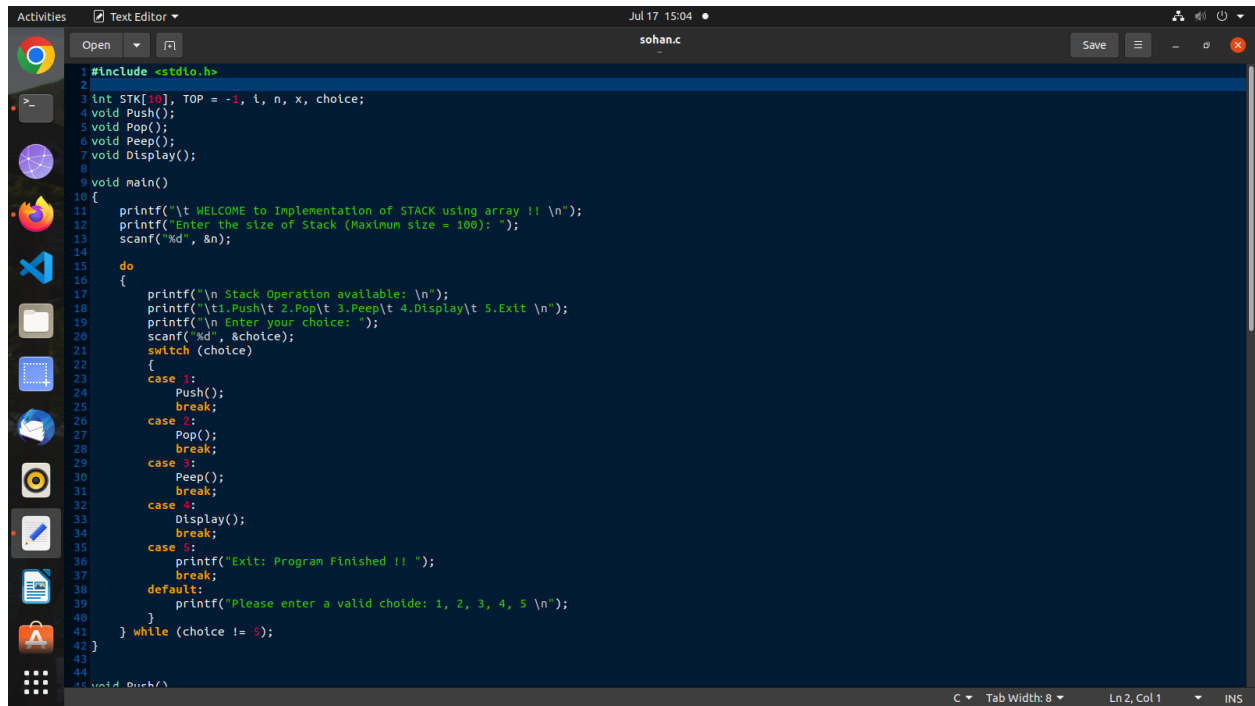
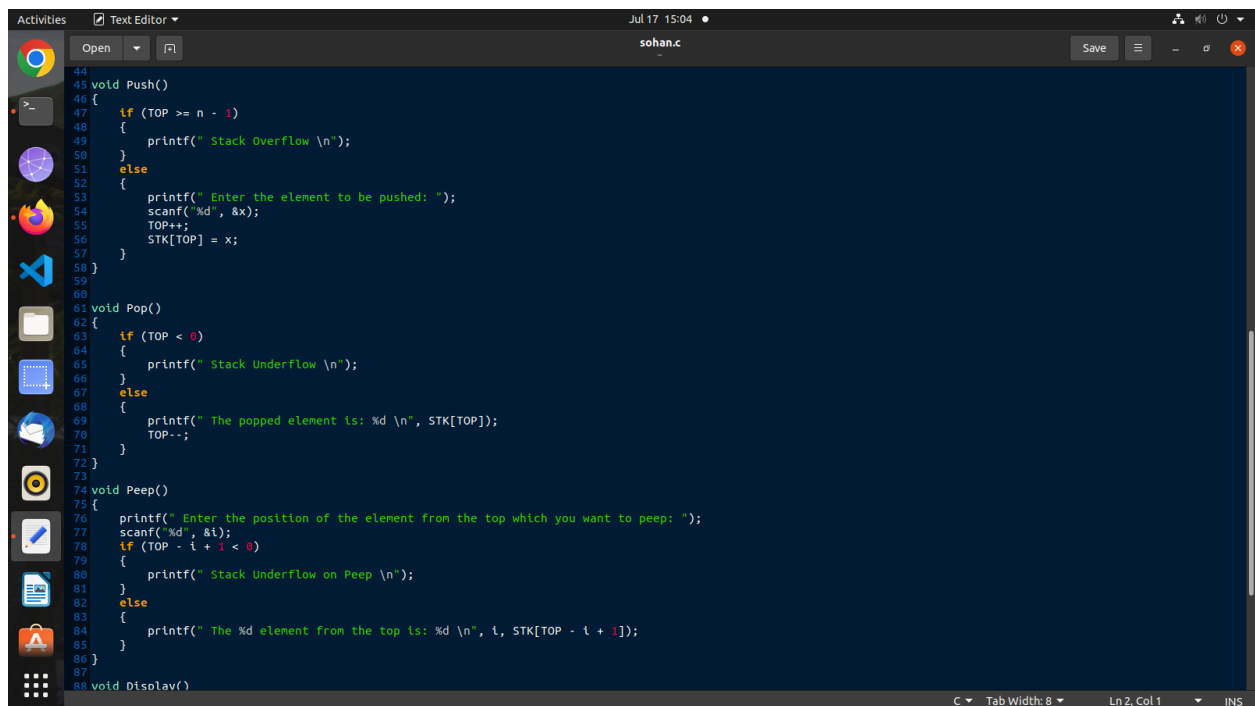


