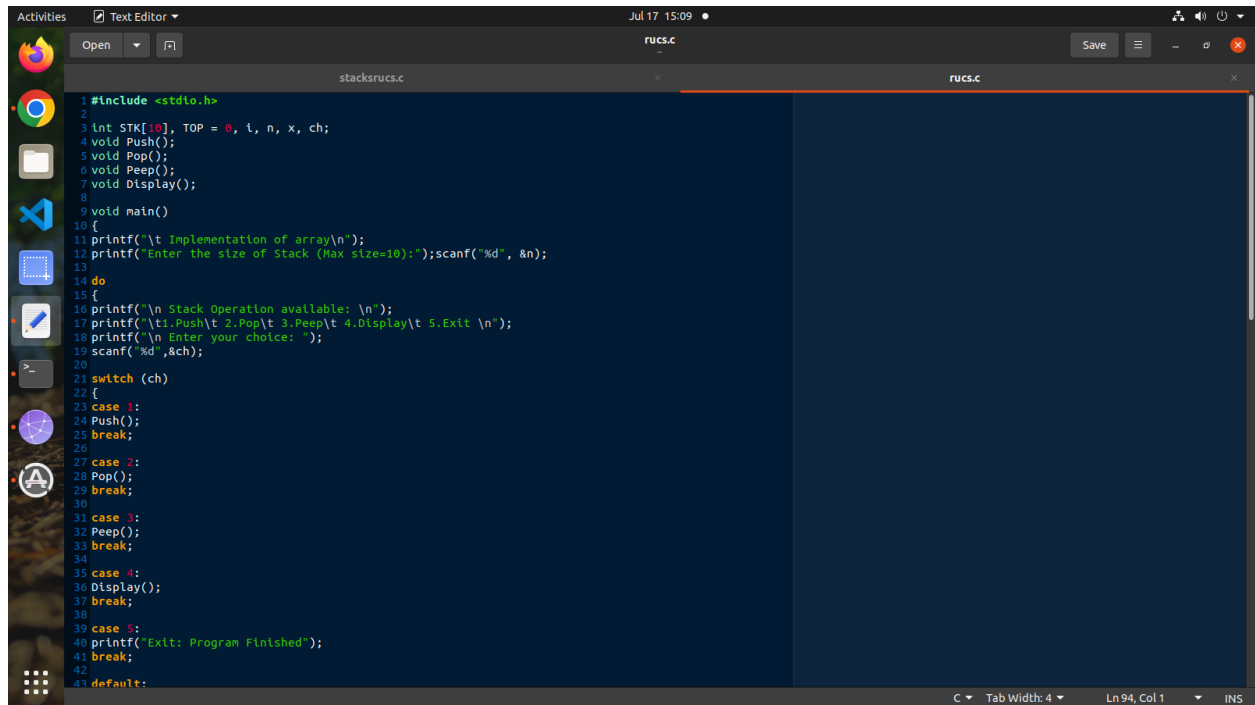
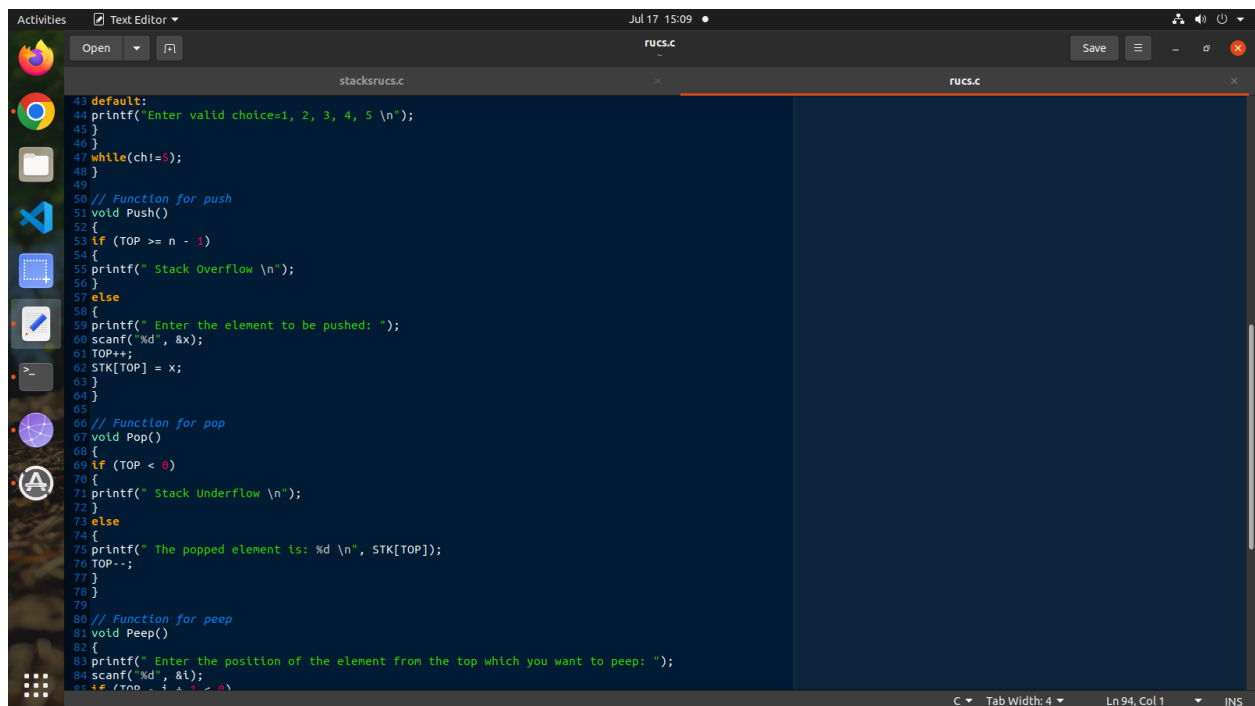


