

Laboratory work #5

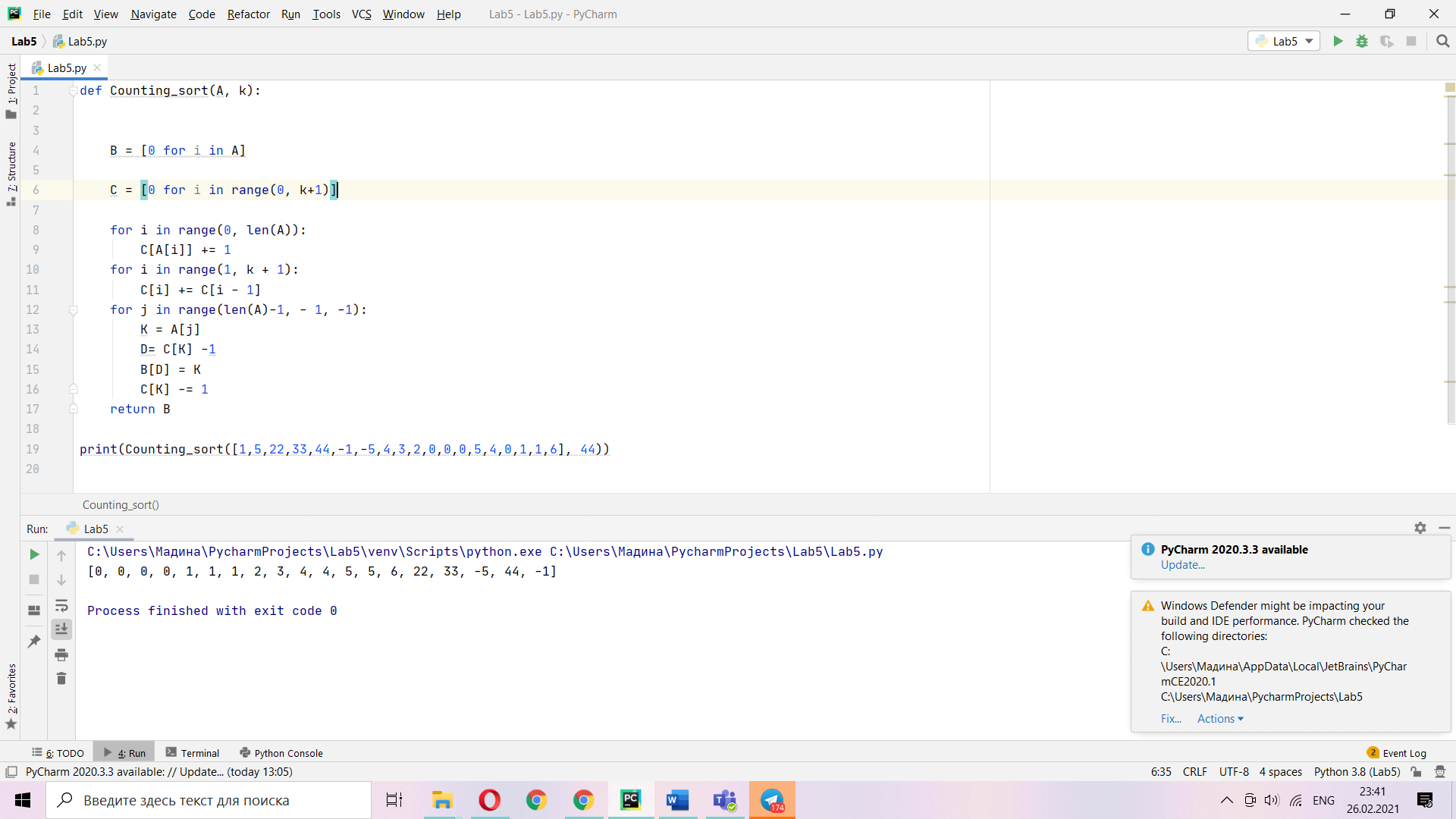
Performance, Data Structure & Algorithms

“Counting Sort”

Group:ITDS-1901

Ibrayeva Madina

1. Implement Counting sort using any programming languages.



def c(A,k):

B=[i]

C= [i]

for i in range(0, n):

C[A[i]]+=1

for i in range(1, k+1):

C[i]+=C[i-1]

for j in range(n-1,-1,-1):

