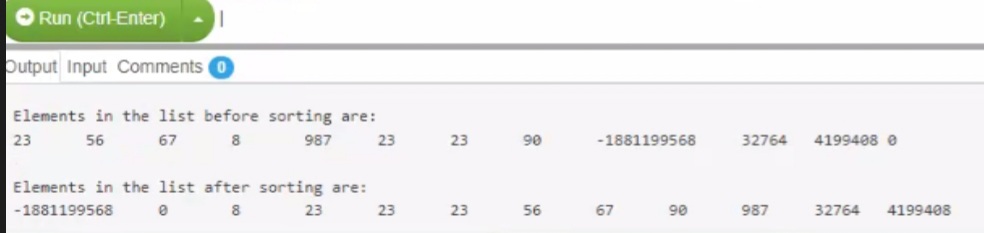
**Experiment- 7**

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

1. #include <stdio.h>
2. #define MAX 10
3. void swap(int \*m,int \*n)
4. {
5. int temp;
6. temp = \*m;
7. \*m = \*n;
8. \*n = temp;
9. }
10. int get\_key\_position(int x,int y )
11. {
12. return((x+y) /2);
13. }
14. // Function for Quick Sort
15. void quicksort(int list[],int m,int n)
16. {
17. int key,i,j,k;
18. if( m < n)
19. {
20. k = get\_key\_position(m,n);
21. swap(&list[m],&list[k]);
22. key = list[m];
23. i = m+1;
24. j = n;
25. while(i <= j)
26. {
27. while((i <= n) && (list[i] <= key))
    1. i++;
28. while((j >= m) && (list[j] > key))
    * 1. j--;
    1. if( i < j)
       1. swap(&list[i],&list[j]);
29. }
30. swap(&list[m],&list[j]);
31. quicksort(list,m,j-1);
32. quicksort(list,j+1,n);
33. }
34. }
35. // Function to read the data
36. void read\_data(int list[],int n)
37. {
38. int j;
39. printf("\n\nEnter the elements:\n");
40. for(j=0;j<n;j++)
41. scanf("%d",&list[j]);
42. }
43. // Function to print the data
44. void print\_data(int list[],int n)
45. {
46. int j;
47. for(j=0;j<n;j++)
48. printf("%d\t",list[j]);
49. }
50. int main()
51. {
52. int list[MAX], num;
53. //clrscr();
54. printf("\n\*\*\*\*\* Enter the number of elements Maximum [10] \*\*\*\*\*\n");
55. scanf("%d",&num);
56. read\_data(list,num);
57. printf("\n\nElements in the list before sorting are:\n");
58. print\_data(list,num);
59. quicksort(list,0,num-1);
60. printf("\n\nElements in the list after sorting are:\n");
61. print\_data(list,num);
62. return 0;

**INPUT-**

****

**Output-**