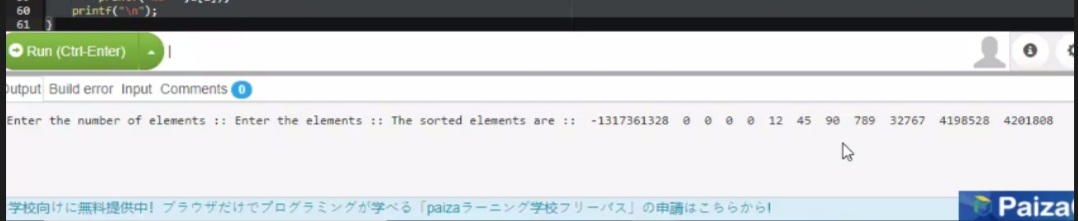
**Experiment -10**

**Aim: Write a program  for Merge sort.**

1. #include<stdio.h>
2. #include<stdlib.h>
3. void Merge(int a[], int tmp[], int lpos, int rpos, int rend)
4. {
5. int i, lend, n, tmppos;
6. lend = rpos - 1;
7. tmppos = lpos;
8. n = rend - lpos + 1;
9. while(lpos <= lend && rpos <= rend)
10. {
11. if(a[lpos] <= a[rpos])
    1. tmp[tmppos++] = a[lpos++];
12. else
    1. tmp[tmppos++] = a[rpos++];
13. }
14. while(lpos <= lend)
15. tmp[tmppos++] = a[lpos++];
16. while(rpos <= rend)
17. tmp[tmppos++] = a[rpos++];
18. for(i = 0; i < n; i++, rend--)
19. a[rend] = tmp[rend];
20. }
21. void MSort(int a[], int tmp[], int left, int right)
22. {
23. int center;
24. if(left < right)
25. {
26. center = (left + right) / 2;
27. MSort(a, tmp, left, center);
28. MSort(a, tmp, center + 1, right);
29. Merge(a, tmp, left, center + 1, right);
30. }
31. }
32. void MergeSort(int a[], int n)
33. {
34. int \*tmparray;
35. tmparray = malloc(sizeof(int) \* n);
36. MSort(a, tmparray, 0, n-1);
37. free(tmparray);
38. }
39. main()
40. {
41. int i, n, a[10];
42. printf("Enter the number of elements :: ");
43. scanf("%d",&n);
44. printf("Enter the elements :: ");
45. for(i = 0; i < n; i++)
46. {
47. scanf("%d",&a[i]);
48. }
49. MergeSort(a,n);
50. printf("The sorted elements are ::  ");
51. for(i = 0; i < n; i++)
52. printf("%d  ",a[i]);
53. printf("\n");
54. }

**Input-**

**Output -**