EXPERIMENT 1



This screenshot shows the first part of a C program in a text editor. The code includes standard headers, defines an array-based stack, and implements basic operations like push, pop, and display. The main function prompts the user for stack size and then enters a loop for operations.

```
1 #include <stdio.h>
2
3 int STK[10], TOP = 0, t, n, x, ch;
4 void Push();
5 void Pop();
6 void Peep();
7 void Display();
8
9 void main()
10 {
11     printf("\t Implementation of array\n");
12     printf("Enter the size of Stack (Max size=10):");scanf("%d", &n);
13
14     do
15     {
16         printf("\n Stack Operation available: \n");
17         printf("\t1.Push\t2.Pop\t3.Peep\t4.Display\t5.Exit \n");
18         printf("\n Enter your choice: ");
19         scanf("%d",&ch);
20
21         switch (ch)
22         {
23             case 1:
24                 Push();
25                 break;
26
27             case 2:
28                 Pop();
29                 break;
30
31             case 3:
32                 Peep();
33                 break;
34
35             case 4:
36                 Display();
37                 break;
38
39             case 5:
40                 printf("Exit: Program Finished");
41                 break;
42
43             default:
```



This screenshot shows the second part of the C program, continuing the implementation of the stack operations. It includes the 'default' case for the main loop, and detailed implementations for the Push, Pop, and Peep functions, including error handling for stack overflow and underflow.

```
43 default:
44     printf("Enter valid choices=1, 2, 3, 4, 5 \n");
45 }
46
47 while(ch!=5);
48
49
50 // Function for push
51 void Push()
52 {
53     if (TOP >= n - 1)
54     {
55         printf(" Stack Overflow \n");
56     }
57     else
58     {
59         printf(" Enter the element to be pushed: ");
60         scanf("%d", &x);
61         TOP++;
62         STK[TOP] = x;
63     }
64 }
65
66 // Function for pop
67 void Pop()
68 {
69     if (TOP < 0)
70     {
71         printf(" Stack Underflow \n");
72     }
73     else
74     {
75         printf(" The popped element is: %d \n", STK[TOP]);
76         TOP--;
77     }
78 }
79
80 // Function for peep
81 void Peep()
82 {
83     printf(" Enter the position of the element from the top which you want to peep: ");
84     scanf("%d", &t);
85     if (TOP - t + 1 < 0)
```

Activities Text Editor Jul 17 15:09

Open rucs.c Save

```
68 {
69 if (TOP < 0)
70 {
71 printf(" Stack Underflow \n");
72 }
73 else
74 {
75 printf(" The popped element is: %d \n", STK[TOP]);
76 TOP--;
77 }
78 }
79
80 // Function for peep
81 void Peep()
82 {
83 printf(" Enter the position of the element from the top which you want to peep: ");
84 scanf("%d", &t);
85 if (TOP - t + 1 < 0)
86 {
87 printf(" Stack Underflow on Peep \n");
88 }
89 else
90 {
91 printf(" The %d element from the top is: %d \n", t, STK[TOP - t + 1]);
92 }
93 }
94
95 // Function for display
96 void Display()
97 {
98 if (TOP < 1)
99 {
100 printf(" Stack is empty \n");
101 }
102 else
103 {
104 printf(" The element in the stack are:");
105 for (l=TOP; l>=0; l--)
106 {
107 printf("\n %d \n", STK[l]);
108 }
109 }
110 }
```

C Tab Width: 4 Ln 94, Col 1 INS

Activities Terminal Jul 17 15:09

dl0419@itadmin: ~

```
dl0419@itadmin:~$ gcc rucs.c
dl0419@itadmin:~$ ./a.out
Implementation of array
Enter the size of Stack (Max size=10):5
Stack Operation available:
1.Push 2.Pop 3.Peep 4.Display 5.Exit
Enter your choice: 1
Enter the element to be pushed: 3
Stack Operation available:
1.Push 2.Pop 3.Peep 4.Display 5.Exit
Enter your choice: 4
The element in the stack are:
3
Stack Operation available:
1.Push 2.Pop 3.Peep 4.Display 5.Exit
Enter your choice: 2
The popped element is: 3
Stack Operation available:
1.Push 2.Pop 3.Peep 4.Display 5.Exit
Enter your choice: 3
Enter the position of the element from the top which you want to peep: 3
Stack Underflow on Peep
Stack Operation available:
1.Push 2.Pop 3.Peep 4.Display 5.Exit
Enter your choice: 5
Exit: Program Finisheddl0419@itadmin:~$
```