

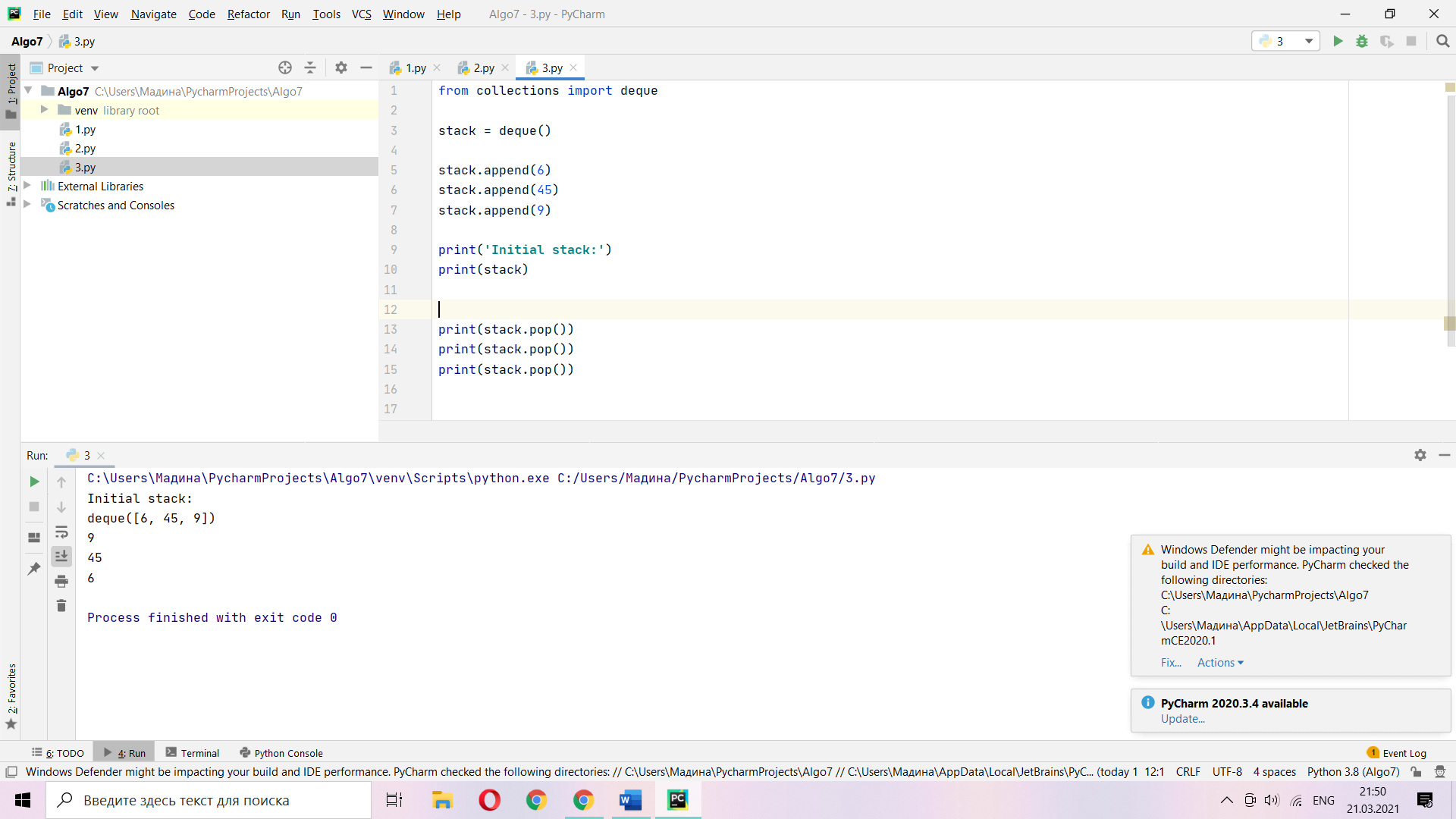
Laboratory work #7

Performance, Data Structure & Algorithms

Group:ITDS-1901

