<https://drive.google.com/file/d/1zCBeu1Xzth0qlatkepqVzy5TQ0EWuGrx/view?usp=sharing>

*# -------------------------------------------------------- 1 ----------------------------------------------------------*with open('text\_1.txt', 'w', encoding='utf-8') as f:  
 while True:  
 line = input('Input new string or empty string to done: ')  
 if not line:  
 break  
 *# f.write(f"{line}\n")* print(line, file=f)  
  
*# ------------------------------------------- вариант решения ---------------------------------------------------------*print('Введите данные для записи в файл \nДля окончания ввода введите пустую строку')  
with open('task\_1.txt', 'w', encoding='utf-8') as my\_file:  
 while (line := input()) != '':  
 my\_file.write(f"{line}\n")  
  
*# ------------------------------------------- вариант решения ---------------------------------------------------------*my\_file = open("text\_1.txt", 'w', encoding='utf-8')  
  
line = " "  
while line:  
 line = input("пишите или не пишите!: ")  
 my\_file.write(f"{line}\n") if line != '' else my\_file.close()  
  
*# -------------------------------------------------------- 2 ----------------------------------------------------------*with open("text\_2.txt", "r", encoding='utf-8') as f\_obj:  
 usefulness = [f'{line}. {" ".join(count.split())} - {len(count.split())} слов'  
 for line, count in enumerate(f\_obj, 1)]  
 print(\*usefulness, f"Всего строк - {len(usefulness)}.", sep="\n")  
  
  
*# ------------------------------------------- вариант решения ---------------------------------------------------------*with open("text\_2.txt", encoding='utf-8') as f:  
 my\_line = f.readlines()  
 for index, value in enumerate(my\_line, 1):  
 number\_of\_words = len(value.split())  
 print(f'Строка {index} содержит {number\_of\_words} слов')  
  
*# -------------------------------------------------------- 3 ----------------------------------------------------------*with open('text\_3.txt', 'r', encoding='utf-8') as f:  
 employees\_dict = {line.split()[0]: float(line.split()[1]) for line in f}  
 print(f'Average salary = {round(sum(employees\_dict.values()) / len(employees\_dict), 3)}\n'  
 f'Employees with salary less than 20k {[e[0] for e in employees\_dict.items() if e[1] < 20000]}')  
  
*# ------------------------------------------- вариант решения ---------------------------------------------------------*def task\_3():  
 wages = {}  
 try:  
 with open('task\_3.txt', 'r', encoding='utf-8') as file:  
 for line in file:  
 wages[line.split()[0]] = float(line.split()[1])  
 print('Меньше 20000 получают:')  
 for name, wage in wages.items():  
 if wage < 20000:  
 print(name)  
 print(f'Средняя зарплата равна {sum(wages.values()) / len(wages)}')  
 except IOError:  
 print('Бухгалтер сбежал с ведомостью. Зарплаты не будет')  
 except:  
 print('Что-то пошло не так')  
  
  
task\_3()  
print('Итого как всегда меньше всех работал и больше всех получает )))')  
  
*# -------------------------------------------------------- 4 ----------------------------------------------------------*rus\_dic = {"One": "Один", "Two": "Два", "Three": "Три", "Four": "Четыре"}  
  
with open("text\_44.txt", "w", encoding='utf-8') as new\_file:  
 with open("text\_4.txt", encoding='utf-8') as my\_file:  
 new\_file.writelines([line.replace(line.split()[0], rus\_dic.get(line.split()[0])) for line in my\_file])  
  
  
*# ------------------------------------------- вариант решения ---------------------------------------------------------  
# pip install googletrans==3.1.0a0  
# обновление до альфа-версии*from googletrans import Translator  
  
with open("text\_4\_translate.txt", 'w', encoding='utf-8') as f:  
 with open("text\_4.txt", 'r', encoding='utf-8') as f1:  
 text = f1.read()  
 try:  
 f.write(Translator().translate(text, dest='ru').text)  
 except AttributeError:  
 print("DDoS-атака на Google не прошла, продолжаем попытки!")  
  
  
*# ------------------------------------------- вариант решения ---------------------------------------------------------*import requests  
import json  
  
"""Переводит с английского на русский файл, результат записывается в новый файл.  
Должен быть установлен модуль requests.  
"""  
token = "trnsl.1.1.20200416T132512Z.0bdb58c00f70557b.b1aec4ed1dc72e76cc6c08980f7ed0c2de92ae86"  
url\_trans = 'https://translate.yandex.net/api/v1.5/tr.json/translate'  
  
with open("task\_4\_text\_yandex.txt", "w", encoding="utf-8") as f\_result:  
 with open("text\_4.txt", encoding="utf-8") as f\_4:  
 for line in f\_4:  
 eng\_text = line  
 trans\_option = {'key': token, 'lang': 'en-ru', 'text': eng\_text}  
 webRequest = requests.get(url\_trans, params=trans\_option)  
 trans\_dict = json.loads(webRequest.text)  
 line\_to\_result = "".join(trans\_dict["text"])  
 f\_result.write(line\_to\_result)  
  
print(f"Text translate from {f\_4.name} has been done in {f\_result.name}")  
  
