Министерство цифрового развития, связи

и массовых коммуникаций Российской Федерации

Ордена Трудового Красного Знамени

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский технический университет связи и информатики»

Лабораторная работа №4

по дисциплине «Структуры и алгоритмы обработки данных»

“Реализация стека и дека”

Выполнила:

студентка группы БСТ2004

Моисеева С.Д.

Проверил:

А.Д. Чайка

Москва 2022

**Цель работы**: изучение основных алгоритмов стеков и деков и их реализация на Python

**Задание:** Создать проект на python (среда PyCharm), включающий реализацию следующих заданий с использованием стека и дека.

1. Реализовать стек и дек
2. Отсортировать строки файла, содержащие названия книг, в алфавитном порядке с использованием двух деков.
3. Дек содержит последовательность символов для шифровки сообщений. Дан текстовый файл, содержащий зашифрованное сообщение. Пользуясь деком, расшифровать текст. Известно, что при шифровке каждый символ сообщения заменялся следующим за ним в деке по часовой стрелке через один.
4. Даны три стержня и n дисков различного размера. Диски можно надевать на стержни, образуя из них башни. Перенести n дисков со стержня А на стержень С, сохранив их первоначальный порядок.
5. Дан текстовый файл с программой на алгоритмическом языке. За один просмотр файла проверить баланс круглых скобок в тексте, используя стек.
6. Дан текстовый файл с программой на алгоритмическом языке. За один просмотр файла проверить баланс квадратных скобок в тексте, используя дек.
7. Дан файл из символов. Используя стек, за один просмотр файла напечатать сначала все цифры, затем все буквы, и, наконец, все остальные символы, сохраняя исходный порядок в каждой группе символов.
8. Дан файл из целых чисел. Используя дек, за один просмотр файла напечатать сначала все отрицательные числа, затем все положительные числа, сохраняя исходный порядок в каждой группе.
9. Дан текстовый файл. Используя стек, сформировать новый текстовый файл, содержащий строки исходного файла, записанные в обратном порядке: первая строка становится последней, вторая – предпоследней и т.д.

**Ход работы.**

**Задание 1**

Отсортировать строки файла, содержащие названия книг, в алфавитном порядке с использованием двух деков.

from dequeclass import \*  
  
d1 = Deque()  
d2 = Deque()  
file = open("books.txt", "r", encoding="utf-8")  
lines = file.readlines()  
print(lines)  
d1.addLast(lines[0])  
for line in lines[1::]:  
 if line > d1.items[-1]:  
 d1.addLast(line)  
 elif line < d1.items[0]:  
 d1.addFirst(line)  
 else:  
 tmpsize = d1.size()  
 for i in range(tmpsize):  
 d2.addLast(d1.items[-1])  
 d1.removeLast()  
 if line > d1.items[-1]:  
 d1.addLast(line)  
 d2.items.reverse()  
 d1.items += d2.items  
 d2.items = []  
 break  
print(d1.items)

Результат выполнения задания представлен на рисунке 1:

