



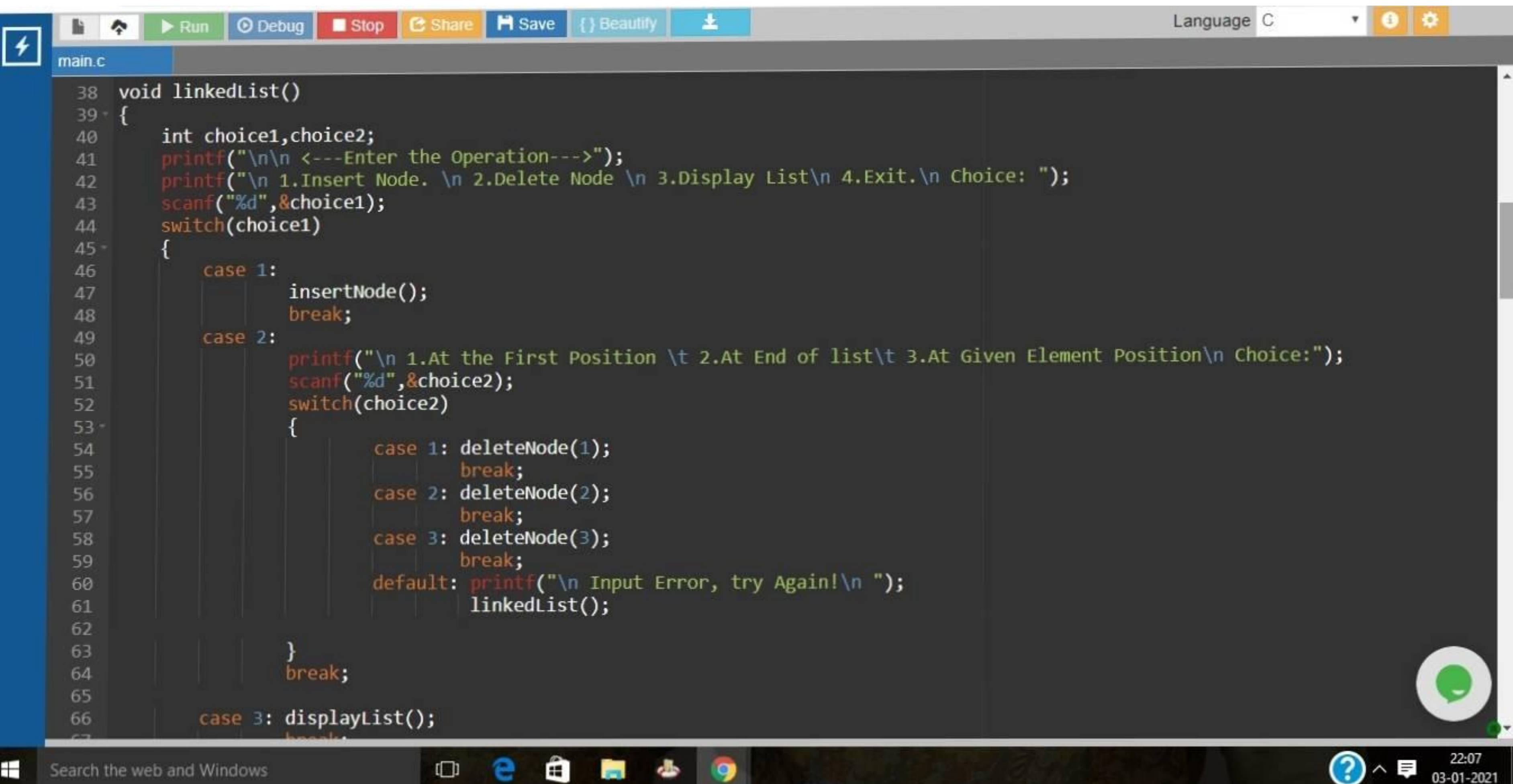
```
9 #include<stdio.h>
10 #include<stdlib.h>
11
12 struct node
13 {
14     int id;
15     char name[20];
16     int sem;
17     struct node *next;
18 };
19
20 struct node *head=NULL;
21
22 void linkedList();
23 void insertNode();
24 void deleteNode(int);
25 void deleteNodeAtBegin();
26 void deleteNodeAtEnd();
27 void deleteNodeOFGiven();
28 void displayList();
29
30 int size=0;
31
32 int main()
33 {
34     linkedList();
35     return 0;
36 }
37
38 void linkedList()
```

