

Міністерство освіти і науки України
Державний університет “Житомирська політехніка”

Кафедра інженерії програмного забезпечення
Група: ВТ-21-1[1]

Програмування мовою Python
Лабораторна робота № 6
«Робота з файлами»

Виконав:

Бабушко А. С.

Прийняв:

Морозов Д. С.

					«Житомирська політехніка».22.121.01.000–Лр6				
Змн.	Арк.	№ докум.	Підпис	Дата					
Розроб.		Бабушко А.С.			Звіт з лабораторної роботи		Літ.	Арк.	Аркушів
Перевір.		Морозов Д.С.						1	23
Керівник				ФІКТ Гр. ВТ-21-1[1]					
Н. контр.									
Затверд.									

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

Хід роботи:

Завдання на лабораторну роботу:

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 файлу і обчислює їх суму, виводить цю суму на екран і, водночас, записує цю суму у інший файл з назвою sum_numbers.txt.

Лістинг програми:

```
""" Lab 6. Python. Andrii Babushko. Repository: https://github.com/AndriiBabushko/Python """
import csv
import os
import io
from datetime import datetime
from datetime import date
from time import perf_counter

# task 1
"""
    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 файлу і обчислює їх суму, виводить цю
    суму на екран і, водночас,
    записує цю суму у інший файл з назвою sum_numbers.txt.
"""

def task_1():
    print('\nTASK 1!')
    if not os.path.isdir(r'./task1'):
        os.mkdir(r'./task1')

    with io.open(r'./task1/numbers.txt', 'wt', encoding='utf-8') as numbers_txt:
        numbers_txt.write(create_numbers_string())

    with io.open(r'./task1/numbers.txt', 'rt', encoding='utf-8') as numbers_txt:
        task_1_sum = 0
        task_1_number_list = []

        for number in numbers_txt:
            try:
                task_1_sum += int(number)
                task_1_number_list.append(int(number))
            except ValueError as err:
                print(f'ERROR! {ValueError}')

        with io.open(r'./task1/sum_numbers.txt', 'wt', encoding='utf-8') as
sum_numbers_txt:
            sum_numbers_txt.write(str(task_1_sum))

        print(f'Received numbers from file: {task_1_number_list}')
        print(f'Sum of numbers in file = {task_1_sum}')

def create_numbers_string():
    from random import randint
    new_string = ''

    for new_number in range(10):
```

		Бабушко А.С.			«Житомирська політехніка».22.121.01.000 – Лр6	Арк.
		Морозов Д.С.				2
Змн.	Арк.	№ докум.	Підпис	Дата		

```

new_string += f'{randint(-100, 100)}\n'

return new_string

task_1()

```

Результат програми:

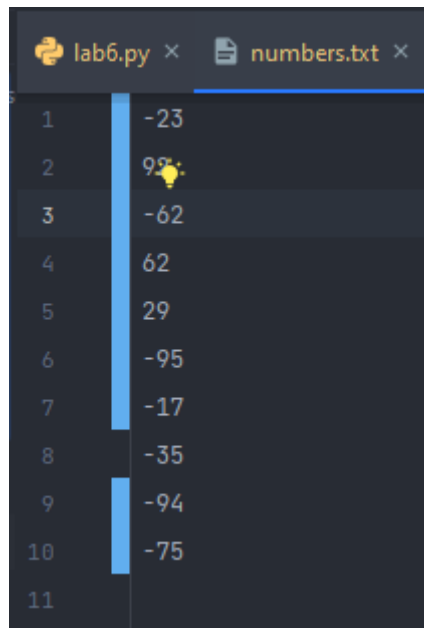
```

TASK 1!

Received numbers from file: [-23, 92, -62, 62, 29, -95, -17, -35, -94, -75]

Sum of numbers in file = -218

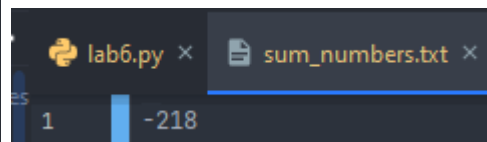
```



```

lab6.py × numbers.txt ×
1 -23
2 92
3 -62
4 62
5 29
6 -95
7 -17
8 -35
9 -94
10 -75
11