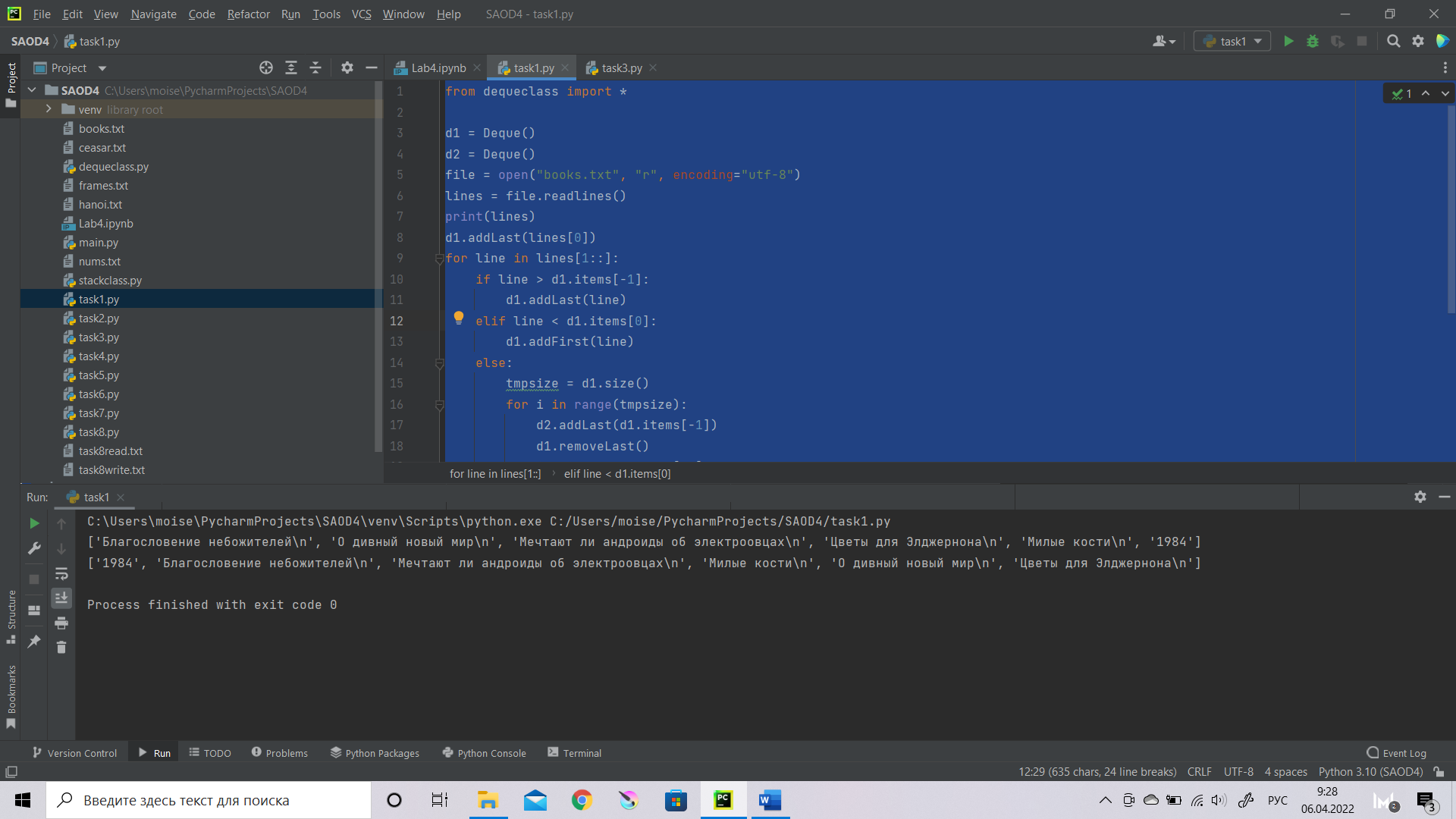


Рисунок 1 – Тестирование задания 1

**Задание 2**

Дек содержит последовательность символов для шифровки сообщений. Дан текстовый файл, содержащий зашифрованное сообщение. Пользуясь деком, расшифровать текст. Известно, что при шифровке каждый символ сообщения заменялся следующим за ним в деке по часовой стрелке через один.

from dequeclass import \*  
  
d1 = Deque()  
d1.items = ["а", "б", "в", "г", "д", "е", "ё", "ж", "з", "и", "й", "к", "л", "м", "н", "о",  
 "п", "р", "с", "т", "у", "ф", "х", "ц", "ч", "ш", "щ", "ъ", "ы", "ь", "э", "ю", "я"]  
file = open("ceasar.txt", "r", encoding="utf-8")  
line = file.read()  
dsize = d1.size()  
dots = '!,.? '  
result = ''  
for i in line:  
 if i.lower() in dots:  
 result += i  
 continue  
 tmp = i.lower()  
 if tmp == d1.items[-1]:  
 for z in range(2):  
 d1.addFirst(d1.items[-1])  
 d1.removeLast()  
 result += d1.items[-1]  
 elif tmp == d1.items[0]:  
 for z in range(2):  
 d1.addLast(d1.items[0])  
 d1.removeFirst()  
 result += d1.items[0]  
 else:  
 d1.addFirst(d1.items[-1])  
 d1.removeLast()  
 for j in range(dsize):  
 if tmp == d1.items[-1]:  
 for z in range(2):  
 d1.addFirst(d1.items[-1])  
 d1.removeLast()  
 result += d1.items[-1]  
 break  
 else:  
 d1.addFirst(d1.items[-1])  
 d1.removeLast()  
print(result)

Результат работы программы представлен на рисунке 2.