main.c

Language C

Search the web and Windows

22:06 03-01-2021

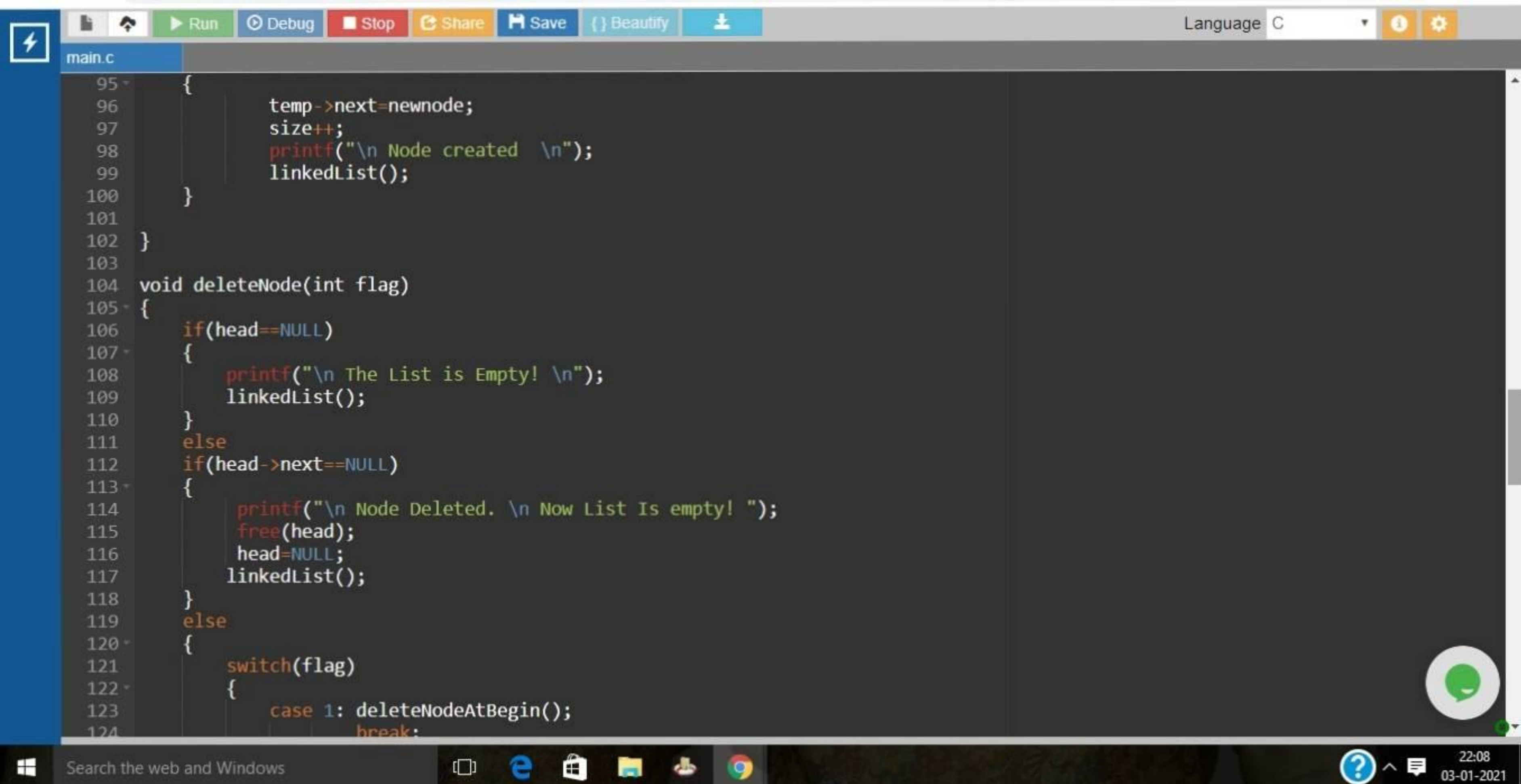


The image shows a screenshot of a C code editor with a dark theme. The editor has a toolbar at the top with buttons for Run, Debug, Stop, Share, Save, and Beautify. The language is set to C. The code is for a linked list and is displayed in a file named main.c. The code includes a void linkedList() function that prompts the user for an operation (1. Insert Node, 2. Delete Node, 3. Display List, 4. Exit) and then performs the corresponding action based on the user's choice. The deleteNode function is also shown, which prompts for a position (1. At the First Position, 2. At End of list, 3. At Given Element Position) and then deletes the node at that position. The displayList function is also shown, which displays the linked list.

```
38 void linkedList()
39 {
40     int choice1, choice2;
41     printf("\n\n <---Enter the Operation--->");
42     printf("\n 1.Insert Node. \n 2.Delete Node \n 3.Display List\n 4.Exit.\n Choice: ");
43     scanf("%d", &choice1);
44     switch(choice1)
45     {
46         case 1:
47             insertNode();
48             break;
49         case 2:
50             printf("\n 1.At the First Position \t 2.At End of list\t 3.At Given Element Position\n Choice:");
51             scanf("%d", &choice2);
52             switch(choice2)
53             {
54                 case 1: deleteNode(1);
55                     break;
56                 case 2: deleteNode(2);
57                     break;
58                 case 3: deleteNode(3);
59                     break;
60                 default: printf("\n Input Error, try Again!\n ");
61                     linkedList();
62             }
63             break;
64         case 3: displayList();
65             break;
66     }
```

