

```
9 #include<stdio.h>
10 #include<stdlib.h>
11
12 struct node *top=NULL;
13
14 struct node
15 {
16     int data;
17     struct node *next;
18 };
19
20 void stack();
21 void push(int);
22 int pop();
23 void display();
24
25 int main()
26 {
27     stack();
28     return 1;
29 }
30
31 void stack()
32 {
33     int choice=0,ele=0;
34     do
35     {
36         printf("\n Enter the choice:\n 1.Push.\n 2.Pop.\n 3.Display.\n 4.Exit\n Choice:");
37         scanf("%d",&choice);
38         switch(choice)
```

main.c

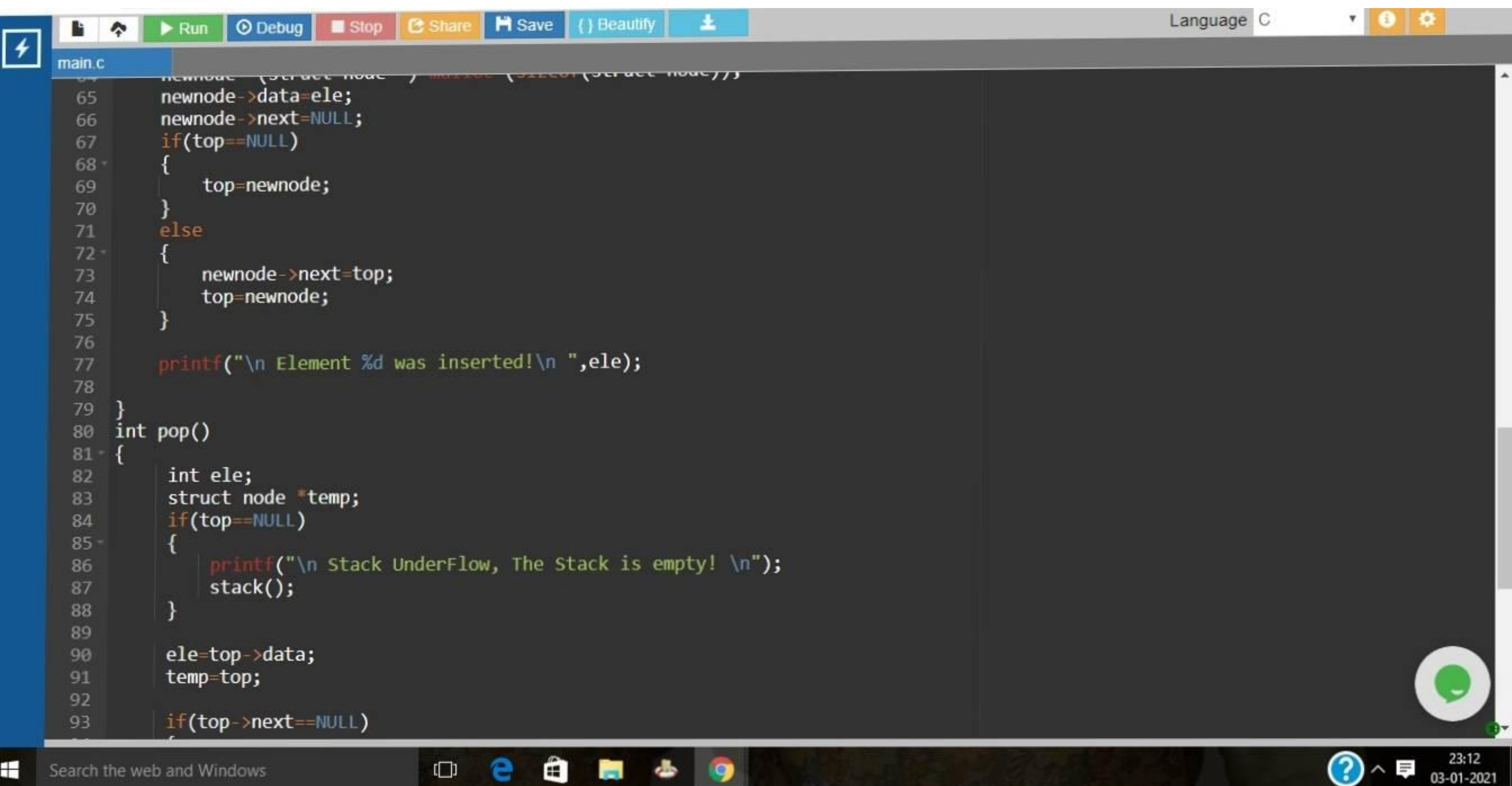
Language C

Run Debug Stop Share Save Beautify

Search the web and Windows

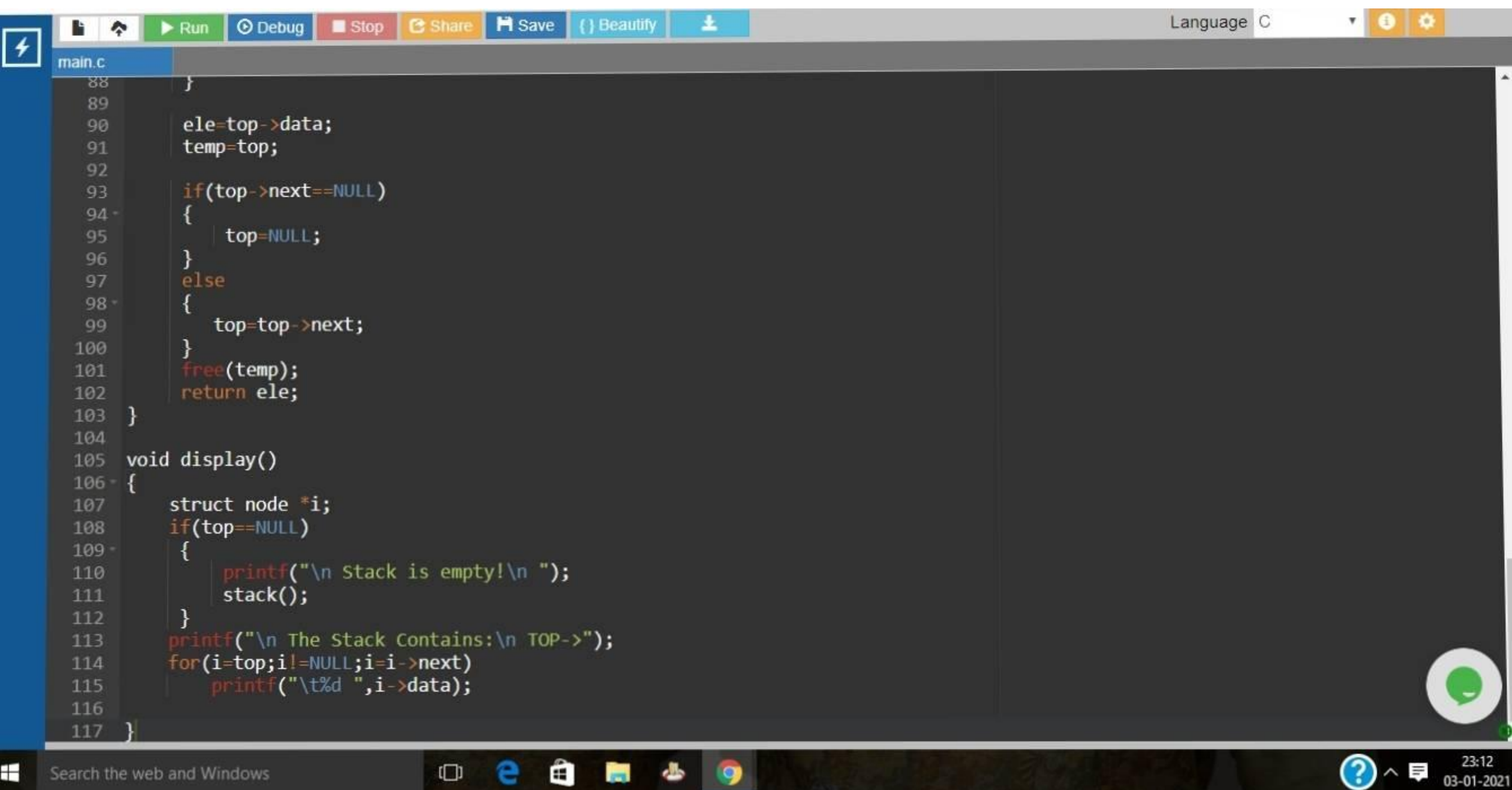
23:11 03-01-2021

```
main.c
36 printf("\n Enter the choice:\n 1.Push.\n 2.Pop.\n 3.Display.\n 4.Exit\n Choice: ");
37 scanf("%d",&choice);
38 switch(choice)
39 {
40     case 1:
41         printf("\n Enter the element to Push: ");
42         scanf("%d",&ele);
43         push(ele);
44         break;
45     case 2:
46         ele=pop();
47         printf("\n %d was deleted from Queue ",ele);
48         break;
49     case 3:
50         display();
51         break;
52     case 4:
53         exit(0);
54     default:
55         printf("\n Input Error Try Again! ");
56         stack();
57 }
58 }while(1);
59 }
60
61 void push(int ele)
62 {
63     struct node *newnode;
64     newnode =(struct node *) malloc (sizeof(struct node));
65     newnode->data=ele;
```