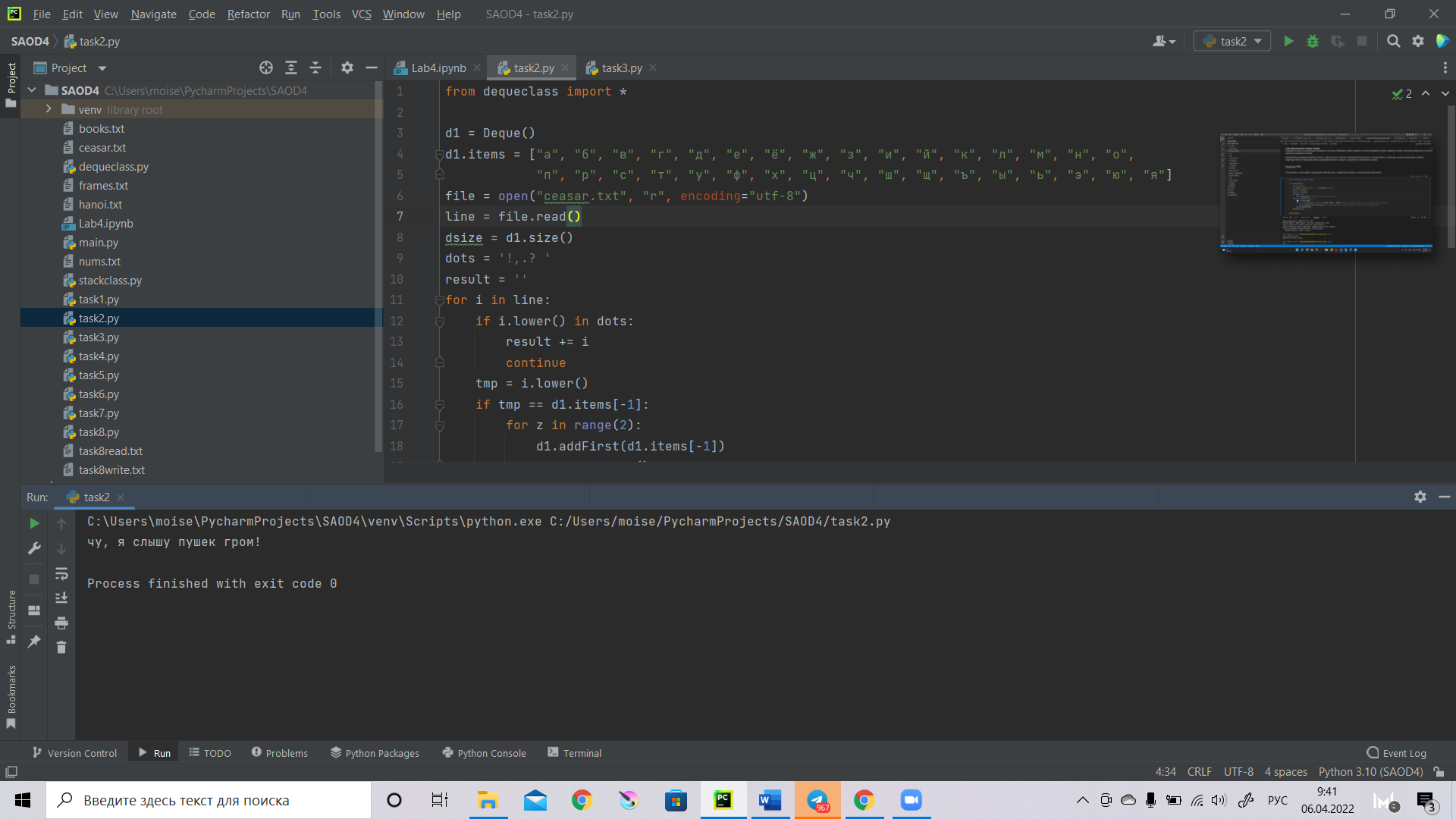


Рисунок 2 – Решение для строки «Щх, б унэъх схъжм етро!»

**Задание 3**

4. Даны три стержня и n дисков различного размера. Диски можно надевать на стержни, образуя из них башни. Перенести n дисков со стержня А на стержень С, сохранив их первоначальный порядок.

from dequeclass import \*  
  
  
def hanoi(n, source, helper, target):  
 if n > 0:  
 hanoi(n - 1, source, target, helper)  
 if not source[0].isEmpty():  
 disk = source[0].removeLast()  
 target[0].addLast(disk)  
 print("Перемещаем из", source[1], "в", target[1], disk)  
 print(source[1], source[0].items, helper[1], helper[0].items, target[1], target[0].items)  
 hanoi(n - 1, helper, source, target)  
  
  
source = (Deque(), "A")  
file = open("hanoi.txt", "r", encoding="utf-8")  
lines = [int(x) for x in file.read().split()]  
print(lines)  
source[0].items = lines # стержень А  
target = (Deque(), "B") # стержень С  
helper = (Deque(), "C") # стержень B  
hanoi(source[0].size(), source, helper, target)  
print("Результат:")  
print("A:", source[0].items, "B:", helper[0].items, "C:", target[0].items)

Результат работы программы представлен на рисунке 3.

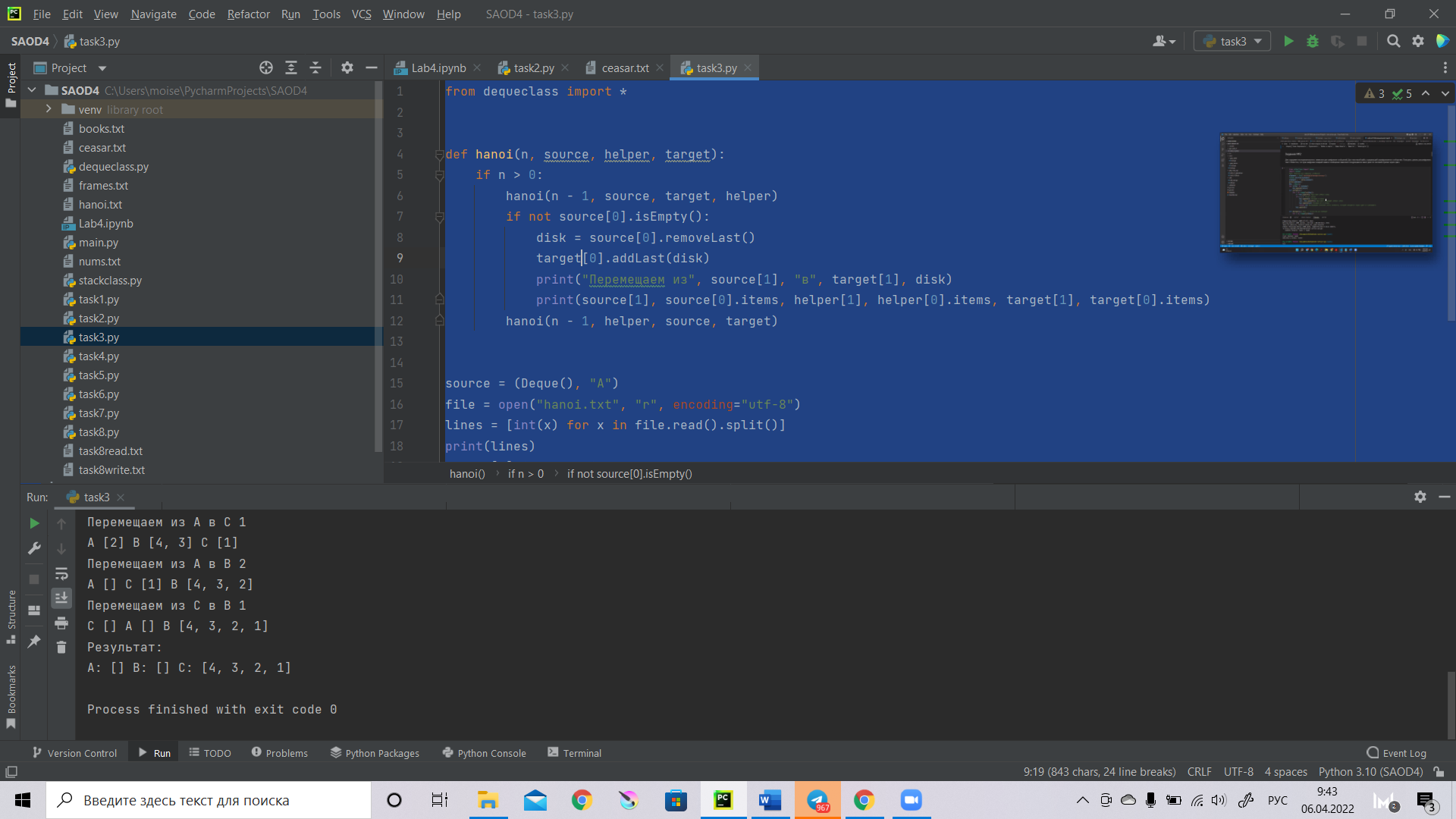


Рисунок 3 – Решение для комбинации 4 3 2 1

**Задание 4**

Дан текстовый файл с программой на алгоритмическом языке. За один просмотр файла проверить баланс круглых скобок в тексте, используя стек.

