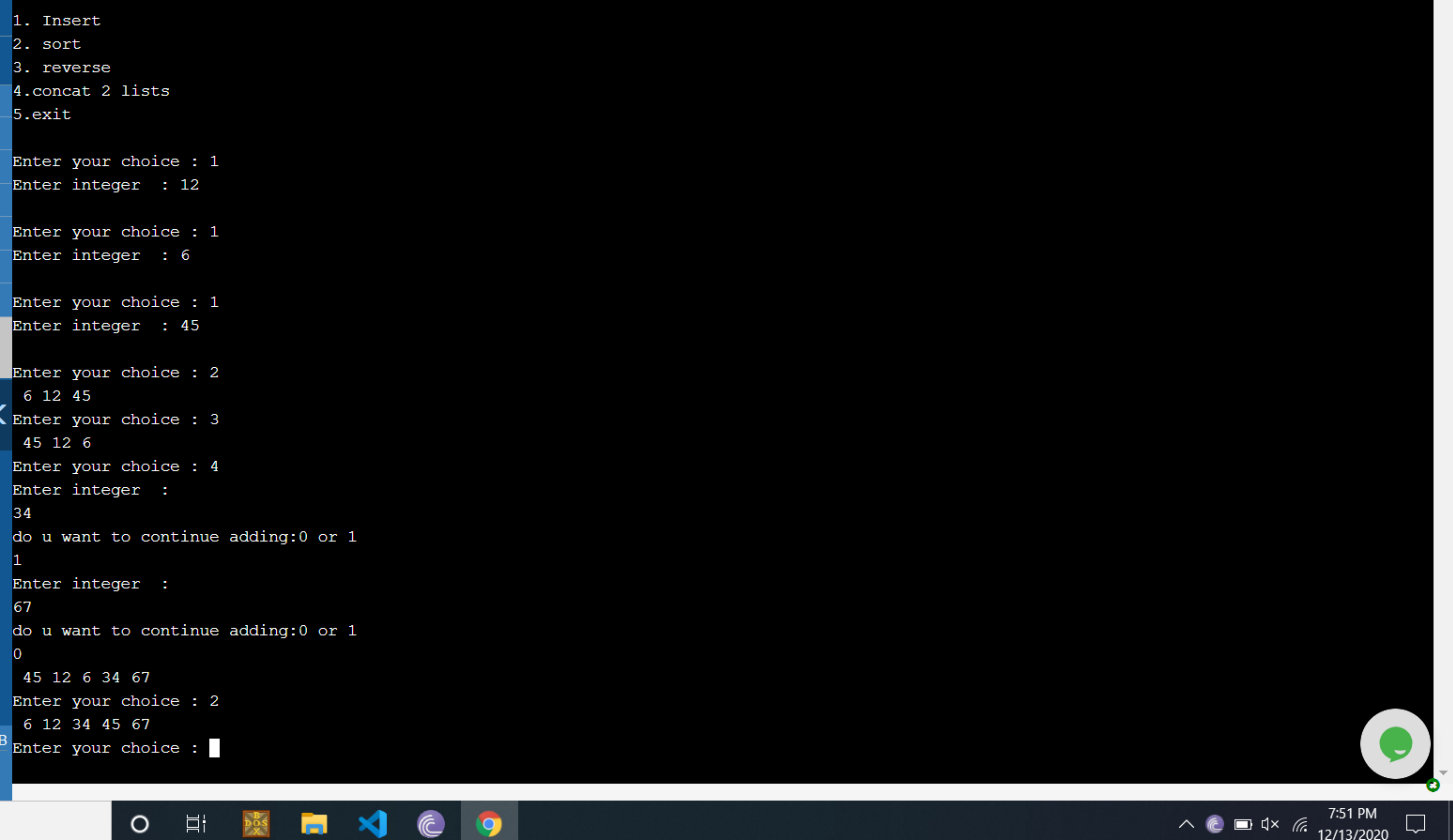
```c
}
void display2()
{
    struct node *ptr;
    ptr=head2;
    int i=1;

    if(ptr==NULL)
    {
        printf("Linked list is empty!\n");
    }
    else
    {
        while(ptr!= NULL)
        {


            printf(" %d",ptr->sem);
            printf("\n");
            i++;
            ptr=ptr->next;
        }

    }

}

int main()
{
    int choice,pos;
        printf("\n1. Insert  \n2. sort \n3. reverse \n4.concat 2 lists \n5.exit\n");
    do
    {


        printf("\nEnter your choice : ");
```



```c
}

int main()
{
    int choice,pos;
        printf("\n1. Insert  \n2. sort \n3. reverse \n4.concat 2 lists \n5.exit\n");
    do
    {


        printf("\nEnter your choice : ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:
            Insert();
            break;

            case 2:
            bubbleSort();
            display1();
            break;

            case 3:
            reverse();
            display1();
            break;

            case 4:
            Insert2();
            concat();
            display1();
            break;

            case 5:
            break;
```



```c


        printf("\nEnter your choice : ");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1:
            Insert();
            break;

            case 2:
            bubbleSort();
            display1();
            break;

            case 3:
            reverse();
            display1();
            break;

            case 4:
            Insert2();
            concat();
            display1();
            break;

            case 5:
            break;

            default:
            printf("Wrong choice!\n");
            break;
        }
    }while(choice!=5);
    return 0;
}
```



```
1. Insert
2. sort
3. reverse
4.concat 2 lists
5.exit

Enter your choice : 1
Enter integer  : 12

Enter your choice : 1
Enter integer  : 6

Enter your choice : 1
Enter integer  : 45

Enter your choice : 2
 6 12 45
Enter your choice : 3
 45 12 6
Enter your choice : 4
Enter integer  :
34
do u want to continue adding:0 or 1
1
Enter integer  :
67
do u want to continue adding:0 or 1
0
 45 12 6 34 67
Enter your choice : 2
 6 12 34 45 67
Enter your choice : █
```