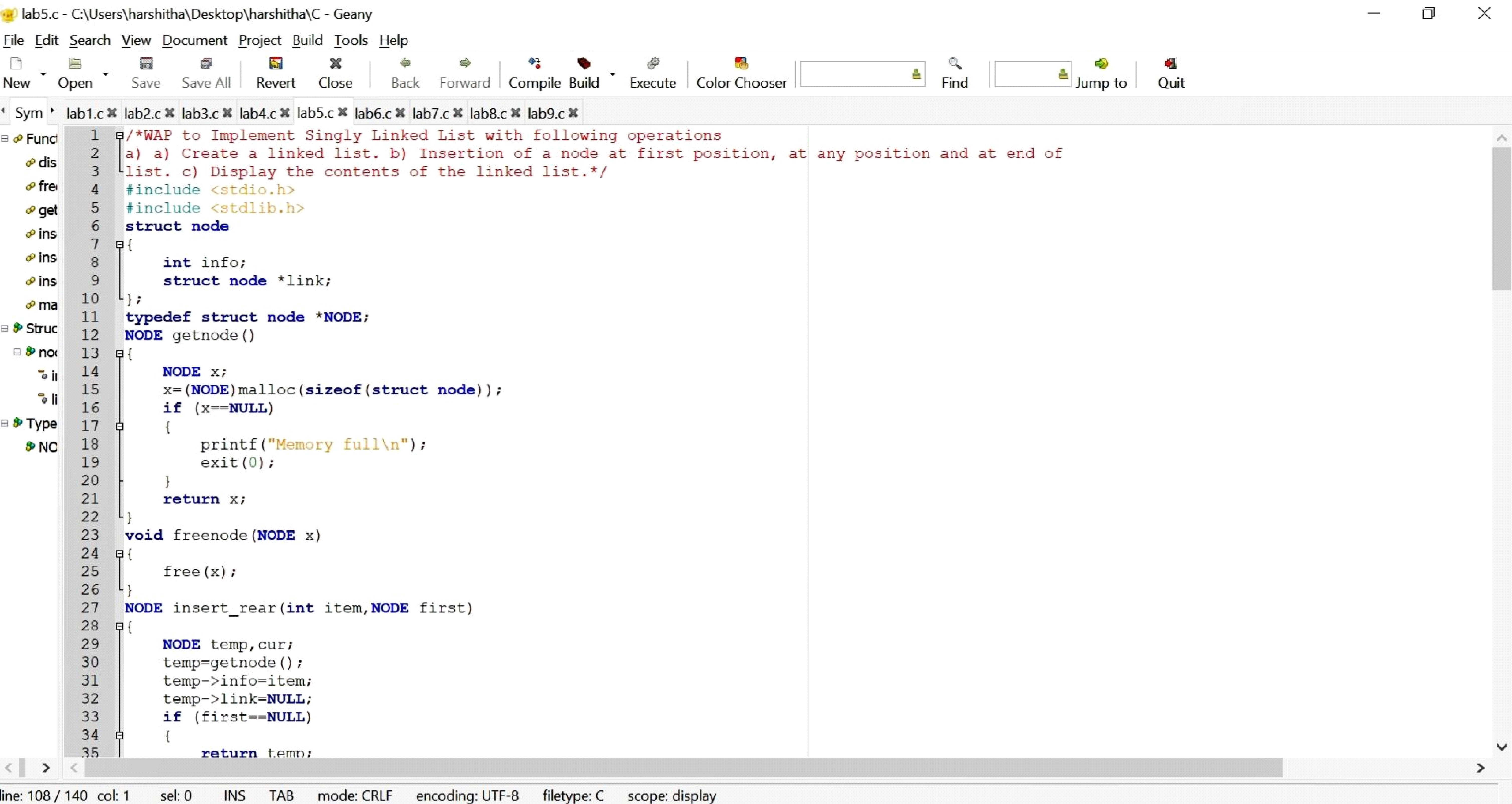
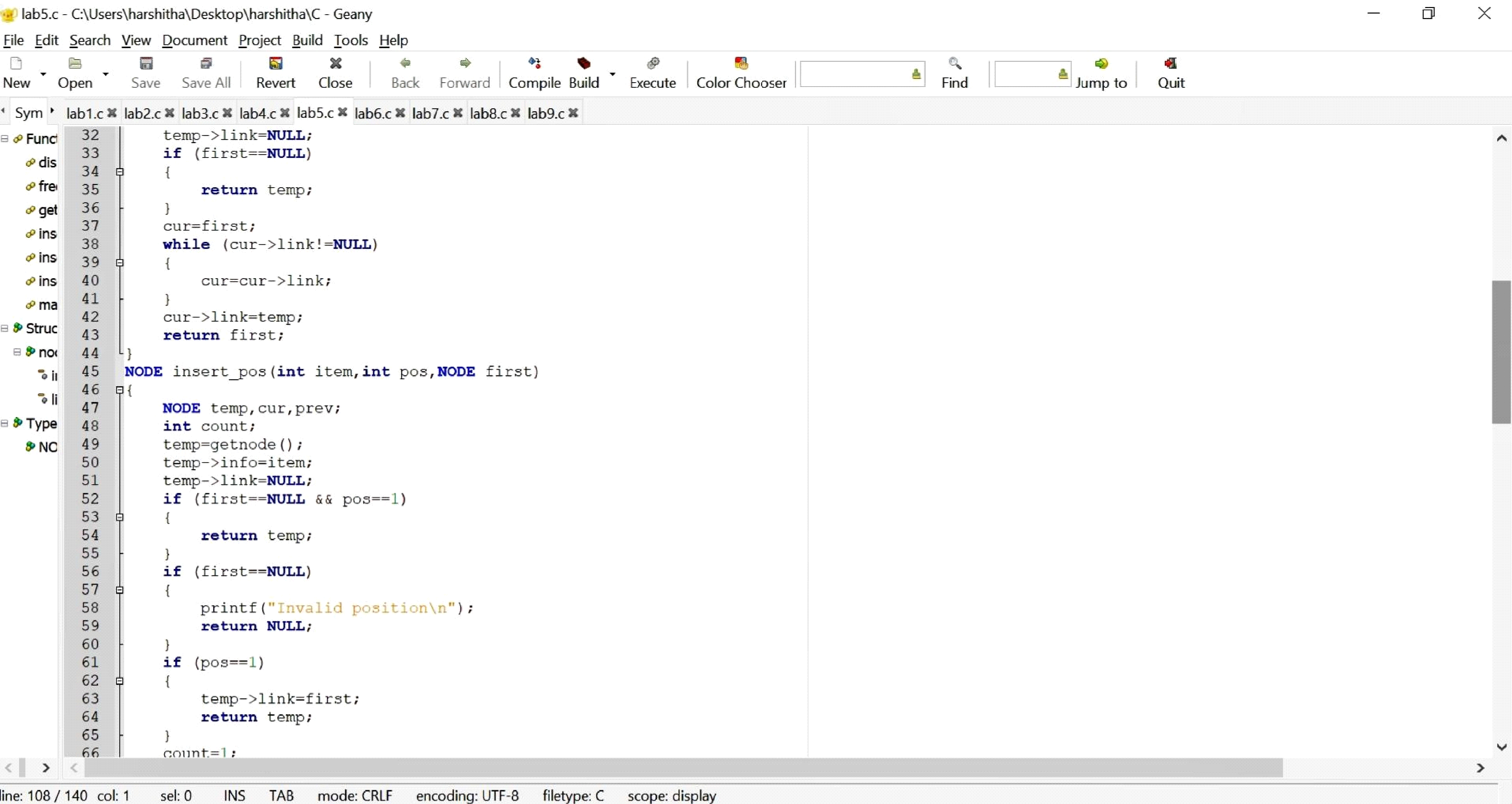


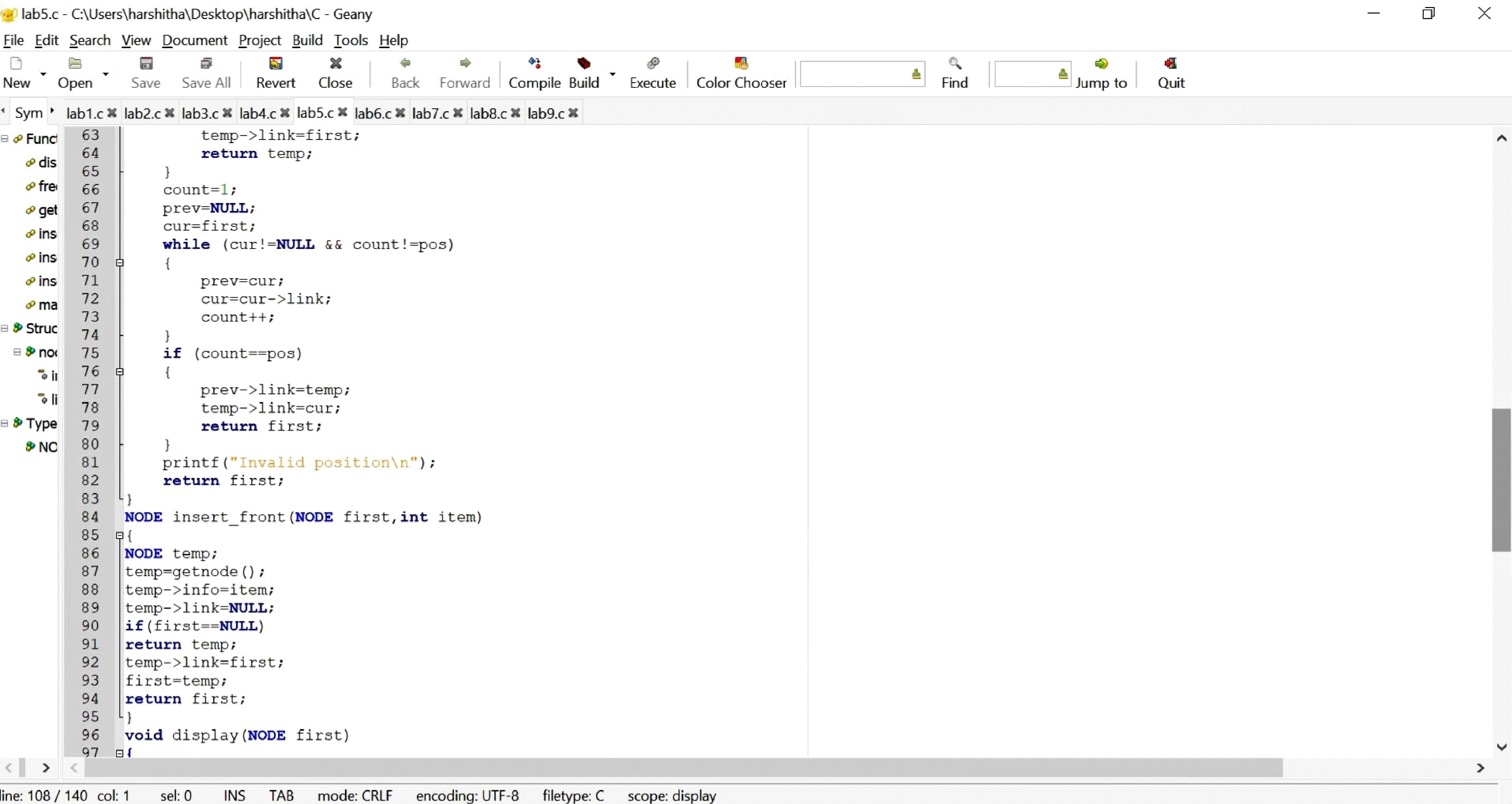
lab5.c - C:\Users\harshitha\Desktop\harshitha\C - Geany
File Edit Search View Document Project Build Tools Help
New
Open
Save
Save All
Revert
Close
Back
Forward
Compile
Build
Execute
Color Chooser
Find
Jump to
Quit
Sym
lab1.c
lab2.c
lab3.c
lab4.c
lab5.c
lab6.c
lab7.c
lab8.c
lab9.c
Func
dis
fre
get
ins
ins
ins
ma
Struc
no
i
li
Type
NO
line: 108 / 140 col: 1 sel: 0 INS TAB mode: CRLF encoding: UTF-8 filetype: C scope: display

```c
/*WAP to Implement Singly Linked List with following operations
a) a) Create a linked list. b) Insertion of a node at first position, at any position and at end of
list. c) Display the contents of the linked list.*/
#include <stdio.h>
#include <stdlib.h>
struct node
{
    int info;
    struct node *link;
};
typedef struct node *NODE;
NODE getnode()
{