from stackclass import \*  
  
file = open("frames.txt", "r", encoding="utf-8")  
line = ''.join(file.readlines())  
st = Stack()  
for i in line:  
 if i in '()':  
 if st.isEmpty():  
 st.addLast(i)  
 elif st.items[-1] != i:  
 st.removeLast()  
 else:  
 st.addLast(i)  
if st.isEmpty():  
 print("Верно")  
else:  
 print("Неверно")

Результат работы программы представлен на рисунке 4.

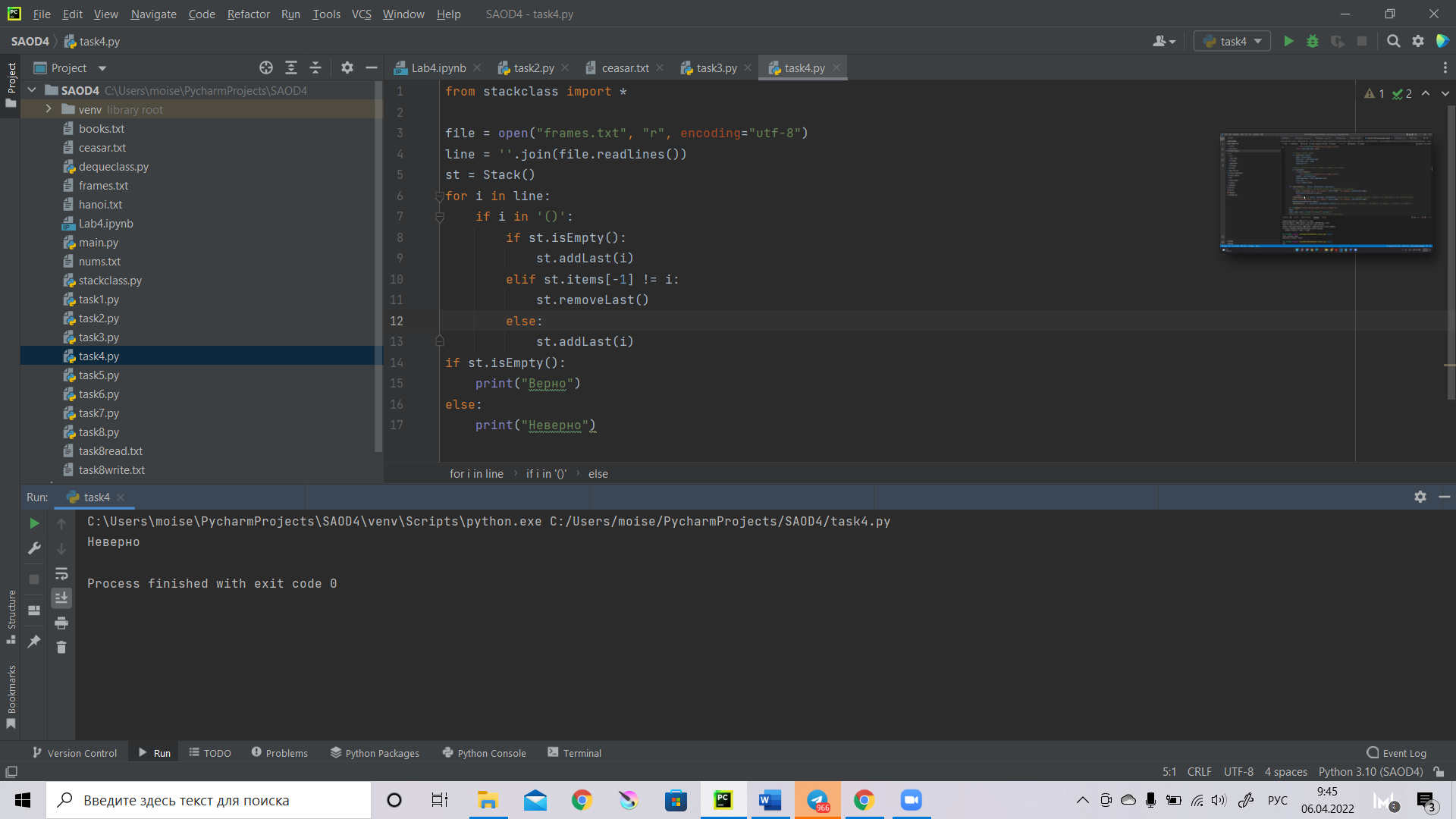


Рисунок 4 – Решение для комбинации ())))

**Задание 5**

Дан текстовый файл с программой на алгоритмическом языке. За один просмотр файла проверить баланс квадратных скобок в тексте, используя дек

from dequeclass import \*  
  
file = open("frames.txt", "r", encoding="utf-8")  
line = ''.join(file.readlines())  
d = Deque()  
for i in line:  
 if i in '[]':  
 d.addLast(i)  
 if d.items[-1] != d.items[0]:  
 d.removeLast()  
 d.removeFirst()  
 print(d.items)  
if d.isEmpty():  
 print("Верно")  
else:  
 print("Неверно")

Результат работы программы представлен на рисунке 5.