K=A[j]

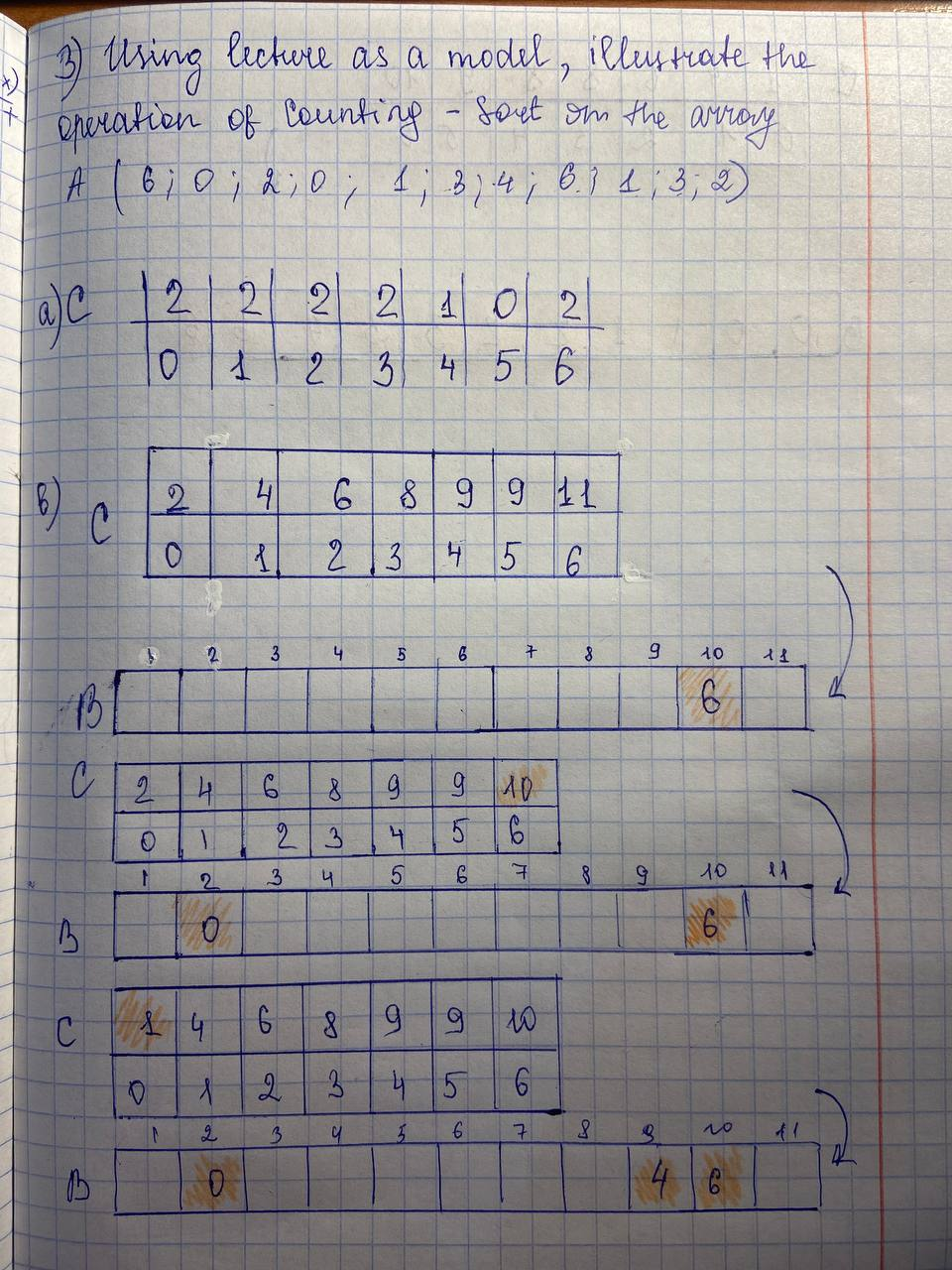
D=C[k] -1

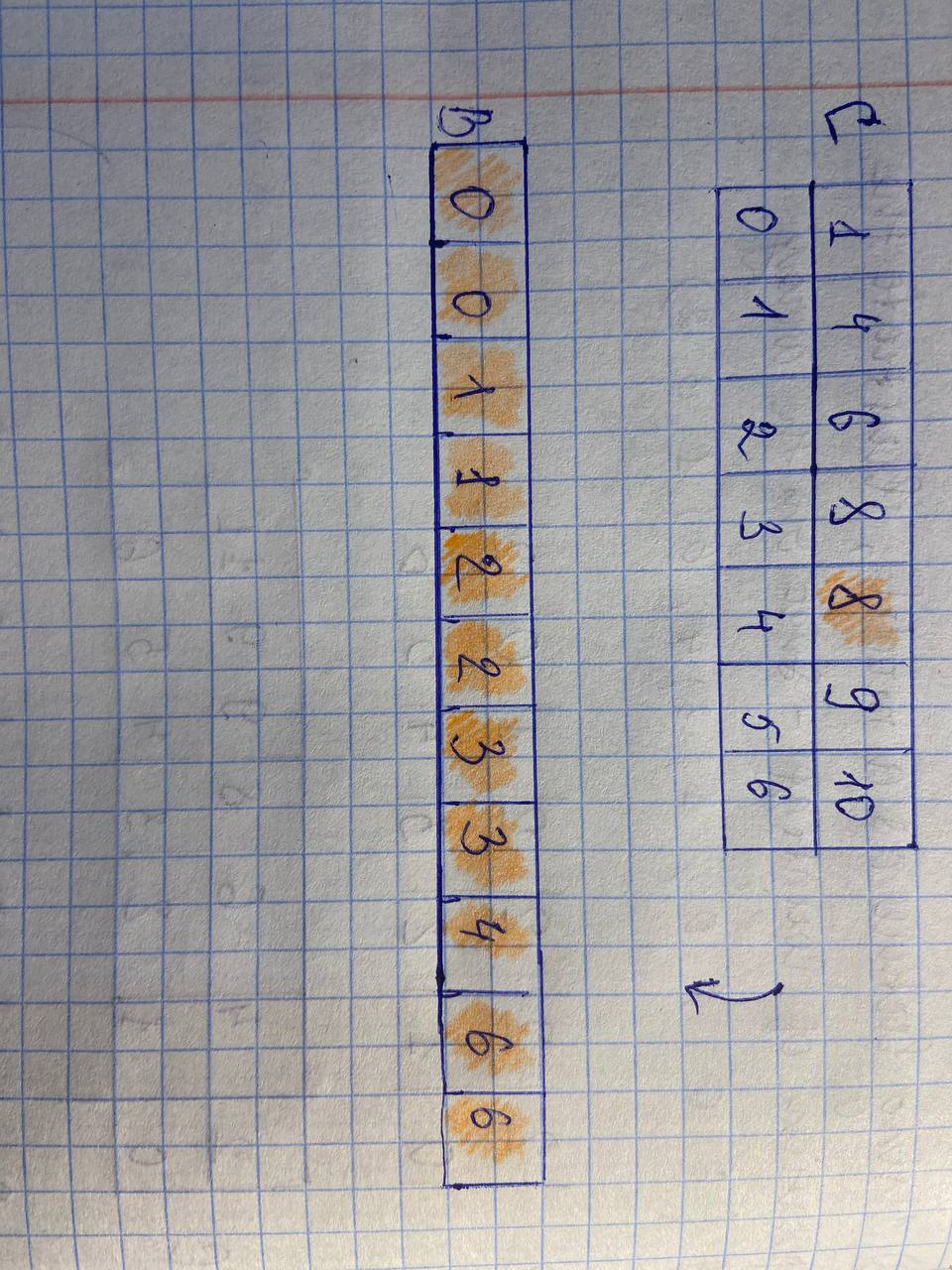
B[D]=K

C[K]-=1

Prove that COUNTING-SORT is stable.

Counting sort is stable because of the time complexity and order that works with the same elements. Example: You have list of three ‘0’, and 2 ‘1’, and 5 ‘7’, stable sort sorting same key elements to output list stable sort put as order of input. Counting-Sort are working like this, so counting sort is stable sort.