Ibrayeva Madina

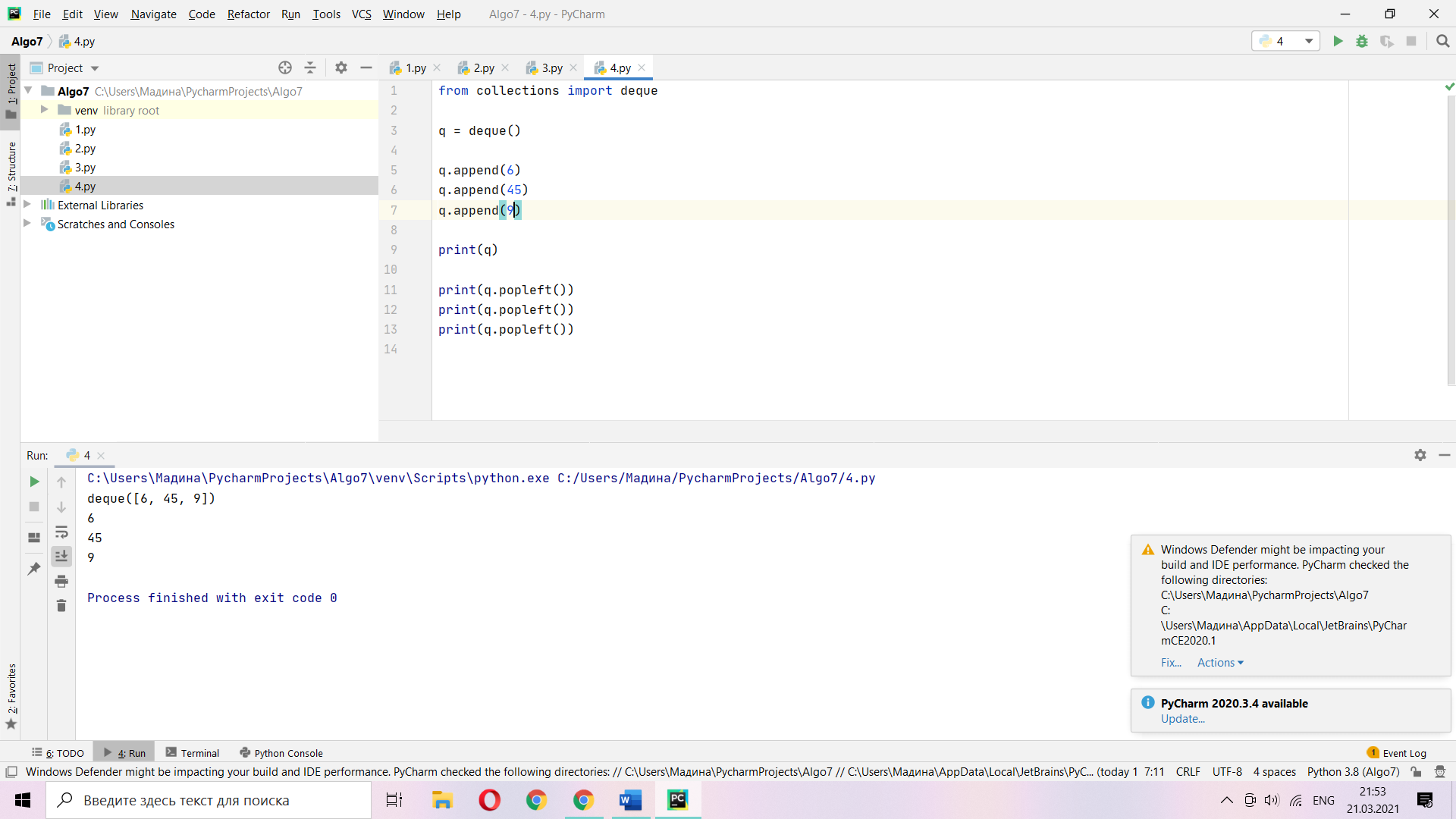
Implement Stack and Queue data structures using OOP.