*# -------------------------------------------------------- 5 ----------------------------------------------------------*from random import randint  
  
with open("text.txt", "w", encoding="utf-8") as my\_file:  
 my\_list = [randint(1, 100) for \_ in range(100)]  
 my\_file.write(" ".join(map(str, my\_list)))  
  
print(f"Sum of elements - {sum(my\_list)}")  
  
  
*# ------------------------------------------- вариант решения ---------------------------------------------------------*from random import randint  
  
  
with open('task\_5\_file.txt', mode='w+', encoding='utf-8') as the\_file:  
 the\_file.write(" ".join([str(randint(1, 1000)) for \_ in range(100000)]))  
 the\_file.seek(0)  
 print(sum(map(int, the\_file.readline().split())))  
  
  
*# -------------------------------------------------------- 6 ----------------------------------------------------------*mydict = {}  
with open("text\_6.txt", encoding="utf-8") as fobj:  
 for line in fobj:  
 name, stats = line.split(':')  
 name\_sum = sum(map(int, "".join([i for i in stats if i == ' ' or '9' >= i >= '0']).split()))  
 mydict[name] = name\_sum  
print(f"{mydict}")  
  
*# ------------------------------------------- вариант решения ---------------------------------------------------------*with open('text\_6.txt', 'r', encoding='utf8') as text\_file:  
 a = text\_file.readlines()  
 for s in a:  
 new\_str = ''.join((i if i in '1234567890' else ' ') for i in s)  
 new\_2 = [int(i) for i in new\_str.split()]  
 print(f'{s.split()[0]} {sum(new\_2)}')  
  
*# ------------------------------------------- вариант решения ---------------------------------------------------------*dic = {}  
numbers = "1234567890 "  
  
with open("text\_6.txt", "r", encoding="utf-8") as file:  
 for line in file:  
 head, hours = line.split(":")  
 dic[head] = sum(map(int, "".join([num for num in hours if num in numbers]).split()))  
print(dic)  
  
*# ------------------------------------------- вариант решения ---------------------------------------------------------*subj = {}  
with open('text\_6.txt', 'r', encoding='utf-8') as file:  
 for line in file:  
 line = line.replace('-', '0').replace(':', '').replace('(л)', '') \  
 .replace('(пр)', '').replace('(лаб)', '').split()  
 subj[line[0]] = sum(map(int, line[1:]))  
 print(f'Общее количество часов по предмету: \n{subj}')  
  
*# ------------------------------------------- вариант решения ---------------------------------------------------------*result = {}  
with open("text\_6.txt", encoding="utf-8") as f\_obj:  
 for line in f\_obj:  
 lesson\_timing = []  
 lessons = ([el for el in line.split(" ")])  
 for el in lessons:  
 lesson\_timing.append(''.join(i for i in list(el) if i.isdigit()))  
 result[line.split(':')[0]] = sum([int(i) for i in lesson\_timing if i.isdigit()])  
  
print(result)  
  
*# ------------------------------------------- вариант решения ---------------------------------------------------------*with open('text\_6.txt', 'r', encoding='utf-8') as file:  
 print({string.split(':')[0]: sum([int(s[:s.index('(')]) for s in string.split() if '(' in s]) for string in file})  
  
  
*# ------------------------------------------- вариант решения ---------------------------------------------------------*import re  
  
subs\_total\_hours = {}  
  
with open("text\_6.txt") as f:  
 for line in f.readlines():  
 subs\_total\_hours[re.findall(r'^\w+', line)[0]] = sum(map(int, re.findall(r'\d+', line)))  
 print(subs\_total\_hours)  
  
*# -------------------------------------------------------- 7 ----------------------------------------------------------*import json  
  
with open("my\_ex7.json", "w", encoding="utf-8") as write\_f:  
 with open("text\_7.txt", encoding="utf-8") as f\_obj:  
 profit = {line.split()[0]: int(line.split()[2]) - int(line.split()[3]) for line in f\_obj}  
 result = [profit, {"average\_profit": round(sum([int(i) for i in profit.values() if int(i) > 0]) /  
 len([int(i) for i in profit.values() if int(i) > 0]))}]  
 json.dump(result, write\_f, ensure\_ascii=False, indent=4)  
  
  
*# ------------------------------------------- вариант решения ---------------------------------------------------------*from json import dump  
  
with open('text\_7.txt', 'r', encoding='utf-8') as read\_file:  
 with open('text\_77.json', 'w', encoding='utf-8') as write\_file:  
 dictionary = {string.split()[0]: int(string.split()[2]) - int(string.split()[3]) for string in read\_file}  
 average\_profit\_lst = []  
 for n in dictionary.values():  
 if n > 0:  
 average\_profit\_lst.append(n)  
 dump([dictionary, {"average\_profit": sum(average\_profit\_lst) / len(average\_profit\_lst)}],  
 write\_file, ensure\_ascii=False, indent=4)