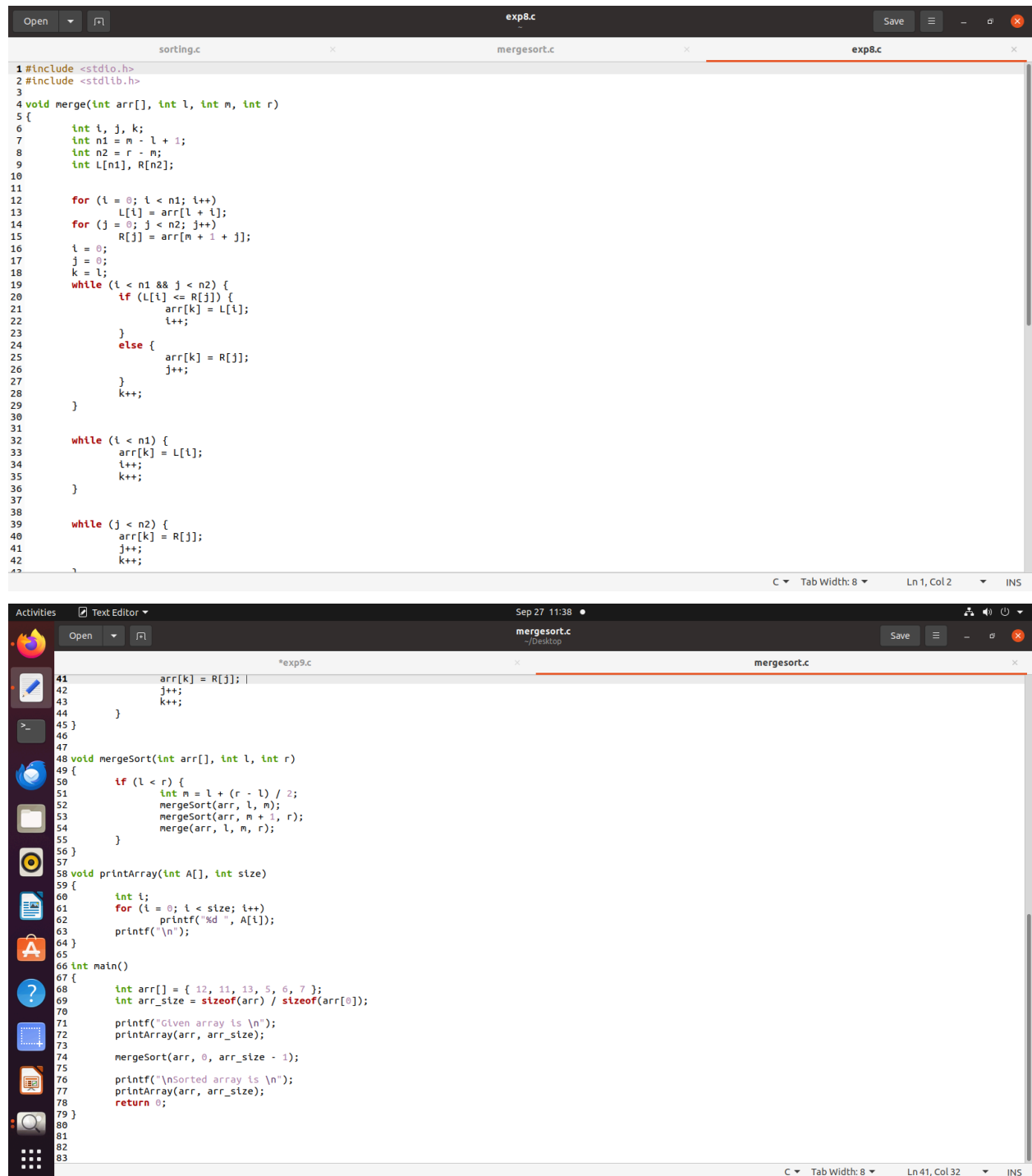


EXPERIMENT NO.08

CODE:



```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 void merge(int arr[], int l, int m, int r)
5 {
6     int i, j, k;
7     int n1 = m - l + 1;
8     int n2 = r - m;
9     int L[n1], R[n2];
10
11
12     for (i = 0; i < n1; i++)
13         L[i] = arr[l + i];
14     for (j = 0; j < n2; j++)
15         R[j] = arr[m + 1 + j];
16
17     i = 0;
18     j = 0;
19     k = l;
20     while (i < n1 && j < n2) {
21         if (L[i] <= R[j]) {
22             arr[k] = L[i];
23             i++;
24         }
25         else {
26             arr[k] = R[j];
27             j++;
28         }
29         k++;
30     }
31
32     while (i < n1) {
33         arr[k] = L[i];
34         i++;
35         k++;
36     }
37
38     while (j < n2) {
39         arr[k] = R[j];
40         j++;
41         k++;
42     }
43 }
```

```
41     arr[k] = R[j];
42     j++;
43     k++;
44 }
45 }
46
47 void mergeSort(int arr[], int l, int r)
48 {
49     if (l < r) {
50         int m = l + (r - l) / 2;
51         mergeSort(arr, l, m);
52         mergeSort(arr, m + 1, r);
53         merge(arr, l, m, r);
54     }
55 }
56
57 void printArray(int A[], int size)
58 {
59     int i;
60     for (i = 0; i < size; i++)
61         printf("%d ", A[i]);
62     printf("\n");
63 }
64
65 int main()
66 {
67     int arr[] = { 12, 11, 13, 5, 6, 7 };
68     int arr_size = sizeof(arr) / sizeof(arr[0]);
69
70     printf("Given array is \n");
71     printArray(arr, arr_size);
72
73     mergeSort(arr, 0, arr_size - 1);
74
75     printf("\nSorted array is \n");
76     printArray(arr, arr_size);
77     return 0;
78 }
79
80
81
82
83
```

OUTPUT:

```
adminl4@adminl4-HP-ProDesk-400-G7-Microtower-PC:~$ gedit exp8.c
adminl4@adminl4-HP-ProDesk-400-G7-Microtower-PC:~$ gcc exp8.c
adminl4@adminl4-HP-ProDesk-400-G7-Microtower-PC:~$ ./a.out
Given array is
12 11 13 5 6 7

Sorted array is
5 6 7 11 12 13
adminl4@adminl4-HP-ProDesk-400-G7-Microtower-PC:~$
```

SUBMITTED BY:SHAMAL BHANUDAS DEORE

CLASS/DIV:SY-IT-A

ROLL NO.18