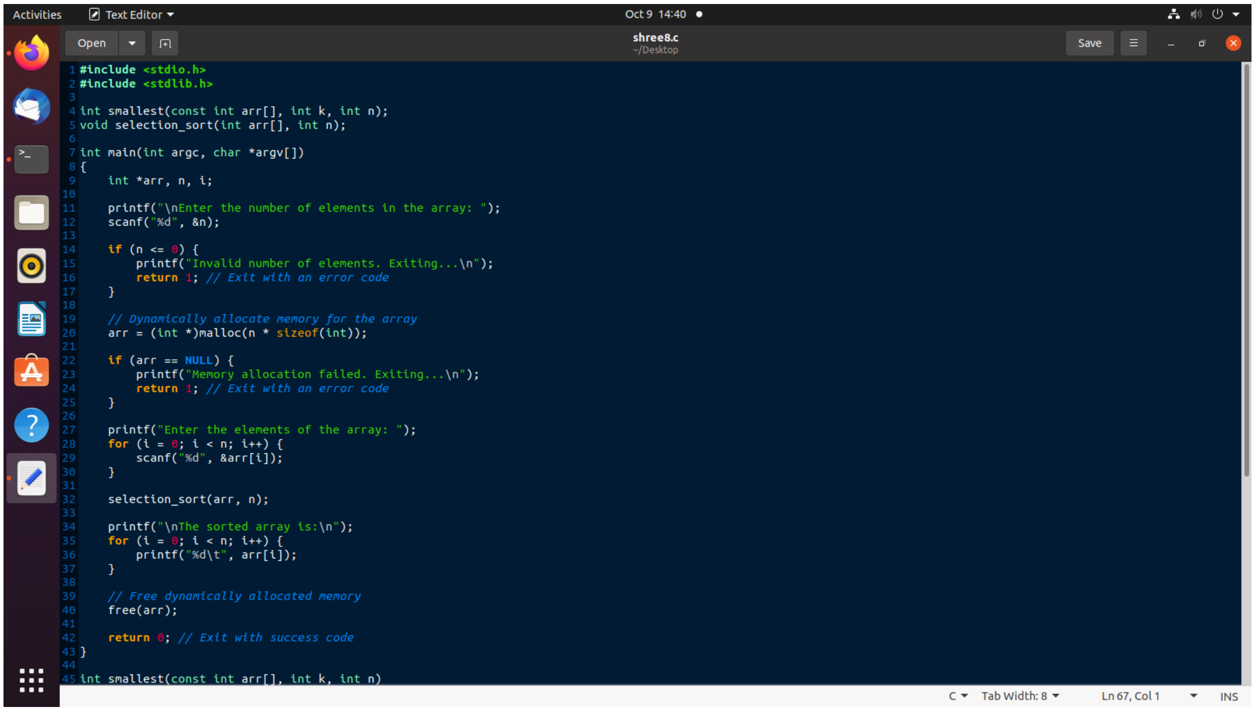
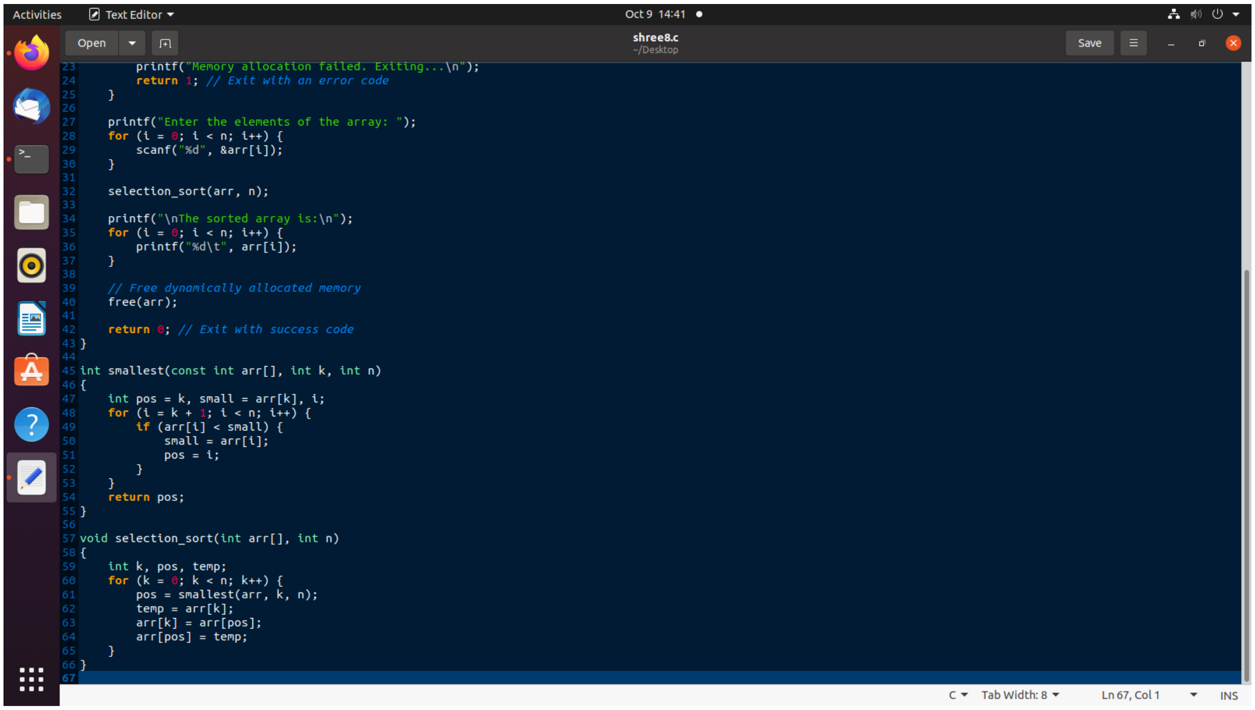
Name: Shreeganesh R Vishwakarma
Roll no. 65
Experiment: 08

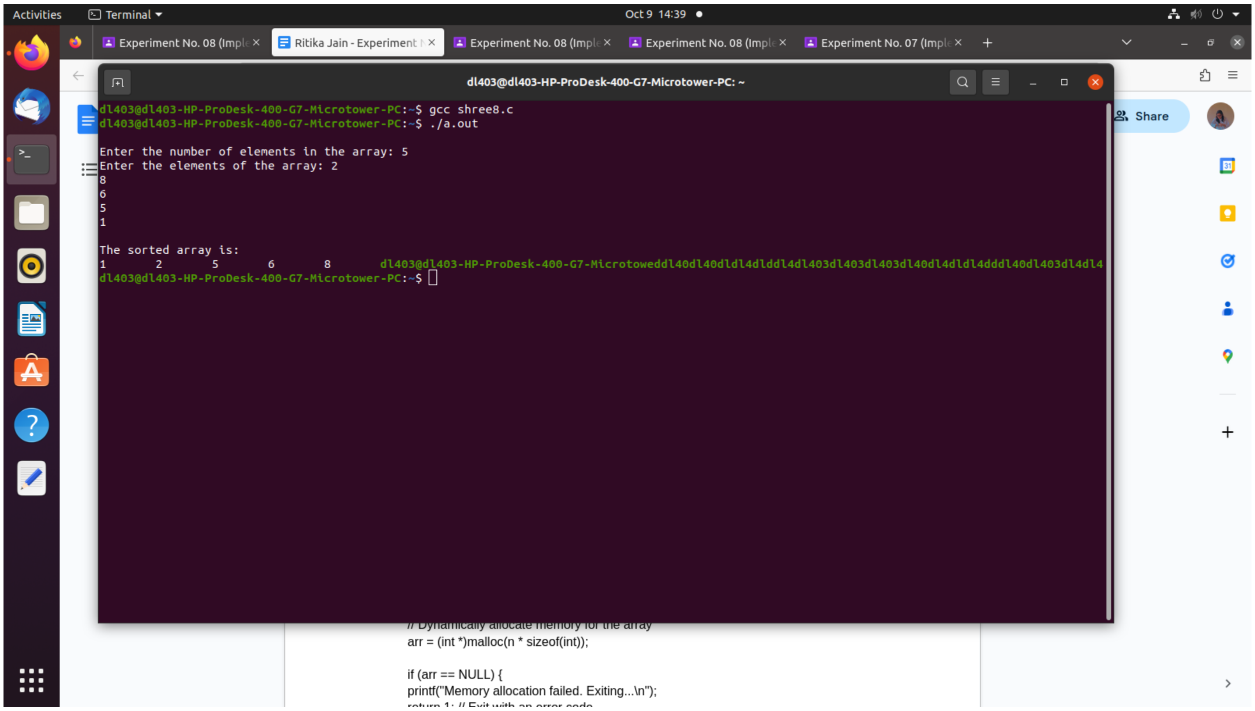
Program:



```c
#include <stdio.h>
#include <stdlib.h>

int smallest(const int arr[], int k, int n);
void selection_sort(int arr[], int n);

int main(int argc, char *argv[])
{
    int *arr, n, i;

    printf("\nEnter the number of elements in the array: ");
    scanf("%d", &n);

    if (n <= 0) {
        printf("Invalid number of elements. Exiting...\n");
        return 1; // Exit with an error code
    }

    // 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
    }

    printf("Enter the elements of the array: ");
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    selection_sort(arr, n);

    printf("\nThe sorted array is:\n");
    for (i = 0; i < n; i++) {
        printf("%d\t", arr[i]);
    }

    // Free dynamically allocated memory
    free(arr);

    return 0; // Exit with success code
}

int smallest(const int arr[], int k, int n)
```



```c
        printf("Memory allocation failed. Exiting...\n");
        return 1; // Exit with an error code
    }

    printf("Enter the elements of the array: ");
    for (i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    selection_sort(arr, n);

    printf("\nThe sorted array is:\n");
    for (i = 0; i < n; i++) {
        printf("%d\t", arr[i]);
    }

    // Free dynamically allocated memory
    free(arr);

    return 0; // Exit with success code
}

int smallest(const int arr[], int k, int n)
{
    int pos = k, small = arr[k], i;
    for (i = k + 1; i < n; i++) {
        if (arr[i] < small) {
            small = arr[i];
            pos = i;
        }
    }
    return pos;
}

void selection_sort(int arr[], int n)
{
    int k, pos, temp;
    for (k = 0; k < n; k++) {
        pos = smallest(arr, k, n);
        temp = arr[k];
        arr[k] = arr[pos];
        arr[pos] = temp;
    }
}
```



```
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-Microtoweddl40dl40dldl4dlddl4dl403dl403dl403dl40dl4dldl4dddl40dl403dl4dl4
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
```