    NODE x;
    x=(NODE)malloc(sizeof(struct node));
    if (x==NULL)
    {
        printf("Memory full\n");
        exit(0);
    }
    return x;
}
void freenode(NODE x)
{
    free(x);
}
NODE insert_rear(int item,NODE first)
{
    NODE temp,cur;
    temp=getnode();
    temp->info=item;
    temp->link=NULL;
    if (first==NULL)
    {
        return temp;
```



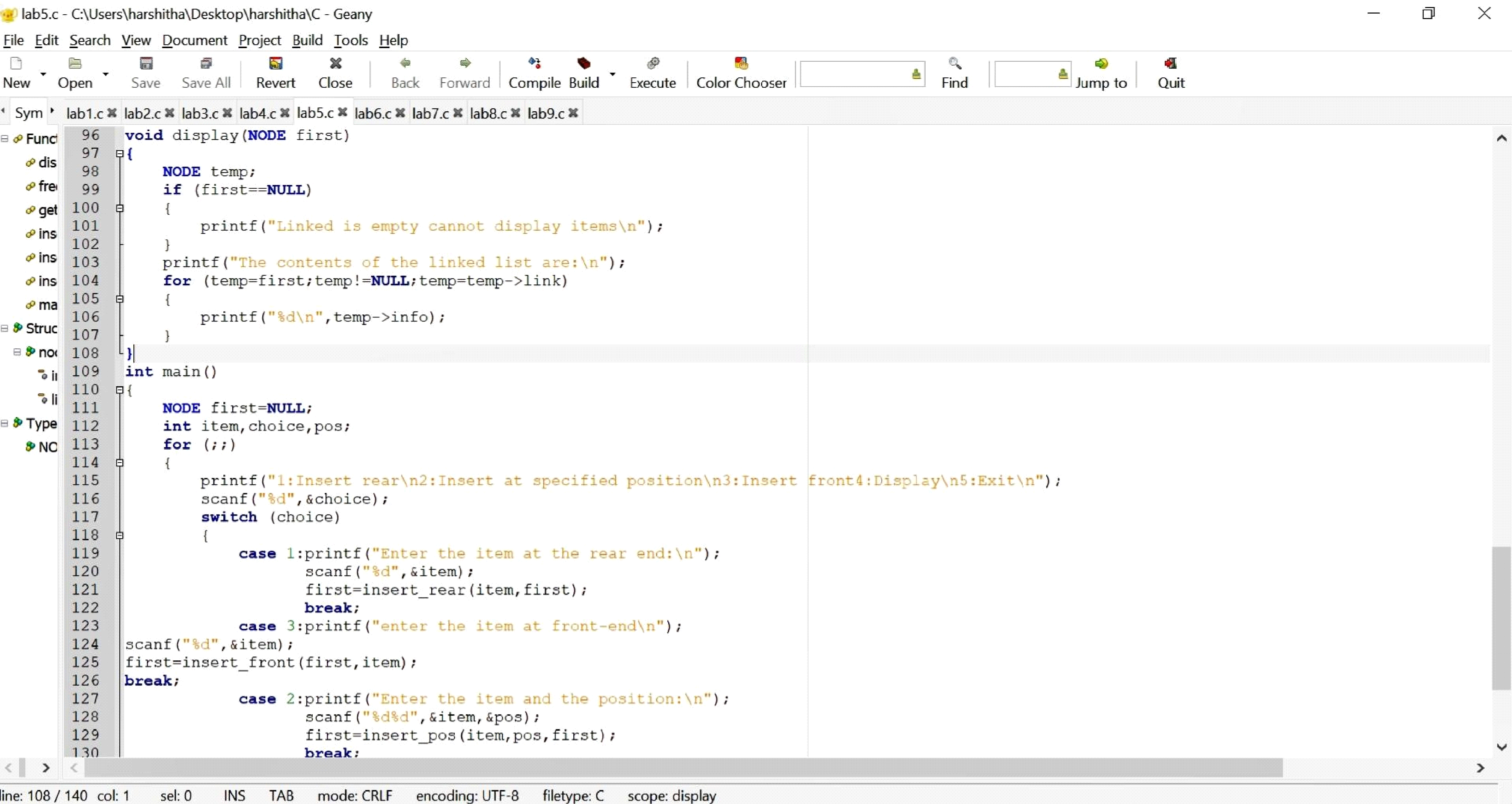
```c
    temp->link=NULL;
    if (first==NULL)
    {
        return temp;
    }
    cur=first;
    while (cur->link!=NULL)
    {
        cur=cur->link;
    }
    cur->link=temp;
    return first;
}
NODE insert_pos(int item,int pos,NODE first)
{
    NODE temp,cur,prev;
    int count;
    temp=getnode();
    temp->info=item;
    temp->link=NULL;
    if (first==NULL && pos==1)
    {
        return temp;
    }
    if (first==NULL)
    {
        printf("Invalid position\n");
        return NULL;
    }
    if (pos==1)