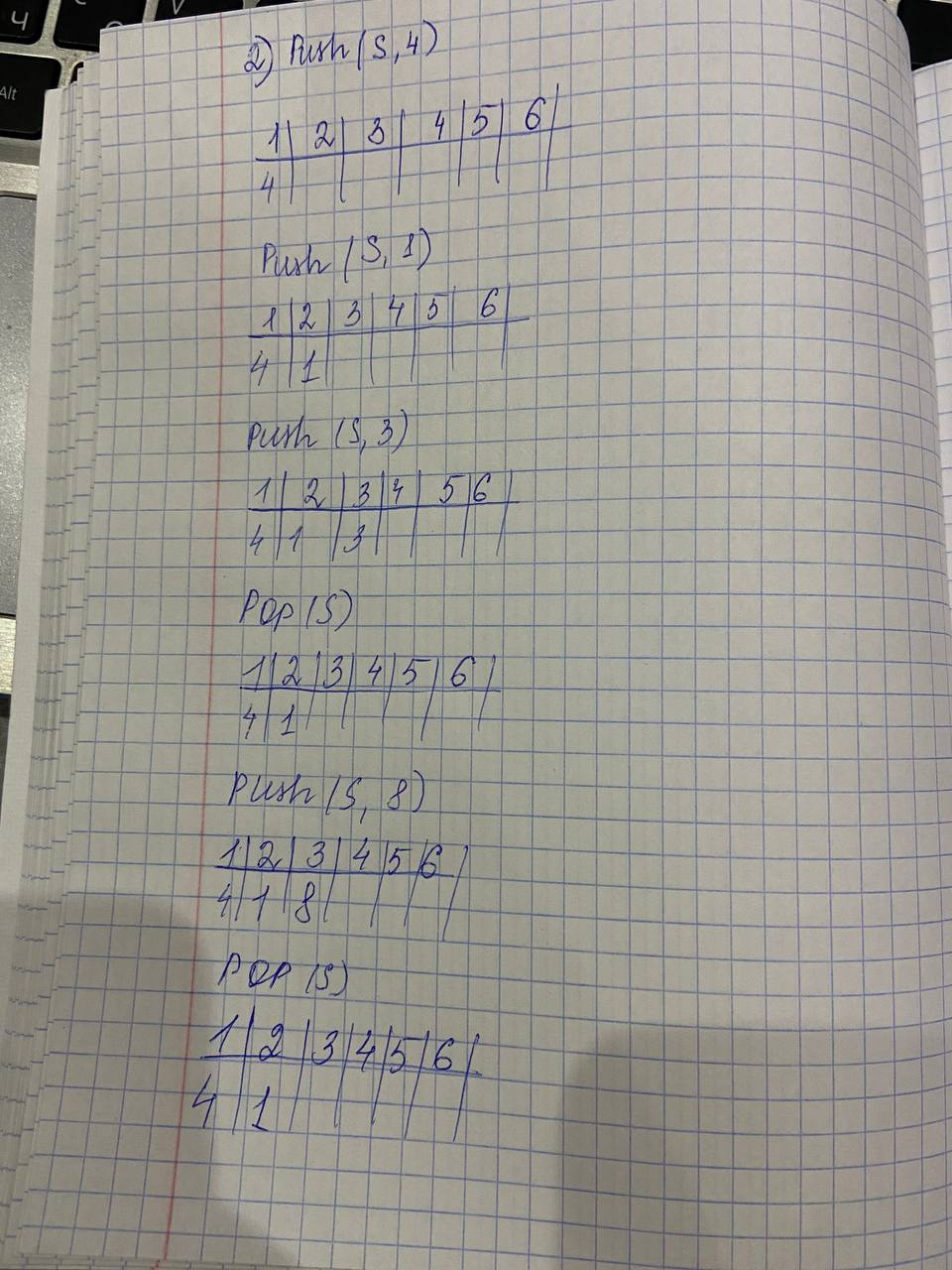
from collections import deque  
  
stack = deque()  
  
stack.append(6)

print(stack)  
  
print(stack.pop())

from collections import deque  
  
q = deque()  
  
q.append(6)  
  
  
print(q)  
  
print(q.popleft())



