Міністерство освіти і науки України

Державний університет ,,Житомирська політехніка”

Кафедра КІ та КБ

Група: КБ-21-1

Програмування мовою Python

Лабораторна робота №6

«РОБОТА З ФАЙЛАМИ»

Виконала: Поліщук К. Р.

Прийняв: Морозов Д. С.

**Мета роботи:** ознайомитися з засобами роботи з файлами в мові Python, діями над ними.

**Хід роботи**

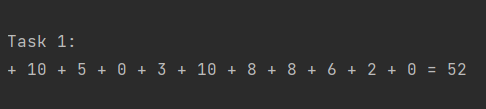
Завдання 1. Створіть новий файл numbers.txt у текстовому редакторі і запишіть у нього 10 чисел, кожне з нового рядка. Напишіть програму, яка зчитує ці числа з файла і обчислює їх суму, виводить цю суму на екран і, водночас, записує цю суму у інший файл з назвою sum\_numbers.txt.

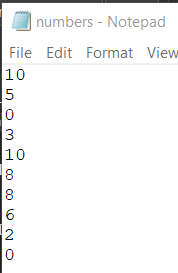
Лістинг програми:

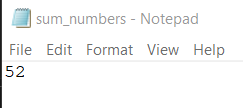
def task1():

f = open("files\\numbers.txt", "w")  
for i in range(10):  
 f.write(f"{random.randint(0, 10)}\n")  
f.close()  
f = open("files\\numbers.txt", "r")  
sum = 0  
print("Task 1:")  
for i in f:  
 sum += int(i)  
 print(f"+ {int(i)} ", end='')  
f.close()  
print(f"= {sum}")  
f = open("files\\sum\_numbers.txt", "w")  
f.write(f"{sum}")  
f.close()

Результат виконання програми:







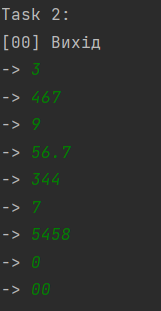
Завдання 2. Реалізуйте програму, яка зчитує довільну кількість цілих чисел, що вводяться з командного рядка, і записує у текстовий файл інформацію, щодо парності або непарності чисел.

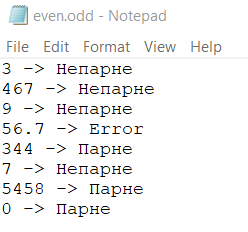
Лістинг програми:

def task2():

print("Task 2: ")  
f = codecs.open("files\\even.odd.txt", "w", "utf-8")  
print("[00] Вихід")  
i = 0  
while i != '00':  
 i = input("-> ")  
 if i == '00':  
 break  
 if not i.isdigit(): # користувач ввів не число  
 f.write(f"{i} -> Error\n")  
 continue  
 i = int(i)  
 if i % 2 == 0:  
 f.write(f"{i} -> Парне\n")  
 else:  
 f.write(f"{i} -> Непарне\n")  
f.close()

Результат виконання програми:



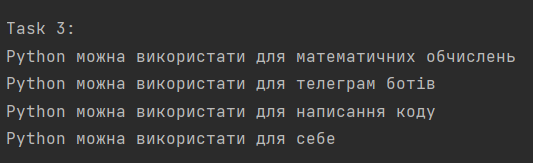


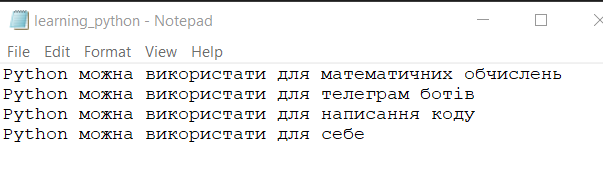
Завдання 3. Створіть новий файл у текстовому редакторі і напишіть кілька рядків тексту у ньому про можливості Python. Кожен рядок повинен починатися з фрази: «Python можна використати для ...» . Збережіть файл з ім’ям learning\_python.txt. Напишіть програму, яка зчитує файл і виводить текст з перебором рядків файла і зі збереженням рядків у списку з подальшим сортуванням списку за довжиною рядків в ньому від найбільшого до найменшого.