    {
        temp->link=first;
        return temp;
    }
    count=1;
```

line: 108 / 140 col: 1 sel: 0 INS TAB mode: CRLF encoding: UTF-8 filetype: C scope: display



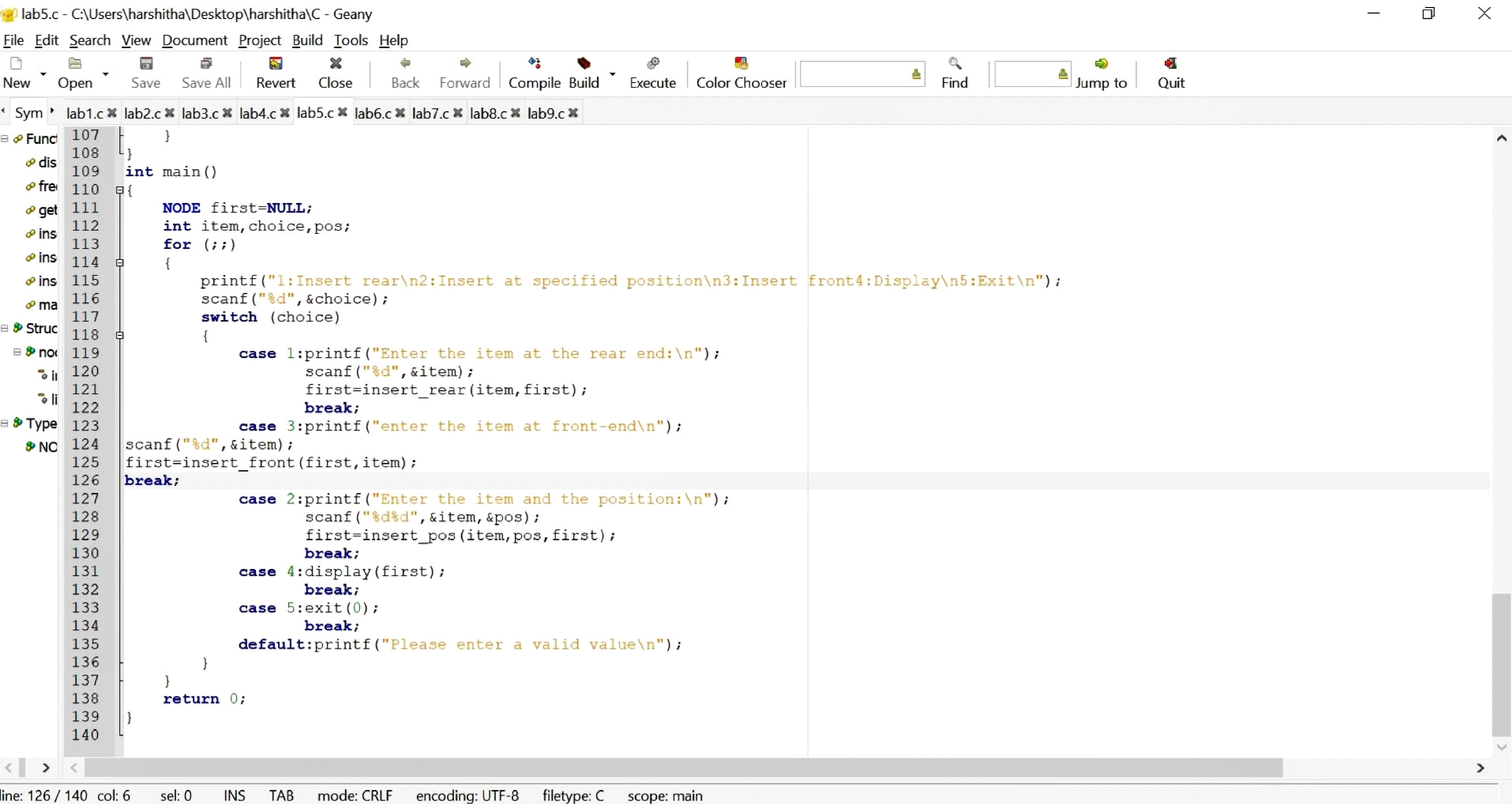
lab5.c - C:\Users\harshitha\Desktop\harshitha\C - Geany
File Edit Search View Document Project Build Tools Help
New
Open
Save
Save All
Revert
Close
Back
Forward
Compile
Build
Execute
Color Chooser
Find
Jump to
Quit
Sym
lab1.c lab2.c lab3.c lab4.c lab5.c lab6.c lab7.c lab8.c lab9.c
Func
dis
fre
get
ins
ins
ins
ma
Struc
no
i
li
Type
NC
line: 108 / 140 col: 1 sel: 0 INS TAB mode: CRLF encoding: UTF-8 filetype: C scope: display

```c
        temp->link=first;
        return temp;
    }
    count=1;
    prev=NULL;
    cur=first;
    while (cur!=NULL && count!=pos)
    {
        prev=cur;
        cur=cur->link;
        count++;
    }
    if (count==pos)
    {
        prev->link=temp;
        temp->link=cur;