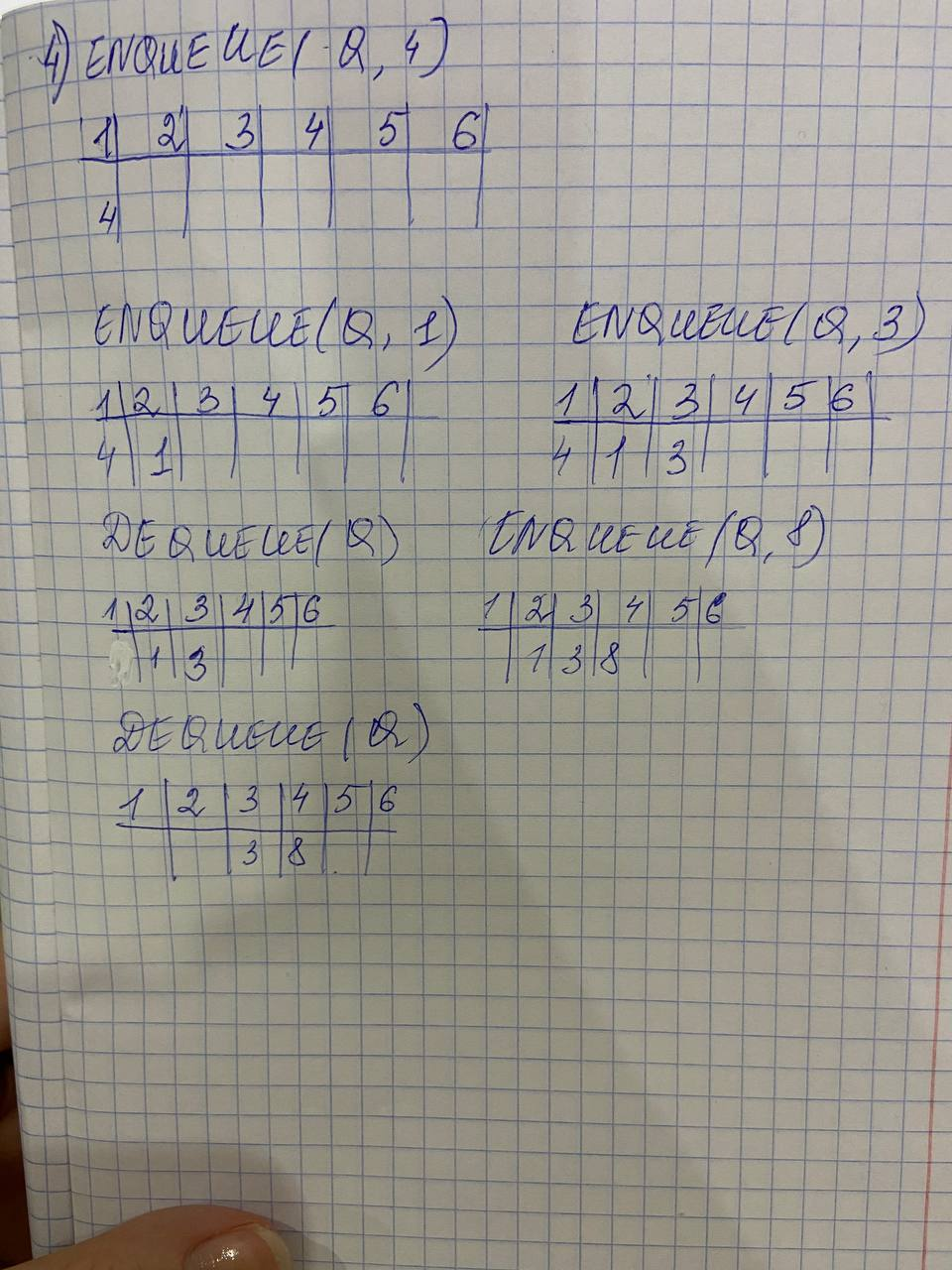
1. Using lecture as a model, illustrate the result of each operation in the sequence PUSH(S, 4), PUSH(S, 1), PUSH(S, 3), POP(S ), PUSH(S, 8), and POP(S) on an initially empty stack S stored in array S(1 … 6).



1. Explain how to implement two stacks in one array A(1…n) in such a way that neither stack overflows unless the total number of elements in both stacks together is n. The PUSH and POP operations should run in O(1) time.

