```
#include<stdio.h>
 1
 2
    #include<stdlib.h>
 3
 4
    int Partition(int array[], int p, int r)
 5
 6
        int pivot = array[r];
 7
        int i = p - 1;
 8
        for(int j = p; j < r; j++)
 9
10
             if(array[j] <= pivot)</pre>
11
12
                 i++;
13
                 swap(&array[i], &array[j]);
14
15
16
        swap(&array[i+1], &array[r]);
17
        return (i+1);
18
19
20
    int Randomized_Partition(int array[], int p, int r)
21
22
        srand(time(NULL));
23
        int random = p + rand() % (r - p);
24
        swap(&array[random], &array[r]);
25
        return Partition(array, p, r);
26
27
28
    void QuickSort(int array[], int p, int r)
29
30
        if(p < r)
31
32
             int q = Randomized_Partition(array, p, r);
33
            QuickSort(array, p, q-1);
34
            QuickSort(array, q+1, r);
35
36
37
38
    void swap(int *a, int *b)
39
40
        int temp = *a;
        *a = *b;
41
42
        *b = temp;
43
44
45
    void display(int array[], int length)
46
47
        int i;
48
        for(i = 0; i < length; i++)
49
50
            printf("%d ", array[i]);
51
52
53
54
55
    int main()
56
57
        int length, i;
58
        printf("##### RANDOMIZED QUICKSORT ALGORITHM TESTING #####\n");
        printf("\n=> Enter array size to create an array = ");
59
        scanf("%d", &length);
60
61
        int array[length];
        printf("\n=> Enter %d array element:\n", length);
62
63
64
        for(i = 0; i < length; i++)
65
66
            scanf("%d", &array[i]);
```

```
67    }
68
69    printf("\n\n=> Before sort array elements are: ");
70
71    display(array, length);
72
73    QuickSort(array, 0, length-1);
74
75    printf("\n\n=> After sort array elements are : ");
76
77    display(array, length);
78
79    printf("\n\n");
80
81    return 0;
```