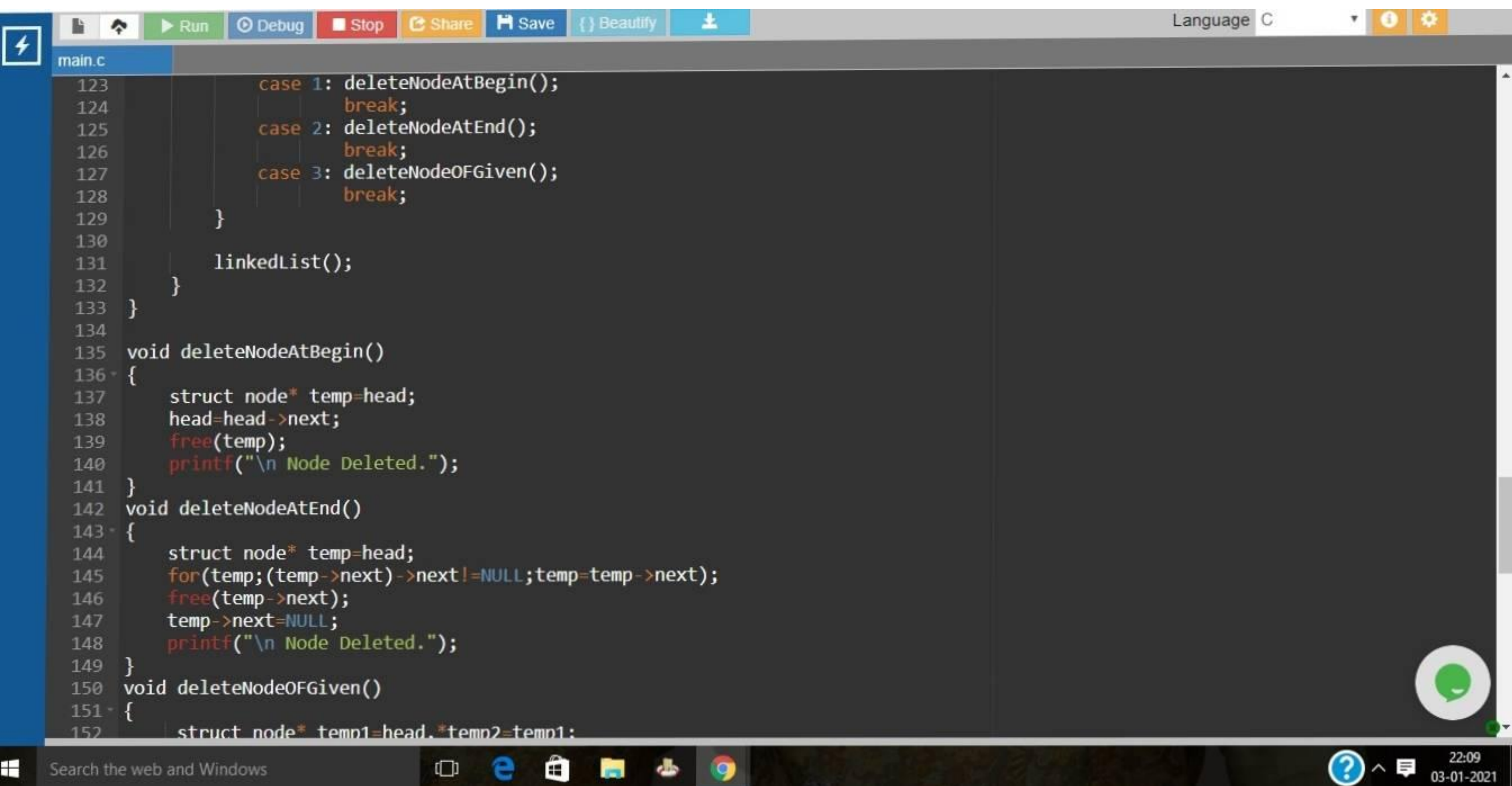


```
main.c
65
66     case 3: displayList();
67             break;
68
69     case 4: exit(0);
70
71     default: printf("\n Input error, Try again!!\n");
72             linkedList();
73 }
74 }
75
76 void insertNode()
77 {
78     struct node *newnode,*temp;
79     newnode =(struct node *) malloc (sizeof(struct node));
80     printf("\n <--Enter the Details--> ");
81     printf("\n ID: "); scanf("%d",&(newnode->id));
82     printf(" Name: ");  scanf("%s",(newnode->name));
83     printf(" Sem: ");  scanf("%d",&(newnode->sem));
84     if (head==NULL)
85     {
86         newnode->next=NULL;
87         head=newnode;
88         printf("\n Node created\n");
89         linkedList();
90         size++;
91     }
92     for(temp=head;(temp->next)!=NULL;temp=(temp->next));
93     newnode->next=NULL;
94 }
```



