1. Создать программно файл в текстовом формате, записать в него построчно данные, вводимые пользователем. Об окончании ввода данных свидетельствует пустая строка.

my\_file = open(**"test.txt"**, **"w"**)  
**if** my\_file.writable():  
 strings = input(**"Введите текст \n"**)  
 my\_file.writelines(strings)  
**else**:  
 print(**"Can not write"**)  
  
my\_file.close()  
  
my\_file = open(**r"test.txt"**)  
content = my\_file.readlines()  
print(content)  
my\_file.close()

1. Создать текстовый файл (не программно), сохранить в нем несколько строк, выполнить подсчет количества строк, количества слов в каждой строке.

my\_file = open(**r"file.txt"**)  
**with** open(**r"file.txt"**) **as** file:  
 print(**f"Количество строк в файле - {**len(file.readlines())**}"**)  
my\_file = open(**r"file.txt"**)  
lines = my\_file.readlines()  
**for** i,value **in** enumerate(lines):  
 print(**f"Количество символов в {**i+1**} строке - {**len(lines[i])-1**}"**)  
my\_file = open(**r"file.txt"**)  
content = my\_file.read()  
content = content.split()  
print(**f"Общее количество слов - {**len(content)**}"**)  
my\_file.close()

3. Создать текстовый файл (не программно), построчно записать фамилии сотрудников и величину их окладов (не менее 10 строк). Определить, кто из сотрудников имеет оклад менее 20 тыс., вывести фамилии этих сотрудников. Выполнить подсчет средней величины дохода сотрудников.

**Пример файла:**

Иванов 23543.12  
Петров 13749.32

**with** open(**"salary.txt"**, **"r"**, encoding=**"utf-8"**) **as** read\_file:  
 f\_average\_salary = 0  
 all\_read\_lines = read\_file.readlines()  
 **for** line **in** all\_read\_lines:  
 s\_name, salary = line.split()  
 salary = float(salary)  
 f\_average\_salary += salary  
 **if** salary < 20000:  
 print(**f"\t{**s\_name**} {**salary**}"**)  
 **if** len(all\_read\_lines) > 0:  
 f\_average\_salary /= len(all\_read\_lines)  
 print(**f"\tСредняя зарплата: {**f\_average\_salary**:.2f}\n"**)

4. Создать (не программно) текстовый файл со следующим содержимым:

One — 1  
Two — 2  
Three — 3  
Four — 4

Необходимо написать программу, открывающую файл на чтение и считывающую построчно данные. При этом английские числительные должны заменяться на русские. Новый блок строк должен записываться в новый текстовый файл.

**with** open(**"string.txt"**, **"r"**, encoding=**"utf-8"**) **as** read\_file:  
 **with** open(**"string1.txt"**, **"w"**, encoding=**"utf-8"**) **as** write\_file:  
 number\_words\_list = [**"Один"**, **"Два"**, **"Три"**, **"Четыре"**]  
 all\_read\_lines = read\_file.readlines()  
 **for** num, line **in** enumerate(all\_read\_lines):  
 **if** len(line):  
 words = line.split()  
 print(**f"{**number\_words\_list[num]**} {**words[1]**} {**words[2]**}"**, file=write\_file)  
 print(**"\tГотово\n"**)

5. Создать (программно) текстовый файл, записать в него программно набор чисел, разделенных пробелами. Программа должна подсчитывать сумму чисел в файле и выводить ее на экран.

**import** random  
  
**with** open(**'number.txt'**, **'w'**, encoding=**'UTF-8'**) **as** file:  
 glue = **''  
 for** \_ **in** range(5):  
 file.write(glue + str(random.randint(0, 10)))  
 glue = **' '  
  
with** open(**'number.txt'**, **'r'**, encoding=**'UTF-8'**) **as** file:  
 numbers\_str = file.read()  
 numbers\_lst = numbers\_str.split(**' '**)  
 print(**f"Содержимое файла:\n{**numbers\_str**}"**)  
 print(**f"Сумма чисел:\n{**sum([int(i) **for** i **in** numbers\_lst])**}"**)

6. Необходимо создать (не программно) текстовый файл, где каждая строка описывает учебный предмет и наличие лекционных, практических и лабораторных занятий по этому предмету и их количество. Важно, чтобы для каждого предмета не обязательно были все типы занятий. Сформировать словарь, содержащий название предмета и общее количество занятий по нему. Вывести словарь на экран.

**Примеры строк файла:** Информатика: 100(л) 50(пр) 20(лаб).  
Физика: 30(л) — 10(лаб)  
Физкультура: — 30(пр) —  
**Пример словаря:** {“Информатика”: 170, “Физика”: 40, “Физкультура”: 30}

**import** re  
  
report = {}  
**with** open(**"task06.txt"**, **"r"**, encoding=**"UTF-8"**) **as** file:  
 text = file.read()  
 file.seek(0)  
 **for** row **in** file:  
 row\_items = row.split(**": "**)  
 hours = re.findall(**r"\d+"**, row\_items[1])  
 report.update({row\_items[0]: sum([int(i) **for** i **in** hours])})  
  
print(**f"Исходный файл:\n{**text**}\n"**)  
print(**f"Словарь:\n{**report**}"**)

7. Создать вручную и заполнить несколькими строками текстовый файл, в котором каждая строка должна содержать данные о фирме: название, форма собственности, выручка, издержки.

**Пример строки файла:** firm\_1 ООО 10000 5000.

Необходимо построчно прочитать файл, вычислить прибыль каждой компании, а также среднюю прибыль. Если фирма получила убытки, в расчет средней прибыли ее не включать.

Далее реализовать список. Он должен содержать словарь с фирмами и их прибылями, а также словарь со средней прибылью. Если фирма получила убытки, также добавить ее в словарь (со значением убытков).

**Пример списка:** [{“firm\_1”: 5000, “firm\_2”: 3000, “firm\_3”: 1000}, {“average\_profit”: 2000}].

Итоговый список сохранить в виде json-объекта в соответствующий файл.

**Пример json-объекта:**  
  
[{"firm\_1": 5000, "firm\_2": 3000, "firm\_3": 1000}, {"average\_profit": 2000}]

**Подсказка:** использовать менеджер контекста.

**import** json  
  
report = []  
**with** open(**'task07.txt'**, **'r'**, encoding=**'UTF-8'**) **as** file:  
 text = file.read()  
 file.seek(0)  
 profits = {}  
 profit\_sum = 0  
 **for** row **in** file:  
 items = row.split(**' '**)  
 profit = int(items[2]) - int(items[3])  
 **if** profit > 0:  
 profits.update({items[0]: profit})  
 profit\_sum += profit  
 report.append(profits)  
 report.append({**'average\_profit'**: (profit\_sum / len(profits))})  
  
**with** open(**'task07.json.txt'**, **'w'**, encoding=**'UTF-8'**) **as** json\_file:  
 json.dump(report, json\_file, ensure\_ascii=**False**)  
  
json\_report = json.dumps(report, ensure\_ascii=**False**)  
  
print(**f"Исходный файл:\n{**text**}\n"**)  
print(**f"Список:\n{**report**}\n"**)  
print(**f"json-объект:\n{**json\_report**}"**)