Лістинг програми:

def task3():  
 f = codecs.open("files\\learning\_python.txt", "w", "utf-8")  
 f.write(f"Python можна використати для телеграм ботів\n")  
 f.write(f"Python можна використати для написання коду\n")  
 f.write(f"Python можна використати для математичних обчислень\n")  
 f.write(f"Python можна використати для себе\n")  
 f.close()  
 f = codecs.open("files\\learning\_python.txt", "r", "utf-8")  
 lines = sorted(f.readlines(), key=len, reverse=True)  
 f.close()  
 f = codecs.open("files\\learning\_python.txt", "w", "utf-8")  
 print("Task 3: ")  
 for i in lines:  
 print(i, end='')  
 f.write(f"{i}")  
 f.close()

Результат виконання програми:



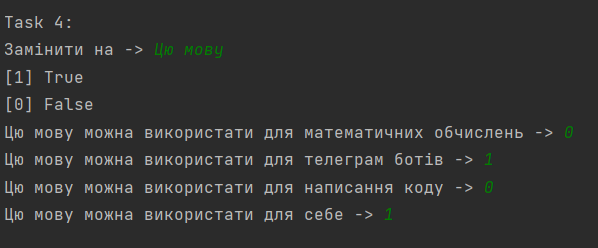


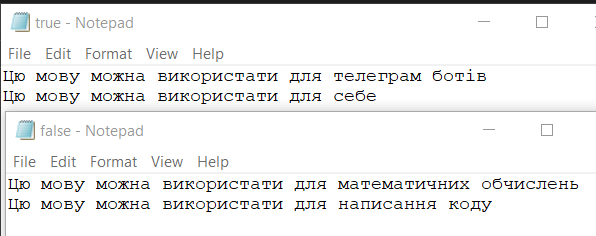
Завдання 4. Прочитайте кожен рядок зі створеного у попередньому завданні файла learning\_python.txt і замініть слово Python назвою іншої мови, наприклад C при виведенні на екран. Отриманий файл має бути створений в новому каталозі, що розміщується в поточному. Відкрийте файл пострічково і дайте можливість користувачеві визначити які змінені фрази є актуальними, наприклад для мови С, а які ні. Всі хибні твердження запишіть в інший файл, а істинні – в поточний.

Лістинг програми:

def task4():import os # Модуль для роботи з каталогами та шляхами  
 print("Task 4:")  
 with codecs.open('files\\learning\_python.txt', 'r', "utf-8") as f:  
 text = f.read().replace('Python', input("Замінити на -> "))  
 if not os.path.exists("files\\directory\_task\_4"):  
 os.mkdir("files\\directory\_task\_4")  
 with codecs.open('files\\directory\_task\_4\\all.txt', 'w', "utf-8") as all:  
 all.write(f"{text}")  
 with codecs.open('files\\directory\_task\_4\\all.txt', 'r', "utf-8") as all:  
 true = codecs.open('files\\directory\_task\_4\\true.txt', 'w', "utf-8")  
 false = codecs.open('files\\directory\_task\_4\\false.txt', 'w', "utf-8")  
 print("[1] True")  
 print("[0] False")  
 for i in all:  
 check = '-1'  
 while check != '0' and check != '1':  
 print(f"{i[:-1]} -> ", end='')  
 check = input()  
 if check == '1':  
 true.write(f"{i}")  
 elif check == '0':  
 false.write(f"{i}")  
 true.close()  
 false.close()  
 os.remove("files\\directory\_task\_4\\all.txt")

Результат виконання програми:



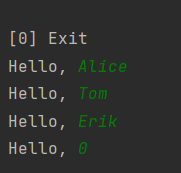


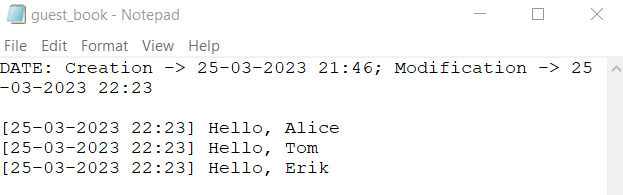
Завдання 5. Створіть порожній файл guest\_book.txt у текстовому редакторі. Напишіть програму, яка запитує у користувачів імена. При введенні кожного імені виведіть на екран рядок з вітанням для користувача і запишіть рядок вітання у файл з ім’ям guest\_book.txt. Простежте за тим, щоб кожне повідомлення розміщувалося в окремому рядку файла з зазначенням часу внесення цього повідомлення. Передбачте зазначення в файлі часу його створення і вказання в ньому часу останніх внесених змін.

Лістинг програми:

def task5():with codecs.open('files\\guest\_book.txt', 'a+', "utf-8") as f:  
 creation = 'Creation -> ' + datetime.datetime.fromtimestamp(  
 pathlib.Path('files\\guest\_book.txt').stat().st\_ctime).strftime("%d-%m-%Y %H:%M")  
  
 modification = 'Modification -> ' + datetime.datetime.fromtimestamp(  
 pathlib.Path('files\\guest\_book.txt').stat().st\_mtime).strftime("%d-%m-%Y %H:%M")  
  
 f.seek(0)  
 if 'DATE' not in f.read():  
 print('error')  
 f.seek(0)  
 content = f.read()  
 with codecs.open('files\\guest\_book.txt', 'w', "utf-8") as w:  
 w.write(f"DATE: {creation}; {modification}\n{content}")  
 print("[0] Exit")  
 name = '-1'  
 while name != '0':  
 name = input("Hello, ")  
 if name == '0':  
 break  
 timeNow = datetime.datetime.now().strftime("%d-%m-%Y %H:%M")  
 f.write(f"\n[{timeNow}] Hello, {name}")  
 f.seek(0)  
 content = f.readlines()  
 modification = 'Modification -> ' + datetime.datetime.fromtimestamp(  
 pathlib.Path('files\\guest\_book.txt').stat().st\_mtime).strftime("%d-%m-%Y %H:%M")  
 with codecs.open('files\\guest\_book.txt', 'w', "utf-8") as w:  
 w.write(f"DATE: {creation}; {modification}\n")  
 for i in content[1:]:  
 w.write(i)

Результат виконання програми:



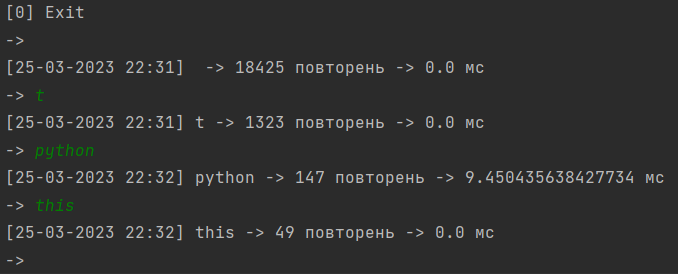


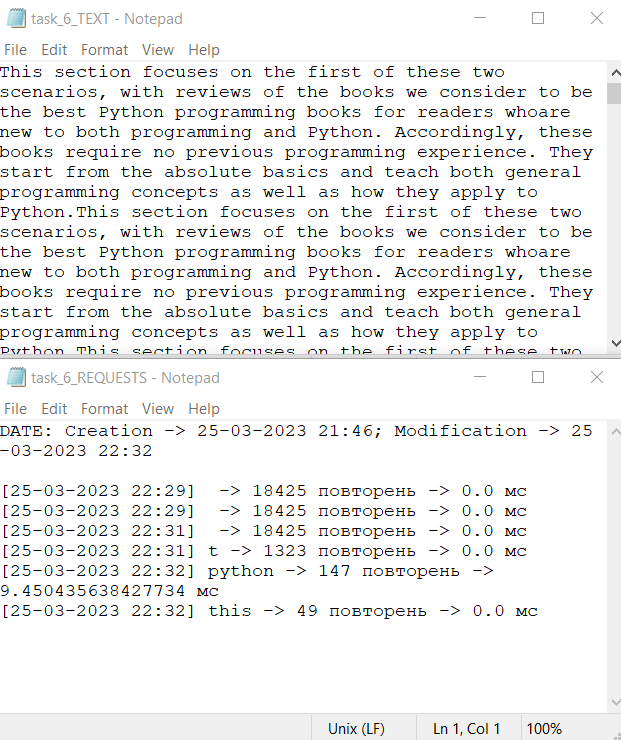
Завдання 6. Збережіть в текстовому файлі публікацію про Python на 3000 слів англійською мовою. Напишіть програму, що аналізуватиме частоту з якою в тексті зустрічатимуться окремі літери чи слова незалежно від їх регістру. Результат робот програми має виводитись в консоль і зберігатись в окремому файлі з зазначенням часу його створення, часу виконання окремих змін, результатів пошуку і часу, що знадобився на виконання цього пошуку.

Лістинг програми:

def task6():with codecs.open('files\\task\_6\_TEXT.txt', 'w', "utf-8") as fileText:  
 str = 'This section focuses on the first of these two scenarios, with reviews ' \  
 'of the books we consider to be the best Python programming books for readers who' \  
 'are new to both programming and Python. Accordingly, these books require no ' \  
 'previous programming experience. They start from the absolute basics and teach ' \  
 'both general programming concepts as well as how they apply to Python.'  
 for i in range(int(round(3000 / len(str.split()), 0))):  
 fileText.write(str)  
 with codecs.open('files\\task\_6\_TEXT.txt', 'r', "utf-8") as fileText:  
 text = fileText.read()  
 with codecs.open('files\\task\_6\_REQUESTS.txt', 'a+', "utf-8") as f:  
 creation = 'Creation -> ' + datetime.datetime.fromtimestamp(  
 pathlib.Path('files\\task\_6\_REQUESTS.txt').stat().st\_ctime).strftime("%d-%m-%Y %H:%M")  
 modification = 'Modification -> ' + datetime.datetime.fromtimestamp(  
 pathlib.Path('files\\task\_6\_REQUESTS.txt').stat().st\_mtime).strftime("%d-%m-%Y %H:%M")  
 f.seek(0)  
 if 'DATE' not in f.read():  
 print('error')  
 f.seek(0)  
 content = f.read()  
 with codecs.open('files\\task\_6\_REQUESTS.txt', 'w', "utf-8") as w:  
 w.write(f"DATE: {creation}; {modification}\n{content}")  
 print("[0] Exit")  
 find = '-1'  
 while find != '0':  
 find = input("-> ")  
 if find == '0':  
 break  
 timeNow = datetime.datetime.now().strftime("%d-%m-%Y %H:%M")  
 start\_time = time.time()  
 count = text.lower().count(find.lower())  
 timeFind = (time.time() - start\_time) \* 10 \*\* 3  
 f.write(f"\n[{timeNow}] {find} -> {count} повторень -> {timeFind} мс")  
 print(f"[{timeNow}] {find} -> {count} повторень -> {timeFind} мс")  
 f.seek(0)  
 content = f.readlines()  
 modification = 'Modification -> ' + datetime.datetime.fromtimestamp(  
 pathlib.Path('files\\task\_6\_REQUESTS.txt').stat().st\_mtime).strftime("%d-%m-%Y %H:%M")  
 with codecs.open('files\\task\_6\_REQUESTS.txt', 'w', "utf-8") as w:  
 w.write(f"DATE: {creation}; {modification}\n")  
 for i in content[1:]:  
 w.write(i)

Результат виконання програми:



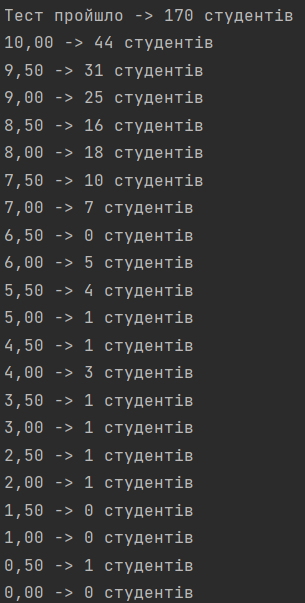


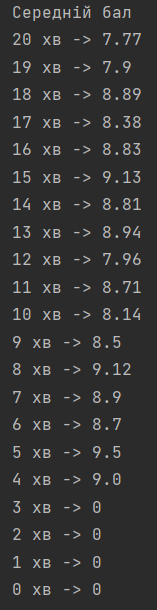
Завдання 7. Завантажте файл marks.csv і визначте кількість студентів, що проходили тестування. Виведіть інформацію про те яку оцінку набрали відповідна кількість студентів. Виведіть інформацію яку середню оцінку отримував студент за певний час виконання КМР (крок – 1 хв). Створіть текстовий файл і запишіть в нього статистику по правильним відповідям для кожного окремого питання (який відсоток правильних і неправильних відповідей на питання дали студенти). В цей же файл внесіть інформацію про 5 найкращих оцінок в співвідношенні оцінка/час витрачений складання КМР.

Лістинг програми:

def task7():  
 import csv # Моудль для роботи з файлами з розширенням .csv  
  
 with codecs.open('files\\marks.lab6.csv', mode='r', encoding='utf-8') as file:  
 allStudent = sum(1 for row in file)  
 print(f"Тест пройшло -> {allStudent} студентів")  
 file.seek(0)  
 rating = []  
 for i in range(21):  
 rating.append(0)  
 for row in csv.reader(file):  
 rating[int(float(row[4].replace(',', '.')) \* 2)] += 1  
 for i in reversed(range(len(rating))):  
 print(f"{str(i / 2).replace('.', ',')}0 -> {rating[i]} студентів")  
 file.seek(0)  
 average = []  
 for i in range(21):  
 average.append([0, 0])  
 for row in csv.reader(file):  
 average[int(row[3][:2])][0] += float(row[4].replace(',', '.'))  
 average[int(row[3][:2])][1] += 1  
 print('Середній бал')  
 for i in reversed(range(21)):  
 try:  
 print(f"{i} хв -> {round(average[i][0] / average[i][1], 2)}")  
 except ZeroDivisionError:  
 print(f"{i} хв -> {0}")  
 with codecs.open('files\\task\_7.txt', 'w', "utf-8") as filetxt:  
 filetxt.write(f"TASK i -> ПРАВИЛЬНІ/НЕПРАВИЛЬНІ")  
 true = 0  
 for i in range(20):  
 file.seek(0)  
 for row in csv.reader(file):  
 try:  
 if float(row[5 + i].replace(',', '.')) == 0.5:  
 true += 1  
 except ValueError:  
 true += 0  
 percent = round((true \* 100) / allStudent, 2)  
 filetxt.write(f"\nTASK {i + 1} -> {percent} % / {round(100 - percent, 2)} % ")  
 true = 0  
 file.seek(0)  
 bestRating = []  
 for row in csv.reader(file):  
 bestRating.append([round(float(row[4].replace(',', '.')) /  
 (int(row[3][:2]) + int(row[3][-6:-3]) / 60), 2), row[0], row[3], row[4]])  
 bestRating = sorted(bestRating, key=lambda x: x[0], reverse=True)  
 filetxt.write("\nСПІВВІДНОШЕННЯ ОЦІНКА/ЧАС")  
 for i in range(5):  
 filetxt.write(f"\n[{i + 1}] {bestRating[i][1]} -> {bestRating[i][3]} / {bestRating[i][2]} = {str(bestRating[i][0]).replace('.', ',')}")

Результат виконання програми:





**Висновок:** у ході виконання лабораторної роботи ми навчилися працювати з засобами роботи з файлами в мові Python, діями над ними.