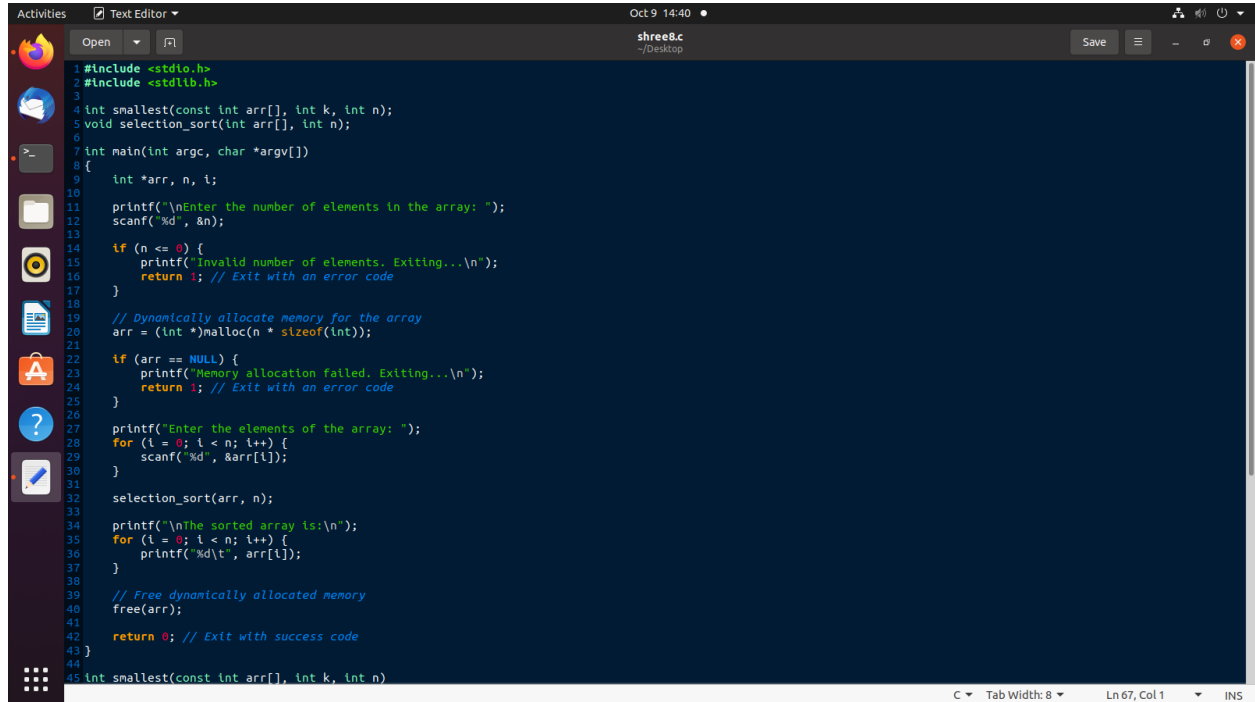


Name: Shreeganesh R Vishwakarma

Roll no. 65

Experiment: 08

Program:



The screenshot shows a Linux desktop environment with a terminal window open. The terminal window has a dark background and displays a C program for selection sort. The program includes headers for `stdio.h` and `stdlib.h`, defines a `smallest` function, and implements a `selection_sort` function. The `main` function prompts the user to enter the number of elements and the elements themselves, then sorts the array and prints the sorted array. The program also includes error handling for invalid input and memory allocation failures. The terminal window title is "shree8.c" and the file path is "~/Desktop". The status bar at the bottom indicates "C", "Tab Width: 8", "Ln 67, Col 1", and "INS".

```
1#include <stdio.h>
2#include <stdlib.h>
3
4int smallest(const int arr[], int k, int n);
5void selection_sort(int arr[], int n);
6
7int main(int argc, char *argv[])
8{
9    int *arr, n, i;
10
11    printf("\nEnter the number of elements in the array: ");
12    scanf("%d", &n);
13
14    if (n <= 0) {
15        printf("Invalid number of elements. Exiting...\n");
16        return 1; // Exit with an error code
17    }
18
19    // Dynamically allocate memory for the array
20    arr = (int *)malloc(n * sizeof(int));
21
22    if (arr == NULL) {
23        printf("Memory allocation failed. Exiting...\n");
24        return 1; // Exit with an error code
25    }
26
27    printf("Enter the elements of the array: ");
28    for (i = 0; i < n; i++) {
29        scanf("%d", &arr[i]);
30    }
31
32    selection_sort(arr, n);
33
34    printf("\nThe sorted array is:\n");
35    for (i = 0; i < n; i++) {
36        printf("%d\t", arr[i]);
37    }
38
39    // Free dynamically allocated memory
40    free(arr);
41
42    return 0; // Exit with success code
43}
44
45int smallest(const int arr[], int k, int n)
```

```
Activities Text Editor Oct 9 14:41 shree8.c ~/Desktop Save
23 printf("Memory allocation failed. Exiting...\n");
24 return 1; // Exit with an error code
25 }
26
27 printf("Enter the elements of the array: ");
28 for (i = 0; i < n; i++) {
29     scanf("%d", &arr[i]);
30 }
31
32 selection_sort(arr, n);
33
34 printf("\nThe sorted array is:\n");
35 for (i = 0; i < n; i++) {
36     printf("%d\t", arr[i]);
37 }
38
39 // Free dynamically allocated memory
40 free(arr);
41
42 return 0; // Exit with success code
43 }
44
45 int smallest(const int arr[], int k, int n)
46 {
47     int pos = k, small = arr[k], i;
48     for (i = k + 1; i < n; i++) {
49         if (arr[i] < small) {
50             small = arr[i];
51             pos = i;
52         }
53     }
54     return pos;
55 }
56
57 void selection_sort(int arr[], int n)
58 {
59     int k, pos, temp;
60     for (k = 0; k < n; k++) {
61         pos = smallest(arr, k, n);
62         temp = arr[k];
63         arr[k] = arr[pos];
64         arr[pos] = temp;
65     }
66 }
67
```

```
Activities Terminal Oct 9 14:39 dl403@dl403-HP-ProDesk-400-G7-Microtower-PC: ~
Experiment No. 08 (Impl: X) Ritika Jain - Experiment: X Experiment No. 08 (Impl: X) Experiment No. 08 (Impl: X) Experiment No. 07 (Impl: X) +
dl403@dl403-HP-ProDesk-400-G7-Microtower-PC:~$ gcc shree8.c
dl403@dl403-HP-ProDesk-400-G7-Microtower-PC:~$ ./a.out
Enter the number of elements in the array: 5
Enter the elements of the array: 2
8
6
5
1
The sorted array is:
1 2 5 6 8 dl403@dl403-HP-ProDesk-400-G7-Microtower-PC:~$
```

```
// Dynamically allocate memory for the array
arr = (int *)malloc(n * sizeof(int));

if (arr == NULL) {
    printf("Memory allocation failed. Exiting...\n");
    return 1; // Exit with an error code
}
```