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. The main area displays C code for a stack implementation. The code includes functions for pushing and popping elements from a stack. The push function checks if the stack is empty and inserts a new node. The pop function checks if the stack is empty and removes the top element. The code is written in C and uses standard data structures like struct and pointers.

```
main.c
65 newnode->data=ele;
66 newnode->next=NULL;
67 if(top==NULL)
68 {
69     top=newnode;
70 }
71 else
72 {
73     newnode->next=top;
74     top=newnode;
75 }
76
77 printf("\n Element %d was inserted!\n ",ele);
78
79 }
80 int pop()
81 {
82     int ele;
83     struct node *temp;
84     if(top==NULL)
85     {
86         printf("\n Stack UnderFlow, The Stack is empty! \n");
87         stack();
88     }
89
90     ele=top->data;
91     temp=top;
92
93     if(top->next==NULL)
```



```
88 }
89
90 ele=top->data;
91 temp=top;
92
93 if(top->next==NULL)
94 {
95     top=NULL;
96 }
97 else
98 {
99     top=top->next;
100 }
101 free(temp);
102 return ele;
103 }
104
105 void display()
106 {
107     struct node *i;
108     if(top==NULL)
109     {
110         printf("\n Stack is empty!\n ");
111         stack();
112     }
113     printf("\n The Stack Contains:\n TOP->");
114     for(i=top;i!=NULL;i=i->next)
115         printf("\t%d ",i->data);
116 }
117 }
```

```
input
Enter the choice:
1.Push.
2.Pop.
3.Display.
4.Exit
Choice:1

Enter the element to Push: 10

Element 10 was inserted!

Enter the choice:
1.Push.
2.Pop.
3.Display.
4.Exit
Choice:1

Enter the element to Push: 20

Element 20 was inserted!

Enter the choice:
1.Push.
2.Pop.
3.Display.
4.Exit
```




```
input
3.Display.
4.Exit
Choice:3

The Stack Contains:
TOP-> 40      30      20      10
Enter the choice:
1.Push.
2.Pop.
3.Display.
4.Exit
Choice:2

40 was deleted from Queue
Enter the choice:
1.Push.
2.Pop.
3.Display.
4.Exit
Choice:2

30 was deleted from Queue
Enter the choice:
1.Push.
2.Pop.
3.Display.
4.Exit
Choice:3
```

```
input
4.Exit
Choice:3

The Stack Contains:
TOP-> 20      10
Enter the choice:
1.Push.
2.Pop.
3.Display.
4.Exit
Choice:2

20 was deleted from Queue
Enter the choice:
1.Push.
2.Pop.
3.Display.
4.Exit
Choice:2

10 was deleted from Queue
Enter the choice:
1.Push.
2.Pop.
3.Display.
4.Exit
Choice:2
```



```
input
10 was deleted from Queue
Enter the choice:
1.Push.
2.Pop.
3.Display.
4.Exit
Choice:2

Stack UnderFlow, The Stack is empty!

Enter the choice:
1.Push.
2.Pop.
3.Display.
4.Exit
Choice:3

Stack is empty!

Enter the choice:
1.Push.
2.Pop.
3.Display.
4.Exit
Choice:4

...Program finished with exit code 0
```