```



```

lab6.py × sum_numbers.txt ×
1 -218

```

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:

```

# task 2
"""
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.
"""

def task_2():
    print('\nTASK 2!')
    task_2_enter_count = int(input('Enter how much numbers you want to enter: '))
    task_2_number_list = task_2_enter_some_numbers(task_2_enter_count)

    if not os.path.isdir(r'./task2'):
        os.mkdir(r'./task2')

    with io.open(r'./task2/even_or_odd_numbers.txt', 'wt', encoding='utf-8') as
even_or_odd_numbers_txt:
        for number in task_2_number_list:
            if number % 2 == 0:
                even_or_odd_numbers_txt.write(f'{number} is even number!\n')
            else:
                even_or_odd_numbers_txt.write(f'{number} is odd number!\n')

```

		Бабушко А.С.			«Житомирська політехніка».22.121.01.000 – Лр6	Арк.
		Морозов Д.С.				3
Змн.	Арк.	№ докум.	Підпис	Дата		

```

with io.open(r'./task2/even_or_odd_numbers.txt', 'rt', encoding='utf-8') as
even_or_odd_numbers_txt:
    for line in even_or_odd_numbers_txt:
        print(line, end="")

def task_2_enter_some_numbers(count):
    numbers_list = []

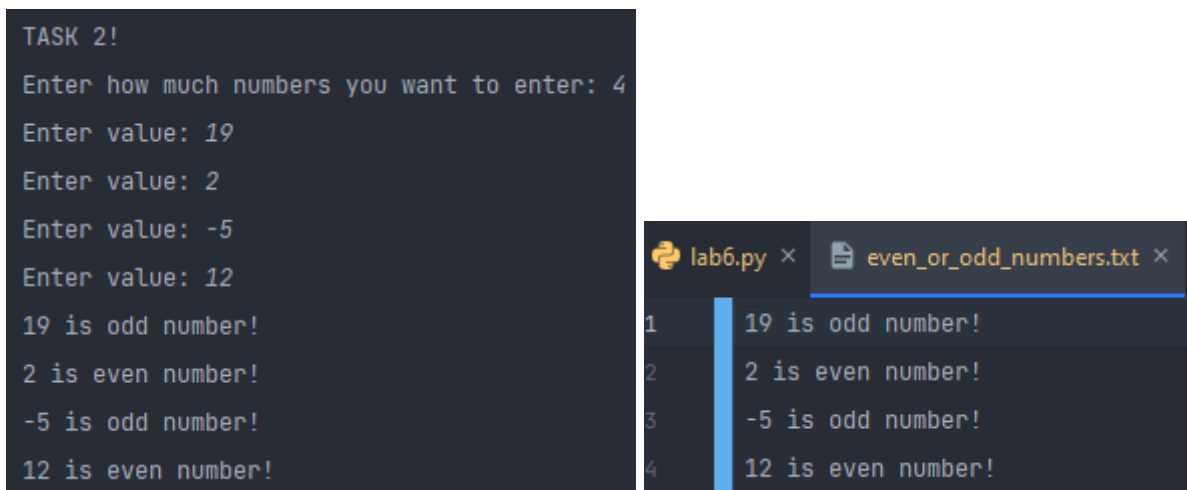
    while count != 0:
        try:
            int_number = int(input('Enter value: '))
            numbers_list.append(int_number)
            count -= 1
        except ValueError as value_error:
            numbers_list.pop()
            print('ERROR:', value_error)

    return numbers_list

task_2()

```

Результат програми:



```

TASK 2!
Enter how much numbers you want to enter: 4
Enter value: 19
Enter value: 2
Enter value: -5
Enter value: 12
19 is odd number!
2 is even number!
-5 is odd number!
12 is even number!

```

```

lab6.py x even_or_odd_numbers.txt x
1 19 is odd number!
2 2 is even number!
3 -5 is odd number!
4 12 is even number!