        return first;
    }
    printf("Invalid position\n");
    return first;
}
NODE insert_front(NODE first,int item)
{
NODE temp;
temp=getnode();
temp->info=item;
temp->link=NULL;
if(first==NULL)
return temp;
temp->link=first;
first=temp;
return first;
}
void display(NODE first)
{
```



lab5.c - C:\Users\harshitha\Desktop\harshitha\C - Geany
File Edit Search View Document Project Build Tools Help
New
Open
Save
Save All
Revert
Close
Back
Forward
Compile
Build
Execute
Color Chooser
Find
Jump to
Quit
Sym
lab1.c
lab2.c
lab3.c
lab4.c
lab5.c
lab6.c
lab7.c
lab8.c
lab9.c
Func
dis
fre
get
ins
ins
ins
ma
Struc
no
i
li
Type
NO
line: 108 / 140 col: 1 sel: 0 INS TAB mode: CRLF encoding: UTF-8 filetype: C scope: display

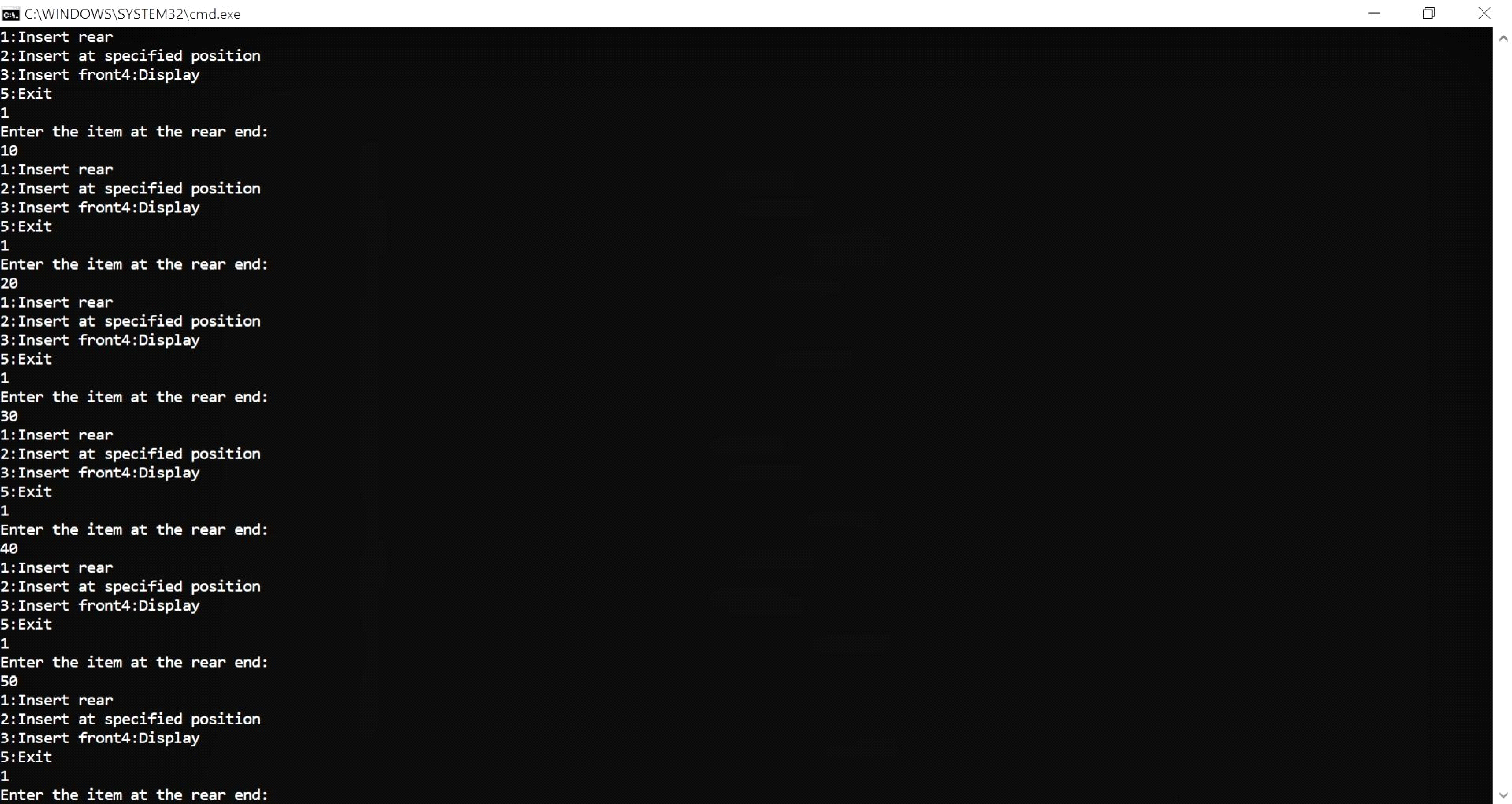
```
void display(NODE first)
{
    NODE temp;
    if (first==NULL)
    {
        printf("Linked is empty cannot display items\n");
    }