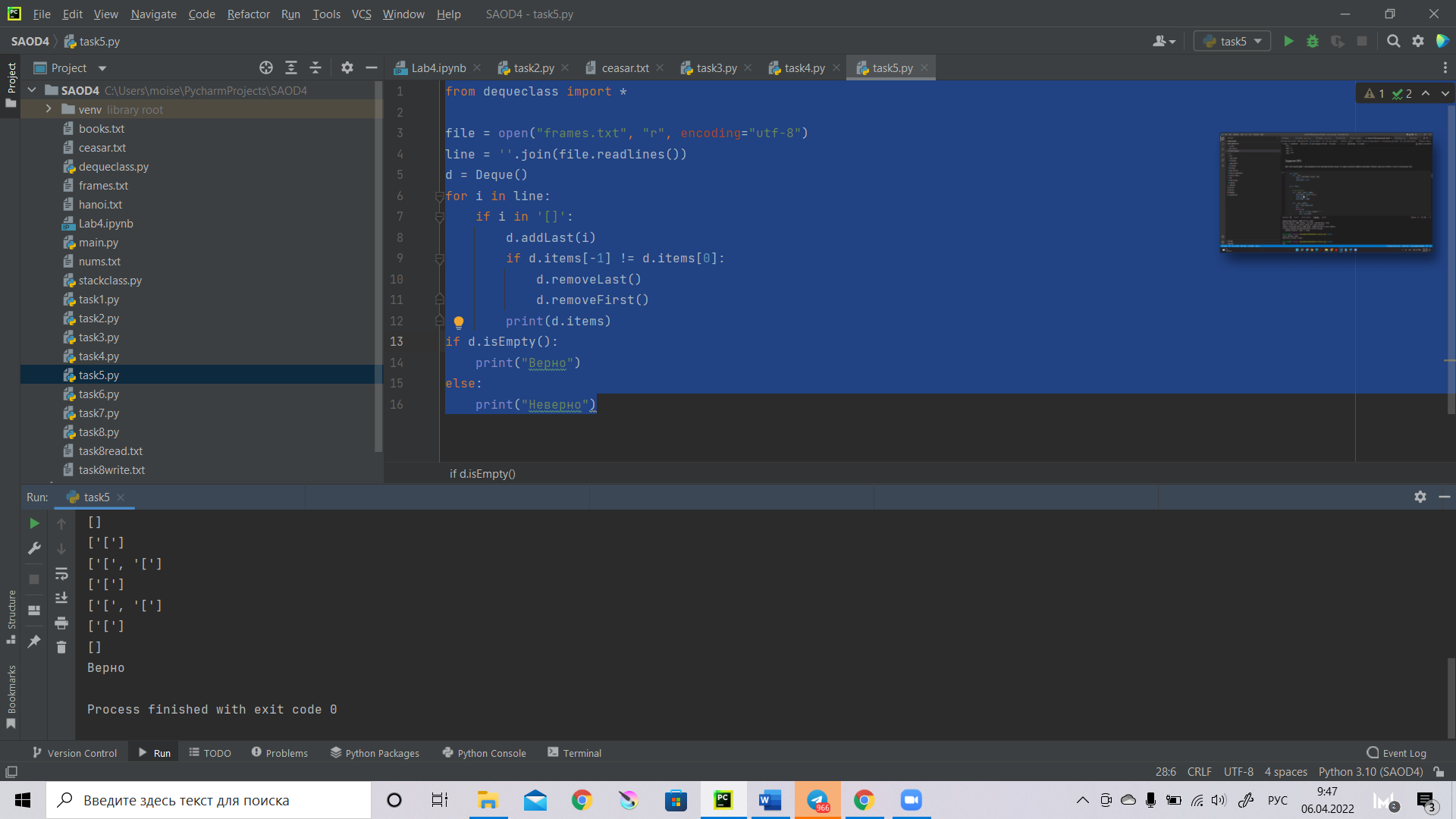


Рисунок 5 – Решение для комбинации [][][[][]]

**Задание 6**

Дан файл из символов. Используя стек, за один просмотр файла напечатать сначала все цифры, затем все буквы, и, наконец, все остальные символы, сохраняя исходный порядок в каждой группе символов.

from stackclass import \*  
  
file = open("frames.txt", "r", encoding="utf-8")  
line = ''.join(file.readlines())  
stalpha = Stack()  
stothers = Stack()  
for i in line:  
 if i.isnumeric():  
 print(i, end="")  
 elif i.isalpha():  
 stalpha.addLast(i)  
 else:  
 stothers.addLast(i)  
helper = Stack()  
print('')  
for i in range(stalpha.size()):  
 helper.addLast(stalpha.items[-1])  
 stalpha.removeLast()  
for i in range(helper.size()):  
 print(helper.items[-1], end="")  
 helper.removeLast()  
print('')  
for i in range(stothers.size()):  
 helper.addLast(stothers.items[-1])  
 stothers.removeLast()  
for i in range(helper.size()):  
 print(helper.items[-1], end="")  
 helper.removeLast()  
print('')

Результат работы программы представлен на рисунке 7.