NAME: Sohansingh Rajput
ROLL NO.: 46
SY-IT



This screenshot shows a C program in a text editor. The program implements a stack using an array. It includes functions for push, pop, and peep operations. The main function prompts the user to enter the size of the stack and then enters a loop where the user can perform various operations. The stack is represented by an array of size 100, and the top of the stack is tracked using a variable 'TOP'.

```
1 #include <stdio.h>
2
3 int STK[100], TOP = -1, i, n, x, choice;
4 void Push();
5 void Pop();
6 void Peep();
7 void Display();
8
9 void main()
10 {
11     printf("\t WELCOME to Implementation of STACK using array !! \n");
12     printf("Enter the size of Stack (Maximum size = 100): ");
13     scanf("%d", &n);
14
15     do
16     {
17         printf("\n Stack Operation available: \n");
18         printf("\t1.Push\t2.Pop\t3.Peep\t4.Display\t5.Exit \n");
19         printf("\n Enter your choice: ");
20         scanf("%d", &choice);
21         switch (choice)
22         {
23             case 1:
24                 Push();
25                 break;
26             case 2:
27                 Pop();
28                 break;
29             case 3:
30                 Peep();
31                 break;
32             case 4:
33                 Display();
34                 break;
35             case 5:
36                 printf("Exit: Program Finished !! ");
37                 break;
38             default:
39                 printf("Please enter a valid choide: 1, 2, 3, 4, 5 \n");
40         }
41     } while (choice != 5);
42 }
43
44 void Push()
```



This screenshot shows the continuation of the C program, specifically the implementation of the push, pop, and peep functions. The push function checks for stack overflow before adding an element. The pop function checks for stack underflow before removing an element. The peep function checks for stack underflow before displaying an element from the stack.

```
45 void Push()
46 {
47     if (TOP >= n - 1)
48     {
49         printf(" Stack Overflow \n");
50     }
51     else
52     {
53         printf(" Enter the element to be pushed: ");
54         scanf("%d", &x);
55         TOP++;
56         STK[TOP] = x;
57     }
58 }
59
60 void Pop()
61 {
62     if (TOP < 0)
63     {
64         printf(" Stack Underflow \n");
65     }
66     else
67     {
68         printf(" The popped element is: %d \n", STK[TOP]);
69         TOP--;
70     }
71 }
72
73 void Peep()
74 {
75     printf(" Enter the position of the element from the top which you want to peep: ");
76     scanf("%d", &i);
77     if (TOP - i + 1 < 0)
78     {
79         printf(" Stack Underflow on Peep \n");
80     }
81     else
82     {
83         printf(" The %d element from the top is: %d \n", i, STK[TOP - i + 1]);
84     }
85 }
86
87 void Display()
88 {
89     for (i = 0; i <= TOP; i++)
90     {
91         printf("%d\t", STK[i]);
92     }
93     printf("\n");
94 }
```

```
Activities Text Editor Jul 17 15:04 sohan.c Save
259
60
61 void Pop()
62 {
63     if (TOP < 0)
64     {
65         printf(" Stack Underflow \n");
66     }
67     else
68     {
69         printf(" The popped element is: %d \n", STK[TOP]);
70         TOP--;
71     }
72 }
73
74 void Peep()
75 {
76     printf(" Enter the position of the element from the top which you want to peep: ");
77     scanf("%d", &t);
78     if (TOP - t + 1 < 0)
79     {
80         printf(" Stack Underflow on Peep \n");
81     }
82     else
83     {
84         printf(" The %d element from the top is: %d \n", t, STK[TOP - t + 1]);
85     }
86 }
87
88 void Display()
89 {
90     if (TOP < 0)
91     {
92         printf(" Stack is empty \n");
93     }
94     else
95     {
96         printf(" The element in the stack are:");
97         for (t = TOP; t > -1; t--)
98         {
99             printf("\n %d \n", STK[t]);
100         }
101     }
102 }
103
```