    printf("The contents of the linked list are:\n");
    for (temp=first;temp!=NULL;temp=temp->link)
    {
        printf("%d\n",temp->info);
    }
}
int main()
{
    NODE first=NULL;
    int item,choice,pos;
    for (;;)
    {
        printf("1:Insert rear\n2:Insert at specified position\n3:Insert front4:Display\n5:Exit\n");
        scanf("%d",&choice);
        switch (choice)
        {
            case 1:printf("Enter the item at the rear end:\n");
                   scanf("%d",&item);
                   first=insert_rear(item,first);
                   break;
            case 3:printf("enter the item at front-end\n");
scanf("%d",&item);
first=insert_front(first,item);
break;
            case 2:printf("Enter the item and the position:\n");
                   scanf("%d%d",&item,&pos);
                   first=insert_pos(item,pos,first);
                   break;
```



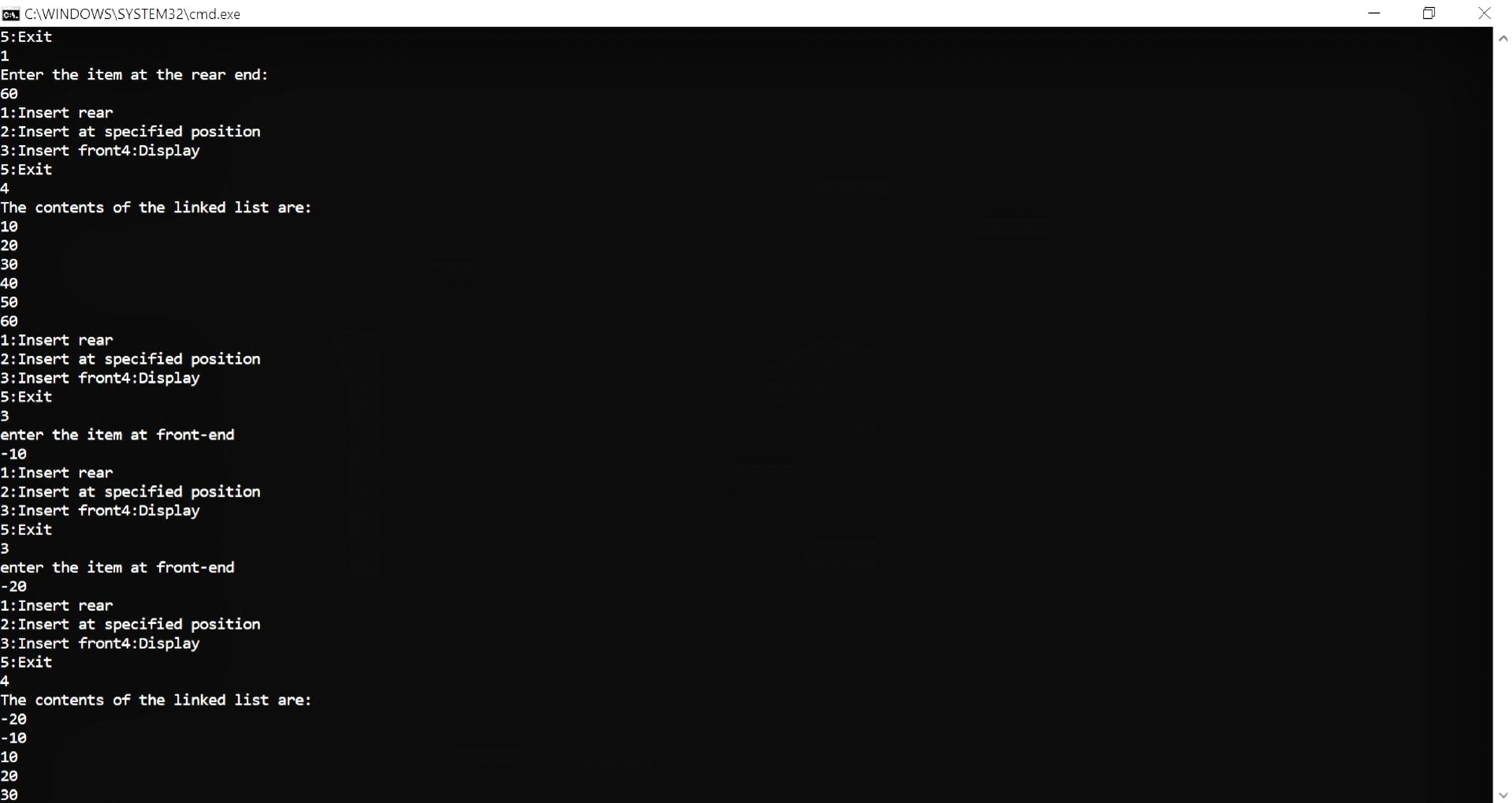