```

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:

```

# task 3
"""
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.
"""

```

		Бабушко А.С.			«Житомирська політехніка».22.121.01.000 – Лр6	Арк.
		Морозов Д.С.				4
Змн.	Арк.	№ докум.	Підпис	Дата		

```
def task_3():
    print('\nTASK 3!')

    if not os.path.isdir(r'./task3'):
        os.mkdir('./task3')

    with io.open(r'./task3/learning_python.txt', 'wt', encoding='utf-8') as
learning_python_txt:
        learning_python_txt.writelines(
            ["Python можна використати для machine learning.\n",
            "Python можна використати для написання простих скриптів.\n",
            "Python можна використати для програмування веб додатків.\n",
            "Python можна використати для написання особистого телеграм боту.\n"])

    with io.open(r'./task3/learning_python.txt', 'rt', encoding='utf-8') as
learning_python_txt:
        python_string_list = []

        for line in learning_python_txt:
            python_string_list.append(line)

        python_string_sorted_list = python_string_list[:]
        python_string_sorted_list.sort(key=len)
        print('\nSorted list:')
        for string in python_string_sorted_list:
            print(string, end="")

        python_string_reversed_sorted_list = python_string_list[:]
        python_string_reversed_sorted_list.sort(key=len, reverse=True)
        print('\nReversed sorted list:')
        for string in python_string_reversed_sorted_list:
            print(string, end="")

task_3()
```

Результат програми:

```
TASK 3!

Sorted list:
Python можна використати для machine learning.
Python можна використати для написання простих скриптів.
Python можна використати для програмування веб додатків.
Python можна використати для написання особистого телеграм боту.

Reversed sorted list:
Python можна використати для написання особистого телеграм боту.
Python можна використати для написання простих скриптів.
Python можна використати для програмування веб додатків.
Python можна використати для machine learning.
```

		Бабушко А.С.			«Житомирська політехніка».22.121.01.000 – Лр6	Арк.
		Морозов Д.С.				5
Змн.	Арк.	№ докум.	Підпис	Дата		

```

lab6.py × learning_python.txt ×
1 Python можна використати для machine learning.
2 Python можна використати для написання простих скриптів.
3 Python можна використати для програмування веб додатків.
4 Python можна використати для написання особистого телеграм боту.
5

```

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 `C`, а які ні. Всі хибні твердження запишіть в інший файл, а істинні – в поточний.

Лістинг програми:

```

# task 4
"""
    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 C, а які ні. Всі хибні твердження
    запишіть в інший файл, а істинні
    - в поточний.
"""

def task_4():
    print('\nTASK 4!')

    if not os.path.isdir(r'./task4'):
        os.mkdir(r'./task4')

    with io.open(r'./task3/learning_python.txt', 'rt', encoding='utf-8') as
learning_python_txt:
        python_strings = []

        for string in learning_python_txt:
            python_strings.append(string.split(' '))

        for i in range(0, len(python_strings)):
            python_strings[i][0] = 'C'
            python_strings[i] = " ".join(python_strings[i])

        with io.open(r'./task4/learning_c.txt', 'wt', encoding='utf-8') as true_about_c_txt:
            for c_string in python_strings:
                true_about_c_txt.write(c_string)

        with io.open(r'./task4/learning_c.txt', 'rt', encoding='utf-8') as learning_c_txt:
            for c_string in learning_c_txt:
                print(c_string, end='')
                answer = enter_true_or_false()
                if answer:
                    with io.open(r'./task4/true_about_c.txt', 'at', encoding='utf-8') as
true_about_c_txt:
                        true_about_c_txt.write(c_string)
                else:

```

		Бабушко А.С.			«Житомирська політехніка».22.121.01.000 – Лр6	Арк.
		Морозов Д.С.				6
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        with io.open(r'./task4/false_about_c.txt', 'at', encoding='utf-8') as
false_about_c_txt:
            false_about_c_txt.write(c_string)

        with io.open(r'./task4/true_about_c.txt', 'rt', encoding='utf-8') as
true_about_c_txt_read:
            for c_string in true_about_c_txt_read:
                print(c_string, end='')

        with io.open(r'./task4/false_about_c.txt', 'rt', encoding='utf-8') as
false_about_c_txt_read:
            for c_string in false_about_c_txt_read:
                print(c_string, end='')

def enter_true_or_false():
    while True:
        try:
            answer = input("Enter 'true' or 'false' about this statement: ")
            if answer != 'true' and answer != 'false':
                raise ValueError('Value was entered incorrectly!')
            else:
                pass
            if answer == 'true':
                return True
            elif answer == 'false':
                return False
        except ValueError as value_err:
            print(f'ERROR! {value_err}')

task_4()

```

Результат програми:

```

TASK 4!