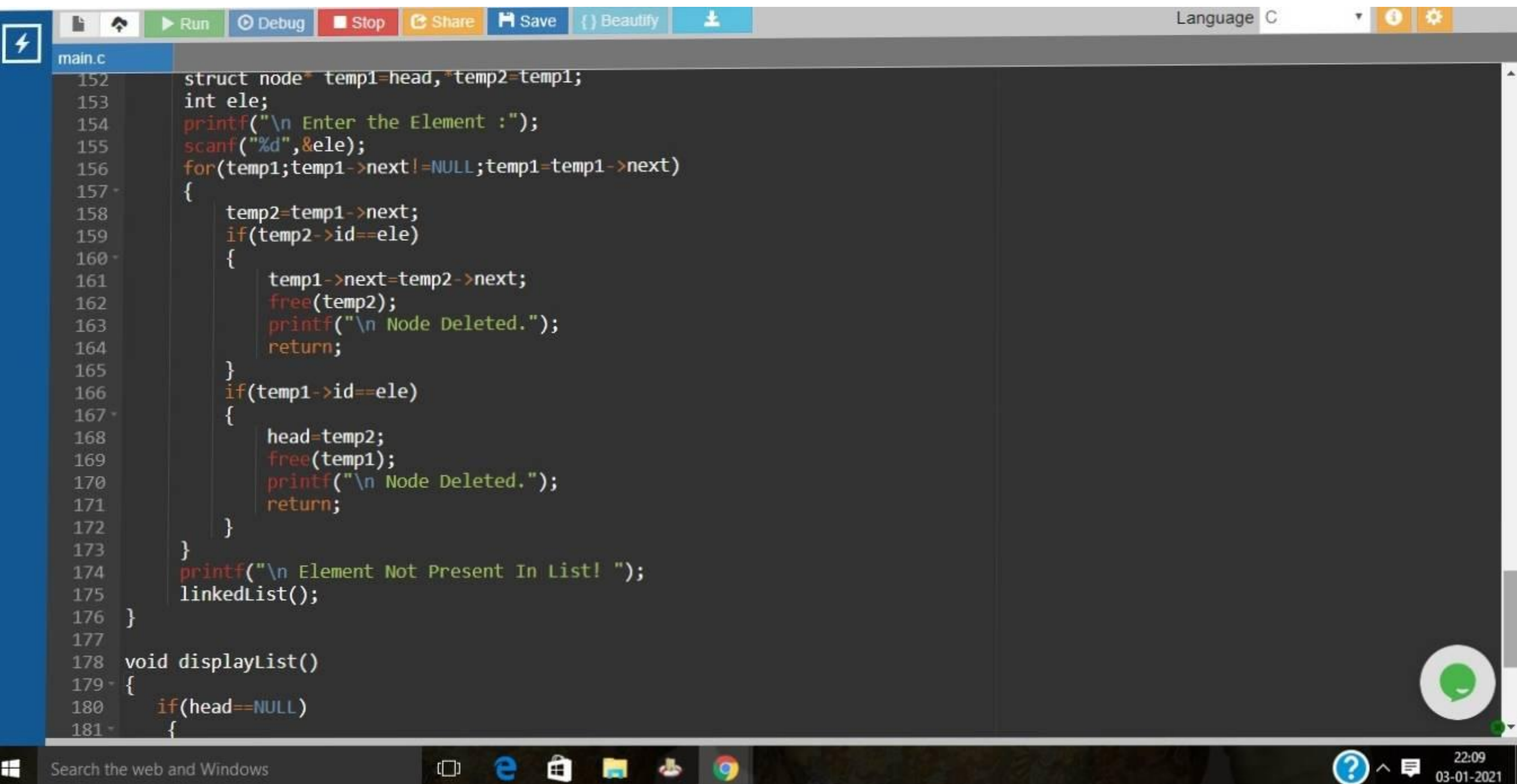
The image shows a screenshot of a C code editor. The editor has a dark theme and a sidebar on the left with a file explorer icon. The top bar contains several buttons: Run, Debug, Stop, Share, Save, Beautify, and a download icon. The language is set to C. The code is in a file named 'main.c' and shows a linked list implementation. The code includes a function to create a new node and a function to delete a node. The delete function uses a switch statement to handle different deletion cases.

```
95 {
96     temp->next=newnode;
97     size++;
98     printf("\n Node created \n");
99     linkedList();
100 }
101
102 }
103
104 void deleteNode(int flag)
105 {
106     if(head==NULL)
107     {
108         printf("\n The List is Empty! \n");
109         linkedList();
110     }
111     else
112     if(head->next==NULL)
113     {
114         printf("\n Node Deleted. \n Now List Is empty! ");
115         free(head);
116         head=NULL;
117         linkedList();
118     }
119     else
120     {
121         switch(flag)
122         {
123             case 1: deleteNodeAtBegin();
124                 break;
```

The image shows a code editor window with a dark theme. The editor displays C code for linked list operations. The code is organized into a main function and three helper functions: deleteNodeAtBegin(), deleteNodeAtEnd(), and deleteNodeOFGiven(). The main function calls these helper functions based on user input (1, 2, or 3) and then calls linkedList(). The helper functions perform the actual deletion of nodes from the linked list. The deleteNodeAtBegin() function removes the first node. The deleteNodeAtEnd() function removes the last node. The deleteNodeOFGiven() function removes a node at a specific position. The code is well-commented and includes error handling for invalid input.

