	1	2	3	4	5	6	7	8	9	10	11
A =	6	0	2	0	1	3	4	6	1	3	2

for i=1 to A.length
 ++count[A[i]]

2	2	2	2	1	0	2

For i =1 to C.length.
 count[i] = count [i] + count [i -1]

	0	1	2	3	4	5	6
C =	2	4	6	8	9	9	11

At index 1 we will store the sum of count at index 0 and 1 at index 2 we will store the sum of count at new index 1 and index 2 and so on.

4) Take each element of the array, see its count value (element = index) in the table and place the element at that place. Update the count of each index by count -1, as wee come accross an element with the value of that index

A[1] = 6 → index - C[6] = 11 , count at index 6 becomes 10
Therefore

_ _ _ _ _ _ _ _ _ _ 6

placing 0 at position 2 , count index 0 becomes 1.

_ 0 _ _ _ _ _ _ _ _ 6

placing 2 at position 6, count at index 2 becomes 5

_ 0 _ _ _ 2 _ _ _ _ 6

placing 0 at position 1, count at index 0 becomes 0

0 0 _ _ _ 2 _ _ _ _ 6

placing 1 at position 4, count at index 1 becom 3

0 0 _ 1 _ 2 _ _ _ _ 6

Solu

placing 3 at position 8, count at index 3 becomes 7

0 0 _ 1 _ 2 _ 3 _ _ 6

placing 4 at position 9, count at index 4 becom

0 0 _ 1 _ 2 _ 3 4 _ 6

placing 6 at position 10, count at index 6 beco

0 0 _ 1 _ 2 _ 3 4 6 6

placing 1 at position 3, count at index 1 becomes 2

0 0 1 1 _ 2 _ 3 4 6 6

Placing 3 at position 7, count at index 3 becomes 6

0 0 1 1 _ 2 3 3 4 6 6

Placing 2 at position 5, count at index 2 becomes 4

0	0	1	1	2	2	3	3	4	6	6

Thus we have obtained a sorted array, in $O(n+k)$ times and space, where n is the number of elements and k is the range of elements.