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

int Partition(int array[], int p, int r)
{
    int pivot = array[r];
    int i = p - 1;
    for(int j = p; j < r; j++)
    {
        if(array[j] <= pivot)
        {
            i++;
            swap(&array[i], &array[j]);
        }
    }
    swap(&array[i+1], &array[r]);
    return (i+1);
}

int Randomized_Partition(int array[], int p, int r)
{
    srand(time(NULL));
    int random = p + rand() % (r - p);
    swap(&array[random], &array[r]);
    return Partition(array, p, r);
}

void QuickSort(int array[], int p, int r)
{
    if(p < r)
    {
        int q = Randomized_Partition(array, p, r);
        QuickSort(array, p, q-1);
        QuickSort(array, q+1, r);
    }
}

void swap(int *a, int *b)
{
    int temp = *a;
    *a = *b;
    *b = temp;
}

void display(int array[], int length)
{
    int i;
    for(i = 0; i < length; i++)
    {
        printf("%d ", array[i]);
    }

}

int main()
{
    int length, i;
    printf("##### RANDOMIZED QUICKSORT ALGORITHM TESTING #####\n");
    printf("\n=> Enter array size to create an array = ");
    scanf("%d", &length);
    int array[length];
    printf("\n=> Enter %d array element:\n", length);

    for(i = 0; i < length; i++)
    {
        scanf("%d", &array[i]);
    }

    printf("\n\n=> Before sort array elements are: ");

    display(array, length);

    QuickSort(array, 0, length-1);

    printf("\n\n=> After sort array elements are : ");

    display(array, length);

    printf("\n\n");

    return 0;
}
```