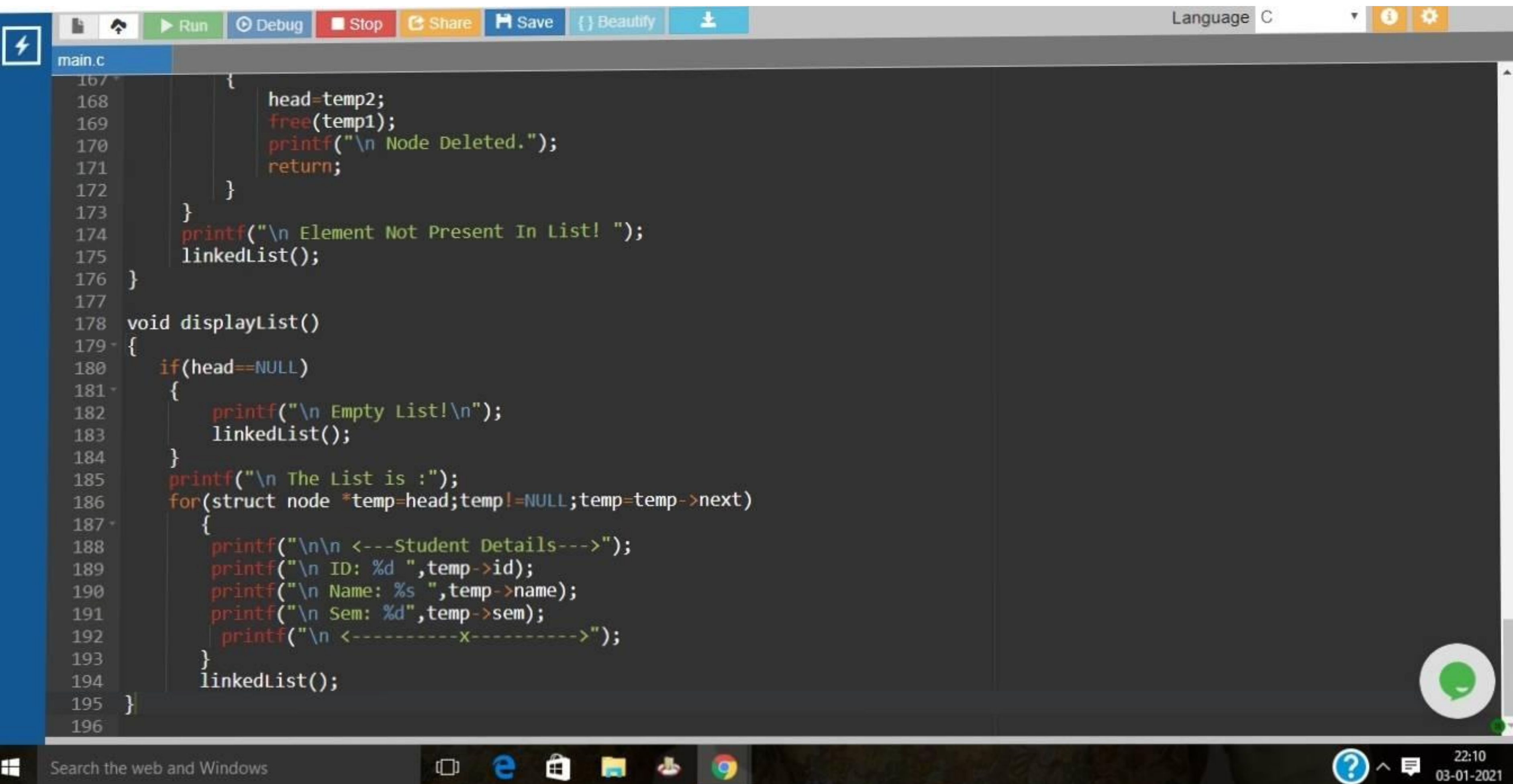
```
123         case 1: deleteNodeAtBegin();
124             break;
125         case 2: deleteNodeAtEnd();
126             break;
127         case 3: deleteNodeOFGiven();
128             break;
129     }
130
131     linkedList();
132 }
133 }
134
135 void deleteNodeAtBegin()
136 {
137     struct node* temp=head;
138     head=head->next;
139     free(temp);
140     printf("\n Node Deleted.");
141 }
142 void deleteNodeAtEnd()
143 {
144     struct node* temp=head;
145     for(temp;(temp->next)->next!=NULL;temp=temp->next);
146     free(temp->next);
147     temp->next=NULL;
148     printf("\n Node Deleted.");
149 }
150 void deleteNodeOFGiven()
151 {
152     struct node* temp1=head,*temp2=temp1;
```