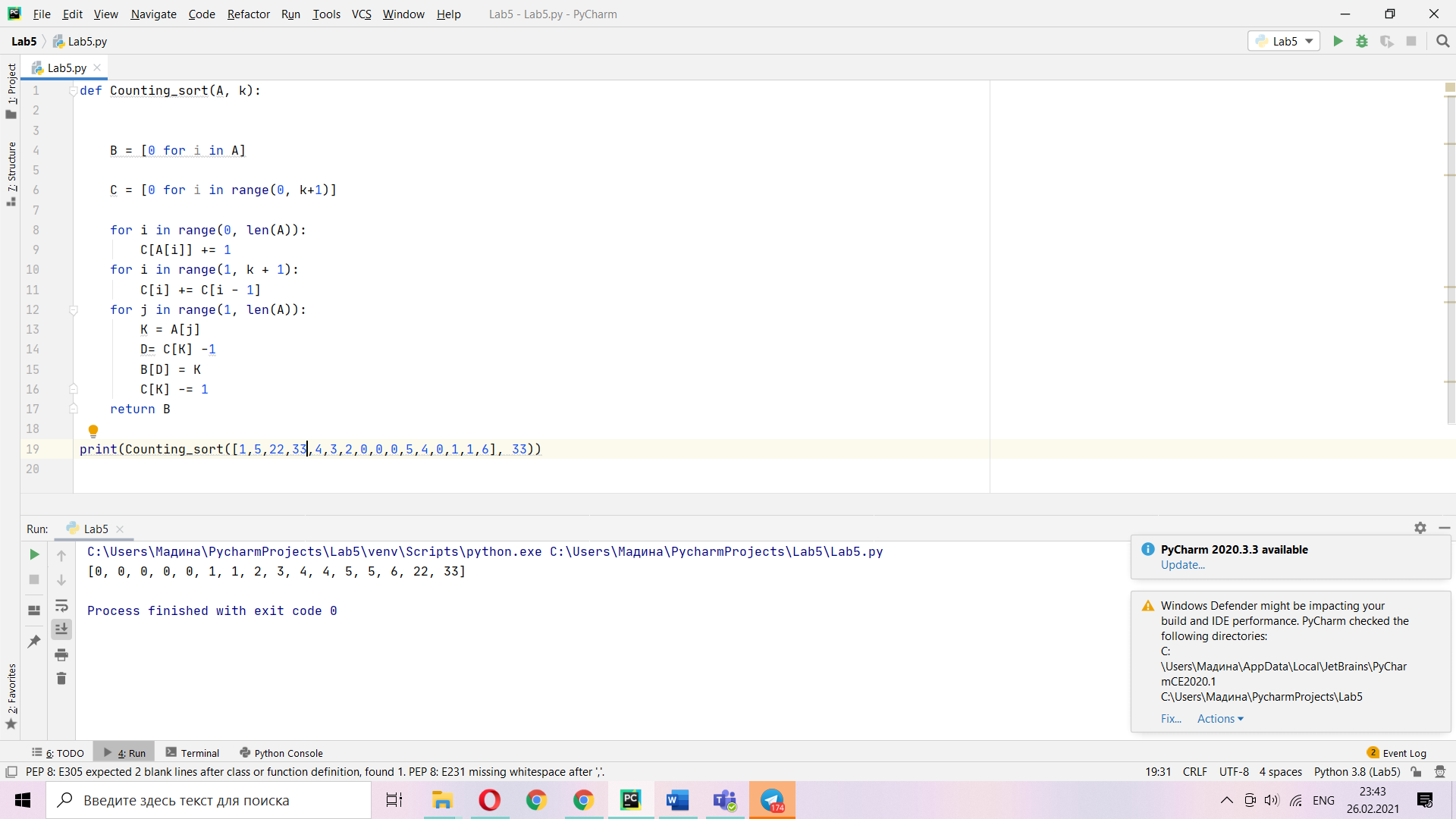
Using lecture as a model, illustrate the operation of COUNTING-SORT on the array A (6; 0; 2; 0; 1; 3; 4; 6; 1; 3; 2). 



Suppose that we were to rewrite the last for loop header in the COUNTINGSORT as

for j =1 to A.length

Show that the algorithm still works properly. Is the modified algorithm stable?



Algorithm still works properly, how we see on the screen the zeroes, 1’s, 4’s and 5’s are sorted in order. Sorting comparing element with elements that is less than this element not that is equal with him. Unstable code is when among integers there is element that placing word in them. Of course, we can do counting sort among elements containing words, but it will be unstable if there will be an integer and stable with only words element.