**<http://cse.iitkgp.ac.in/~abhij/course/theory/Algo1/Autumn13/lab/A7.html>**

**Inline Counting Sort**

Implement the counting sort algorithm taught in the class with two variations:

1. The entries in the input array *A* are now allowed to be integer values in the range −*k* to +*k*.
2. You are not allowed to use the output array *B*. Use the count array *C*, and in the final pass through the array *A*, modify *A* itself so that *A* is sorted inline. This version of counting sort need not be stable.

Notice that once you prepare the count array, you may think about printing the integers −*k*, −*k* + 1, −*k* + 2, ..., *k* − 1, *k* in that sequence the required numbers of times. In practice, however, you sort a record of items with respect to some key value. Therefore, this flat listing of only the key values is not treated as sorting *A*. Use suitable swap operations to effect inline sorting. The running time of the inline sorting should continue to remain O(*n* + *k*).

Write a program to do the following.

* Write a function that takes as input the array *A*, its size *n*, and the limit <i<k< i="">. The function sorts *A* according to the inline counting sort algorithm.</i<k<>
* Write a main() function that reads *n* and *k* from the user. It then creates a random array *A* with *n* integer entries each in the range −k to +k. It calls the above function to sort *A*. The array *A* is printed before and after sorting.

**Sample Output**

n = 100

k = 10

+++ The array before sorting

4 0 -1 6 -3 3 -3 6 7 8 4 10 -5 2 0 -5

1 -9 1 2 9 -10 4 -10 1 -1 6 4 -3 -1 -8 1

0 10 -2 1 2 -10 4 5 4 -9 -4 -7 7 -6 5 1

-5 -4 9 5 3 -6 -9 0 -3 -9 6 10 1 -4 -9 -9

-5 9 -5 2 -3 4 5 2 8 -1 2 -3 2 10 2 -1

7 -7 10 -8 3 -3 0 -1 -2 9 -8 -5 6 -8 9 -1

5 4 3 -6

+++ The array after sorting

-10 -10 -10 -9 -9 -9 -9 -9 -9 -8 -8 -8 -8 -7 -7 -6

-6 -6 -5 -5 -5 -5 -5 -5 -4 -4 -4 -3 -3 -3 -3 -3

-3 -3 -2 -2 -1 -1 -1 -1 -1 -1 -1 0 0 0 0 0

1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 3

3 3 3 4 4 4 4 4 4 4 4 5 5 5 5 5

6 6 6 6 6 7 7 7 8 8 9 9 9 9 9 10

10 10 10 10