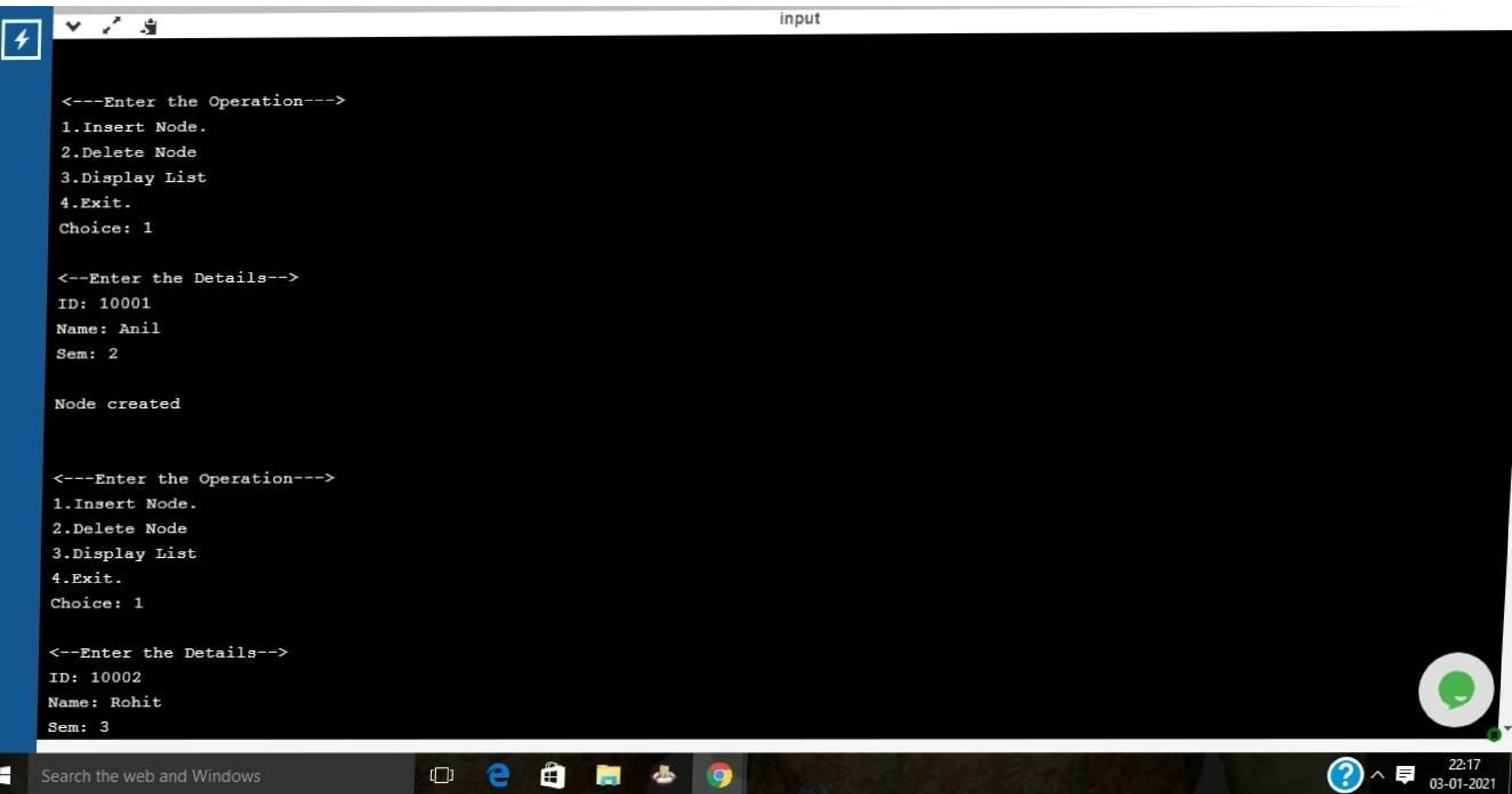
The image shows a screenshot of a C code editor. The editor has a dark theme and a sidebar on the left showing a file named 'main.c'. The code is written in C and implements a function to delete an element from a linked list. The code is as follows:

```
152 struct node* temp1=head,*temp2=temp1;
153 int ele;
154 printf("\n Enter the Element :");
155 scanf("%d",&ele);
156 for(temp1;temp1->next!=NULL;temp1=temp1->next)
157 {
158     temp2=temp1->next;
159     if(temp2->id==ele)
160     {
161         temp1->next=temp2->next;
162         free(temp2);
163         printf("\n Node Deleted.");
164         return;
165     }
166     if(temp1->id==ele)
167     {
168         head=temp2;
169         free(temp1);
170         printf("\n Node Deleted.");
171         return;
172     }
173 }
174 printf("\n Element Not Present In List! ");
175 linkedList();
176 }
177
178 void displayList()
179 {
180     if(head==NULL)
181     {
```