```
    }
}
int main()
{
    NODE first=NULL;
    int item,choice,pos;
    for (;;)
    {
        printf("1:Insert rear\n2:Insert at specified position\n3:Insert front4:Display\n5:Exit\n");
        scanf("%d",&choice);
        switch (choice)
        {
            case 1:printf("Enter the item at the rear end:\n");
                   scanf("%d",&item);
                   first=insert_rear(item,first);
                   break;
            case 3:printf("enter the item at front-end\n");
scanf("%d",&item);
first=insert_front(first,item);
break;
            case 2:printf("Enter the item and the position:\n");
                   scanf("%d%d",&item,&pos);
                   first=insert_pos(item,pos,first);
                   break;
            case 4:display(first);
                   break;
            case 5:exit(0);
                   break;
            default:printf("Please enter a valid value\n");
        }
    }
    return 0;
}

```

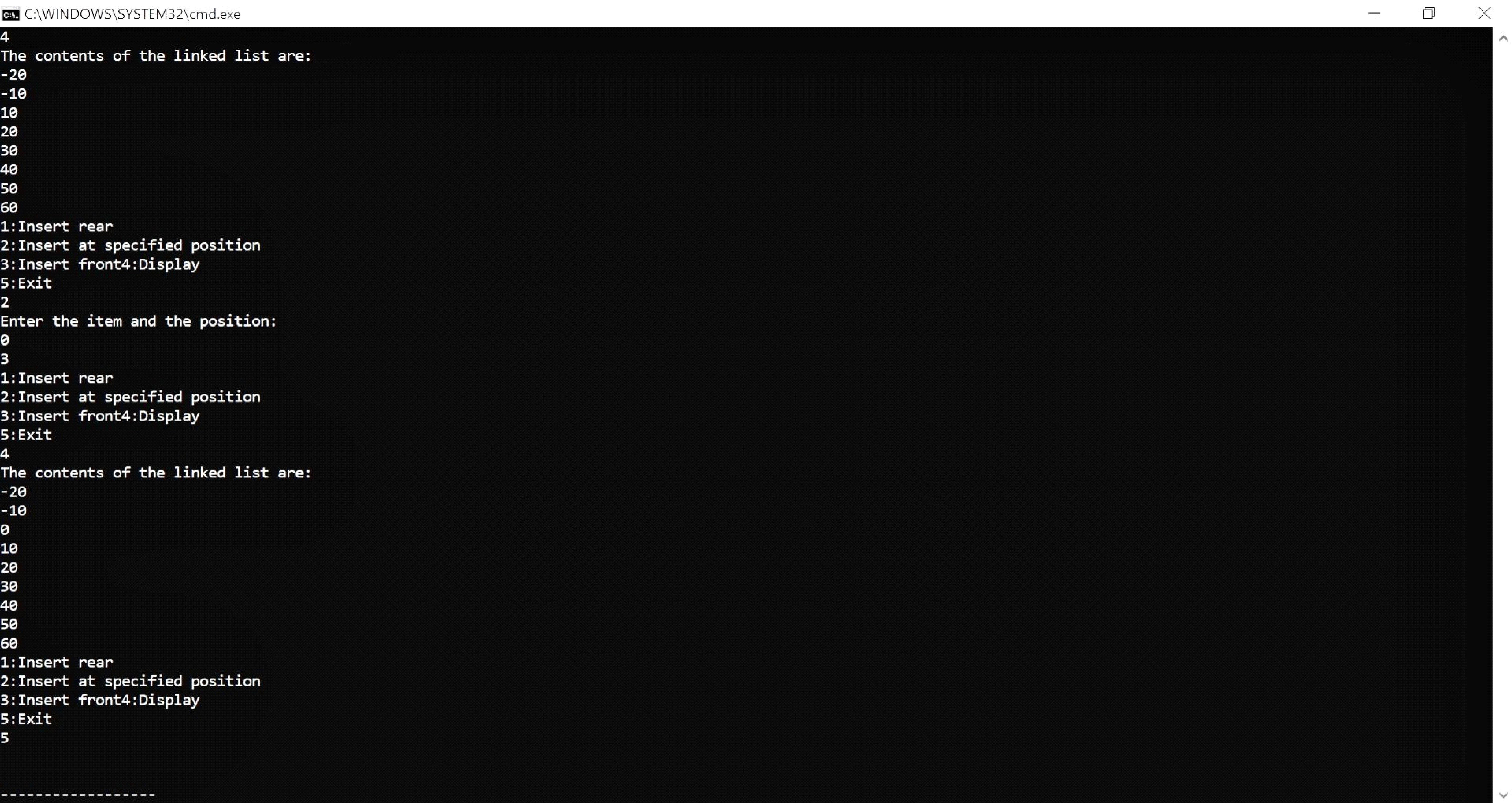
line: 126 / 140 col: 6 sel: 0 INS TAB mode: CRLF encoding: UTF-8 filetype: C scope: main



```
1:Insert rear
2:Insert at specified position
3:Insert front4:Display
5:Exit
1
Enter the item at the rear end:
10
1:Insert rear
2:Insert at specified position
3:Insert front4:Display
5:Exit
1
Enter the item at the rear end:
20
1:Insert rear
2:Insert at specified position
3:Insert front4:Display
5:Exit
1
Enter the item at the rear end:
30
1:Insert rear
2:Insert at specified position
3:Insert front4:Display
5:Exit
1
Enter the item at the rear end:
40
1:Insert rear
2:Insert at specified position
3:Insert front4:Display
5:Exit
1
Enter the item at the rear end:
50
1:Insert rear
2:Insert at specified position
3:Insert front4:Display
5:Exit
1
Enter the item at the rear end:
```



```
5:Exit
1
Enter the item at the rear end:
60
1:Insert rear
2:Insert at specified position
3:Insert front4:Display
5:Exit
4
The contents of the linked list are:
10
20
30
40
50
60
1:Insert rear
2:Insert at specified position
3:Insert front4:Display
5:Exit
3
enter the item at front-end
-10
1:Insert rear
2:Insert at specified position
3:Insert front4:Display
5:Exit
3
enter the item at front-end
-20
1:Insert rear
2:Insert at specified position
3:Insert front4:Display
5:Exit
4
The contents of the linked list are:
-20
-10
10
20
30
```



```
4
The contents of the linked list are:
-20
-10
10
20
30
40
50
60
1:Insert rear
2:Insert at specified position
3:Insert front4:Display
5:Exit
2
Enter the item and the position:
0
3
1:Insert rear
2:Insert at specified position
3:Insert front4:Display
5:Exit
4
The contents of the linked list are:
-20
-10
0
10
20
30
40
50
60
1:Insert rear
2:Insert at specified position
3:Insert front4:Display
5:Exit
5



------------------
```