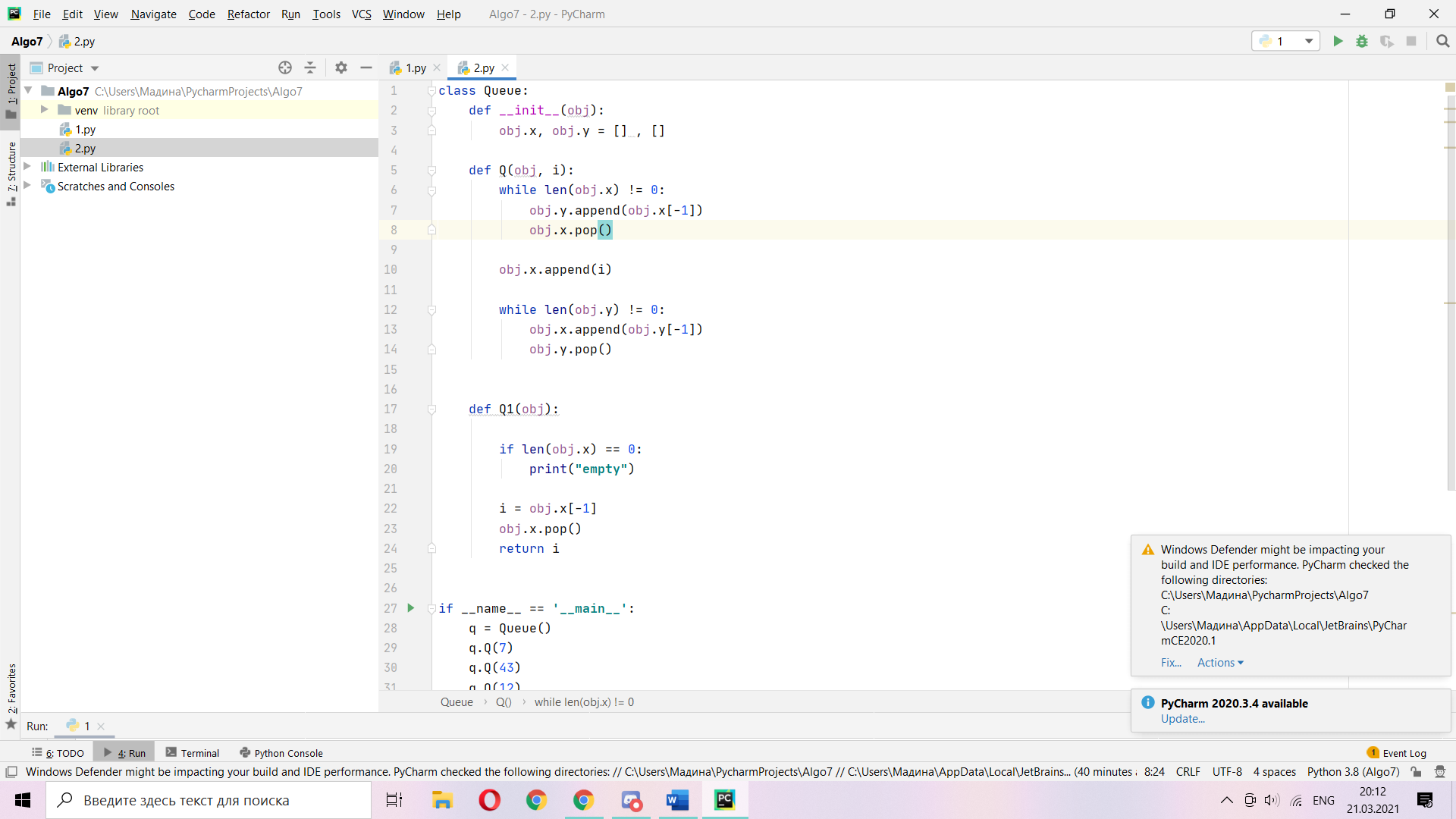
Implementing two stacks in one array, first stack will be pushing elements from 1 to n, second stack will be pushing from n to . So, for this procedure we can give only a half array to the first and second stacks but if they cross this half array it will be stack overflow, so it must be value n in this array.