The editor's toolbar at the top includes buttons for Run, Debug, Stop, Share, Save, and Beautify. The language is set to C. The Windows taskbar at the bottom shows the search bar and several application icons. The system clock in the bottom right corner displays the time 22:09 and the date 03-01-2021.



```
167 {
168     head=temp2;
169     free(temp1);
170     printf("\n Node Deleted.");
171     return;
172 }
173 }
174 printf("\n Element Not Present In List! ");
175 linkedList();
176 }
177
178 void displayList()
179 {
180     if(head==NULL)
181     {
182         printf("\n Empty List!\n");
183         linkedList();
184     }
185     printf("\n The List is :");
186     for(struct node *temp=head;temp!=NULL;temp=temp->next)
187     {
188         printf("\n\n <---Student Details--->");
189         printf("\n ID: %d ",temp->id);
190         printf("\n Name: %s ",temp->name);
191         printf("\n Sem: %d",temp->sem);
192         printf("\n <-----x----->");
193     }
194     linkedList();
195 }
196
```




```
input
<---Enter the Operation--->
1.Insert Node.
2.Delete Node
3.Display List
4.Exit.
Choice: 1

<--Enter the Details-->
ID: 10003
Name: Mayank
Sem: 8

Node created

<---Enter the Operation--->
1.Insert Node.
2.Delete Node
3.Display List
4.Exit.
Choice: 3

The List is :

<---Student Details--->
ID: 10001
Name: Anil
Sem: 2
```

⚡

input

```
<-----x----->

<---Student Details--->
ID: 10002
Name: Rohit
Sem: 3
<-----x----->

<---Student Details--->
ID: 10003
Name: Mayank
Sem: 8
<-----x----->

<---Enter the Operation--->
1.Insert Node.
2.Delete Node
3.Display List
4.Exit.
Choice: 2

1.At the First Position      2.At End of list      3.At Given Element Position
Choice:1

Node Deleted.

<---Enter the Operation--->
1.Insert Node.
```

Search the web and Windows


```
input
<---Enter the Operation--->
1.Insert Node.
2.Delete Node
3.Display List
4.Exit.
Choice: 3

The List is :

<---Student Details--->
ID: 10002
Name: Rohit
Sem: 3
<-----x----->

<---Student Details--->
ID: 10003
Name: Mayank
Sem: 8
<-----x----->

<---Enter the Operation--->
1.Insert Node.
2.Delete Node
3.Display List
4.Exit.
Choice: 2
```




```
input
4.Exit.
Choice: 2

1.At the First Position      2.At End of list      3.At Given Element Position
Choice:1

Node Deleted.
Now List Is empty!

<---Enter the Operation--->
1.Insert Node.
2.Delete Node
3.Display List
4.Exit.
Choice: 3

Empty List!

<---Enter the Operation--->
1.Insert Node.
2.Delete Node
3.Display List
4.Exit.
Choice:

...Program finished with exit code 9
Press ENTER to exit console.
```