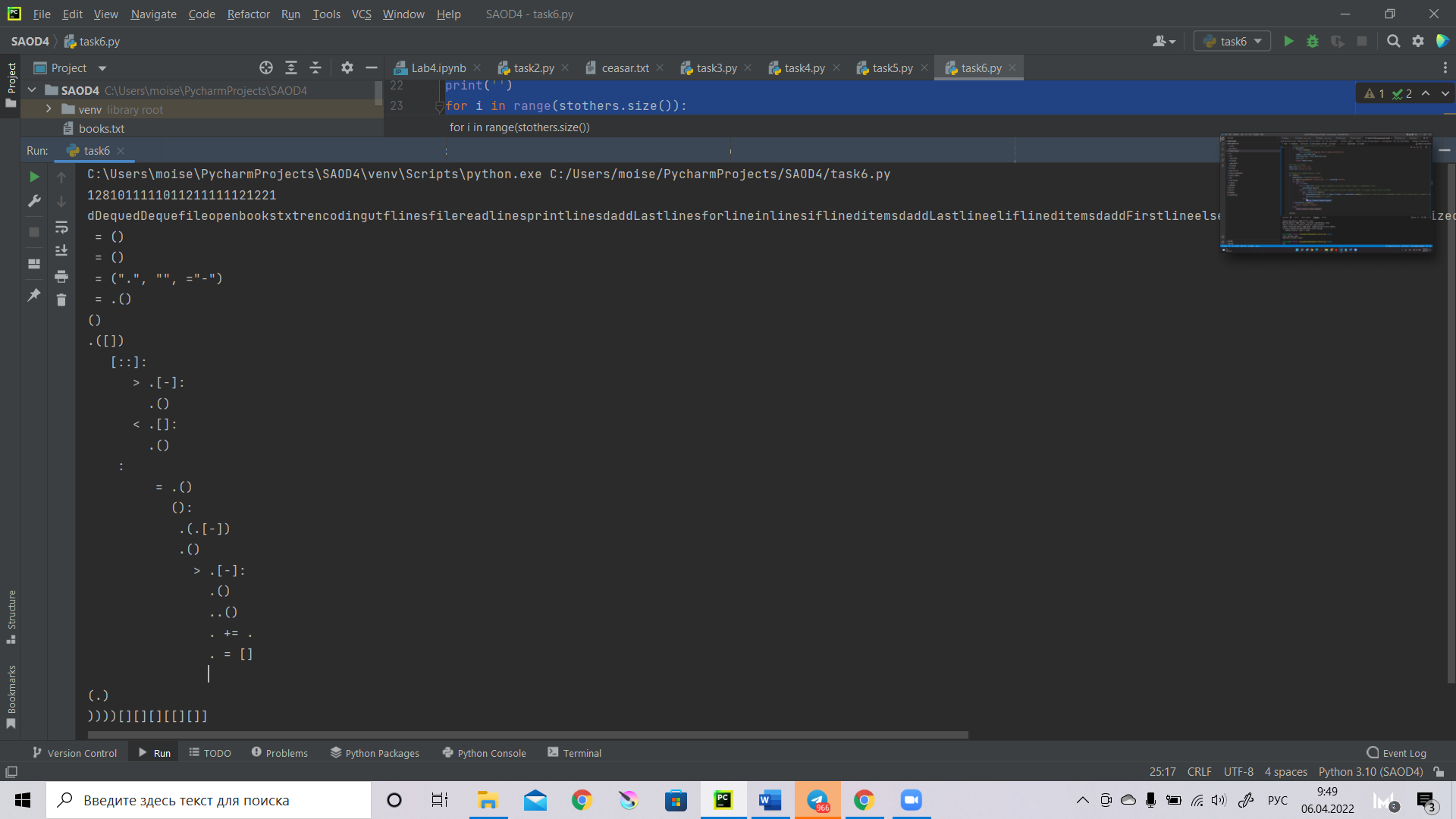


Рисунок 7 – Решение для задания 7

**Задание 7**

Дан файл из целых чисел. Используя дек, за один просмотр файла напечатать сначала все отрицательные числа, затем все положительные числа, сохраняя исходный порядок в каждой группе.

from dequeclass import \*  
  
d1 = Deque()  
file = open("nums.txt", "r", encoding="utf-8")  
line = ''.join(file.readlines()).split()  
print(line)  
for i in line:  
 if int(i) >= 0:  
 d1.addLast(i)  
 else:  
 d1.addFirst(i)  
print(d1.items)  
s = ''  
s2 = ''  
for i in range(d1.size()):  
 if int(d1.items[-1]) < 0:  
 s2 += d1.items[-1] + " "  
 else:  
 s = d1.items[-1] + " " + s  
 d1.removeLast()  
print(s)  
print(s2)

Результат работы программы представлен на рисунке 8.