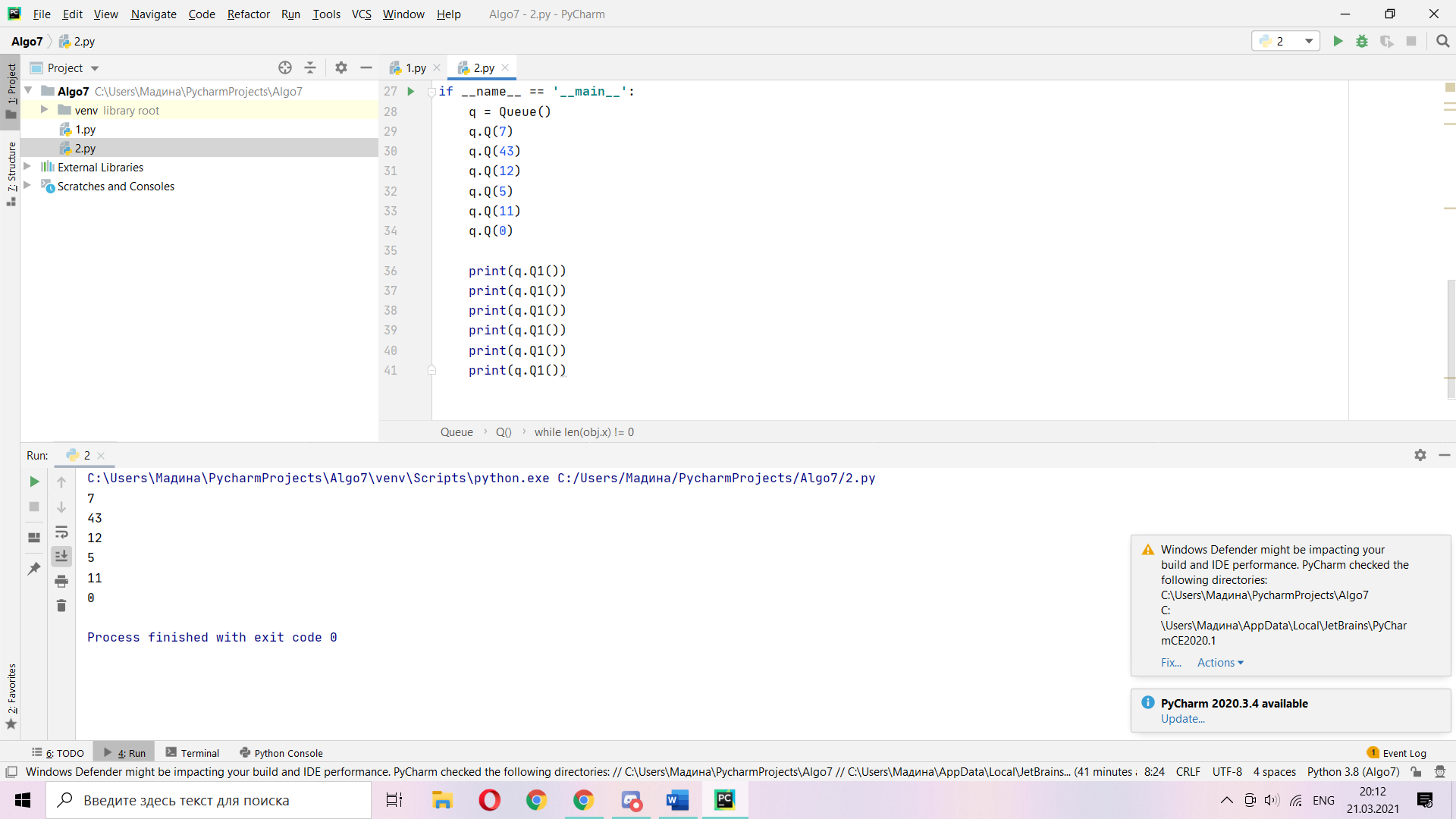
1. Using lecture as a model, illustrate the result of each operation in the sequence ENQUEUE(Q,4), ENQUEUE(Q, 1), ENQUEUE(Q, 3), DEQUEUE(Q), ENQUEUE(Q, 8), and DEQUEUE(Q) on an initially empty queue Q stored in array Q(1…6)



1. Show how to implement a queue using two stacks. Analyze the running time of the queue operations.

Enqueue takes O(1) time, but Dequeue takes O(n) time





1. Show how to implement a stack using two queues. Analyze the running time of the stack operations.

Running time will be O(n) because of push operation that takes O(1) and pop operation that takes O(n), n is the number of elements in array/lists