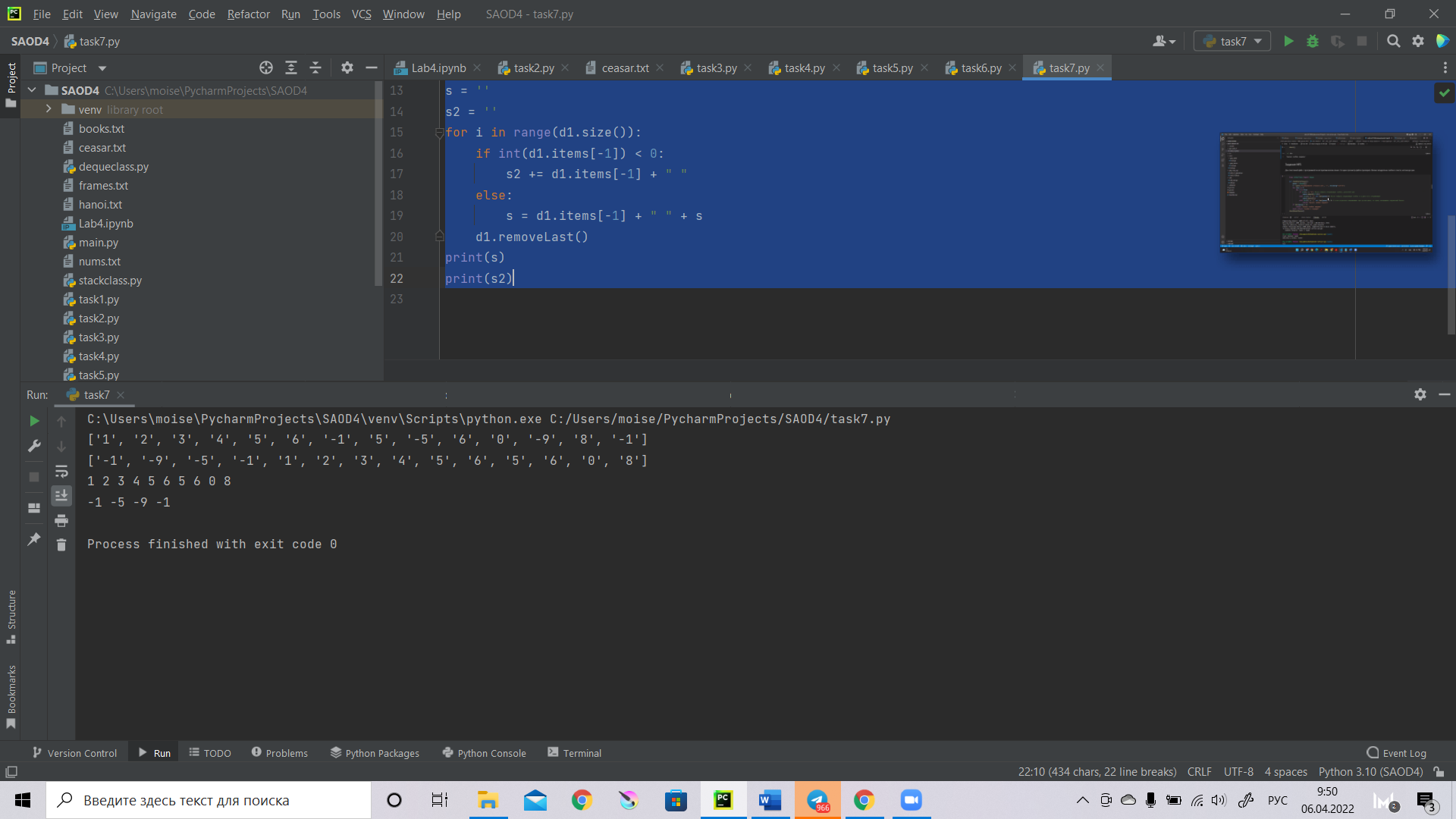


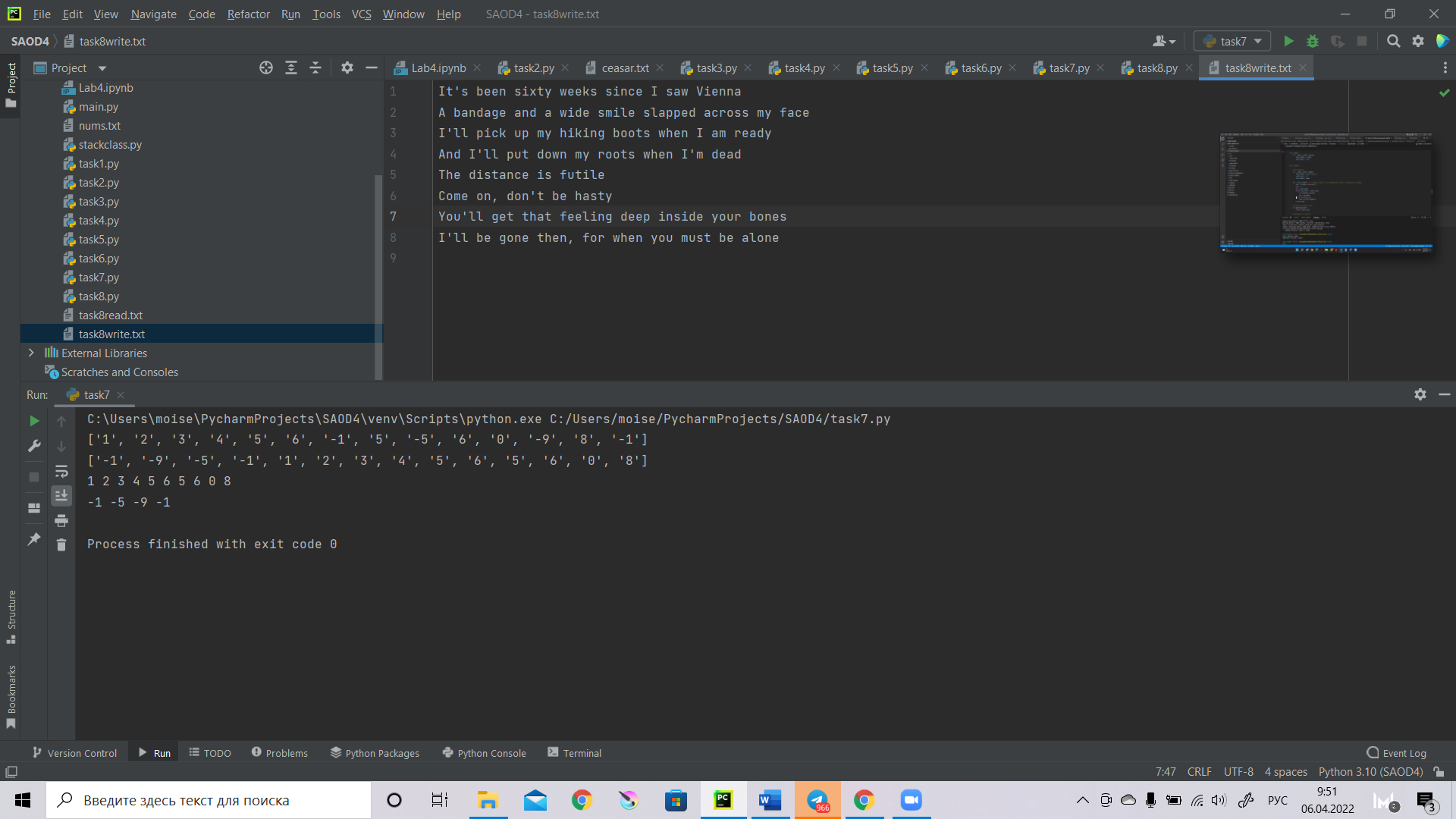
Рисунок 8 – Решение для задания 7

**Задание 8**

Дан текстовый файл. Используя стек, сформировать новый текстовый файл, содержащий строки исходного файла, записанные в обратном порядке: первая строка становится последней, вторая – предпоследней и т.д.

from stackclass import \*  
  
file = open("task8read.txt", "r", encoding="utf-8")  
lines = file.readlines()  
st = Stack()  
for i in lines:  
 st.addLast(i)  
file = open("task8write.txt", "w")  
for i in range(st.size()):  
 file.write(st.items[-1])  
 st.removeLast()  
file.close()

Результат работы программы представлен на рисунке 8.



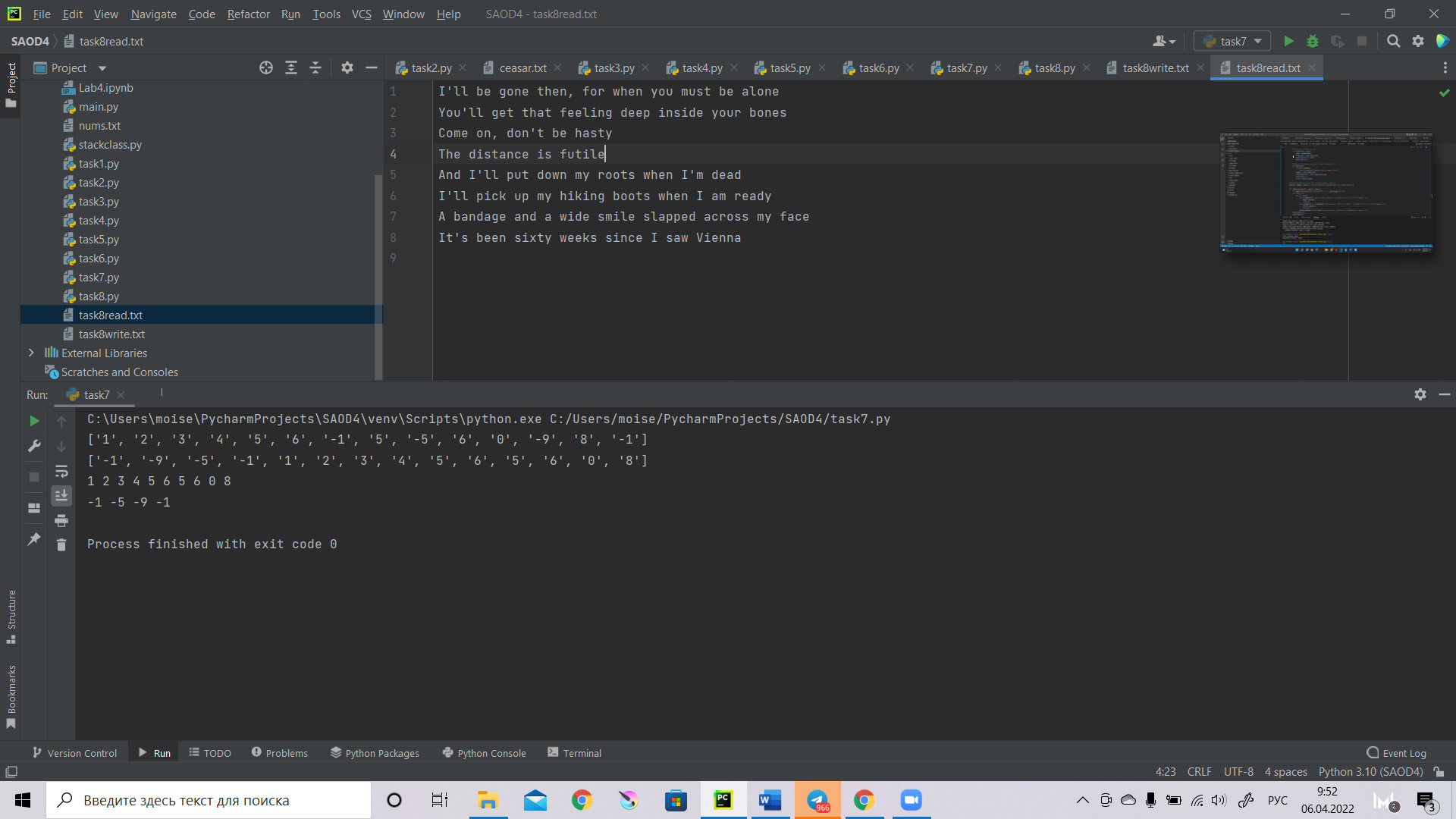


Рисунок 8 – Решение для задания 8

Вывод: Все задания были реализованы с использованием стеков и деков

Ссылка на github: https://github.com/mmaider/SAOD4

**Список литературы**

1. Камаев В.А., Костерин В.В. Технологии программирования. М.: Высшая школа,

2006.

2. Жоголев Е.А.Технология программирования. – М.: Научный мир, 2004.

Git:

3. Scott Chacon, Ben Straub «Pro Git»

4. git-scm.com