С можна використати для machine learning.
Enter 'true' or 'false' about this statement: false
С можна використати для написання простих скриптів.
Enter 'true' or 'false' about this statement: true
С можна використати для програмування веб додатків.
Enter 'true' or 'false' about this statement: false
С можна використати для написання особистого телеграм боту.
Enter 'true' or 'false' about this statement: false
True statements about C:
С можна використати для написання простих скриптів.
False statements about C:
С можна використати для machine learning.
С можна використати для програмування веб додатків.
С можна використати для написання особистого телеграм боту.

```

		Бабушко А.С.			«Житомирська політехніка».22.121.01.000 – Лр6	Арк.
		Морозов Д.С.				7
Змн.	Арк.	№ докум.	Підпис	Дата		

```
lab6.py × learning_c.txt × false_about_c.txt × true_about_c.txt ×
1 С можна використати для machine learning.
2 С можна використати для написання простих скриптів.
3 С можна використати для програмування веб додатків.
4 С можна використати для написання особистого телеграм боту.
5
```

```
lab6.py × learning_c.txt × false_about_c.txt × true_about_c.txt ×
1 С можна використати для machine learning.
2 С можна використати для програмування веб додатків.
3 С можна використати для написання особистого телеграм боту.
4
```

```
lab6.py × learning_c.txt × false_about_c.txt × true_about_c.txt ×
1 С можна використати для написання простих скриптів.
2
```

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:

```
# task 5
"""
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
"""

def task_5():
    print('\nTASK 5!')

    if not os.path.isdir(r'./task5'):
        os.mkdir(r'./task5')

    task_5_count = enter_count()
    file_create_time = date.today().strftime('%B %d, %Y')
```



```

with io.open(r'./task5/guest_book.txt', 'wt', encoding='utf-8') as guest_book_txt:
    guest_book_txt.write(f'File was created: {file_create_time}\n\n')

    for counter in range(0, task_5_count):
        name_greeting = f'{counter + 1}) Hello, ' + enter_name() + '! '
        message_time = datetime.now().strftime('%H:%M:%S')
        name_greeting += f'Name was entered in time: {message_time}.\n'
        print(name_greeting, end="")
        guest_book_txt.write(name_greeting)

    last_changes_date = date.today().strftime('%B %d, %Y')
    last_changes_time = datetime.now().strftime('%H:%M:%S')
    guest_book_txt.write(f'\nLast changes time: {last_changes_date}
{last_changes_time}.\n')

def enter_name():
    while True:
        try:
            name = input('Enter name: ')
            if len(name) < 2:
                raise ValueError('Name\'s length is less than 2!')
            else:
                pass
            return name
        except ValueError as value_err:
            print(f'ERROR! {value_err}')

def enter_count():
    while True:
        try:
            count = int(input('Enter count of names: '))
            if count <= 0:
                raise ValueError('Count names was entered incorrectly!')
            else:
                pass
            return count
        except ValueError as value_err:
            print(f'ERROR! {value_err}')

task_5()

```

Результат програми:

```

TASK 5!
Enter count of names: 4
Enter name: Andrii
1) Hello, Andrii! Name was entered in time: 07:24:12.
Enter name: Nikole
2) Hello, Nikole ! Name was entered in time: 07:24:21.
Enter name: Abraham
3) Hello, Abraham! Name was entered in time: 07:24:31.
Enter name: Keira
4) Hello, Keira! Name was entered in time: 07:24:40.

```

		Бабушко А.С.			«Житомирська політехніка».22.121.01.000 – Лр6	Арк.
		Морозов Д.С.				9
Змн.	Арк.	№ докум.	Підпис	Дата		

```

lab6.py x guest_book.txt x
1 File was created: October 07, 2022
2
3 1) Hello, Andrii! Name was entered in time: 07:24:12.
4 2) Hello, Nikole ! Name was entered in time: 07:24:21.
5 3) Hello, Abraham! Name was entered in time: 07:24:31.
6 4) Hello, Keira! Name was entered in time: 07:24:40.
7
8 Last changes time: October 07, 2022 07:24:40.

```

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 роботи 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:

```

# task 6
"""
    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 роботи 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.
"""

def task_6():
    import re
    print('\nTASK 6!')

    with io.open(r'./task6/post_about_python.txt', 'rt', encoding='utf-8') as
post_about_python_txt:
        words = re.split(r' |\n| |? |\.\.|\, |\: |\;|', post_about_python_txt.read().lower())

    file_create_date = date.today().strftime('%B %d, %Y')
    file_create_time = datetime.now().strftime('%H:%M:%S')
    with io.open(r'./task6/word_frequency.txt', 'wt', encoding='utf-8') as
word_frequency_txt:
        word_frequency_txt.write(f'File was created: {file_create_date}
{file_create_time}\n\n')
        print(f'File was created: {file_create_time}\n')

    words_without_duplicates = []
    for string in words:
        if string not in words_without_duplicates:
            words_without_duplicates.append(string)

    print(words_without_duplicates)
    for i in range(0, len(words_without_duplicates)):
        time_started = perf_counter()
        counter = 0

```

		Бабушко А.С.			«Житомирська політехніка».22.121.01.000 – Лр6	Арк.
		Морозов Д.С.				10
Змн.	Арк.	№ докум.	Підпис	Дата		

```

for j in range(0, len(words)):
    if words_without_duplicates[i] == words[j]:
        counter += 1
time_finished = perf_counter()
write_time = round(time_finished - time_started, 8)
word_frequency_txt.write(f'"{words[i]}" repeats "{counter}" times. Write time:
{write_time}.\n')
print(f'"{words[i]}" repeats "{counter}" times. Write time: {write_time}.')

last_changes_date = date.today().strftime('%B %d, %Y')
last_changes_time = datetime.now().strftime('%H:%M:%S')
word_frequency_txt.write(f'\nLast changes time: {last_changes_date}
{last_changes_time}.\n')
print(f'\nLast changes time: {last_changes_date} {last_changes_time}.')

task_6()

```

Результат програми:

TASK 6!

File was created: 07:24:41

```

['what', 'is', 'python', 'used', 'for', 'a', 'beginner
"what" repeats "7" times. Write time: 7.85e-05.
"is" repeats "16" times. Write time: 9.95e-05.
"python" repeats "35" times. Write time: 9.86e-05.
"used" repeats "17" times. Write time: 9.92e-05.
"for" repeats "21" times. Write time: 9.94e-05.
"a" repeats "28" times. Write time: 9.69e-05.
"beginner's" repeats "1" times. Write time: 9.53e-05.
"guide" repeats "1" times. Write time: 9.75e-05.
"python" repeats "8" times. Write time: 0.0001006.
"has" repeats "2" times. Write time: 9.64e-05.
"become" repeats "3" times. Write time: 9.74e-05.

```

```

"learning" repeats "1" times. Write time: 7.3e-05.
"algorithms" repeats "1" times. Write time: 7.3e-05.
"manipulate" repeats "1" times. Write time: 8.57e-05.
"and" repeats "1" times. Write time: 7.28e-05.
"analyze" repeats "1" times. Write time: 7.17e-05.
"data" repeats "1" times. Write time: 7.14e-05.
"and" repeats "1" times. Write time: 7.15e-05.
"complete" repeats "1" times. Write time: 7.09e-05.
"other" repeats "1" times. Write time: 7.22e-05.
"data-related" repeats "1" times. Write time: 7.11e-05.

Last changes time: October 07, 2022 07:24:41.

```

		Бабушко А.С.			«Житомирська політехніка».22.121.01.000 – Лр6	Арк.
		Морозов Д.С.				11
Змн.	Арк.	№ докум.	Підпис	Дата		

```

lab6.py × post_about_python.txt × word_frequency.txt ×
1 What Is Python Used For? A Beginner's Guide
2 Python has become one of the most popular programming languages in the world in recent years.
3 Python, one of the most popular programming languages in the world, has created everything fr
4 Let's take a closer look at what Python is, what it can do, and how you can start learning it
5 What is Python?
6 Python is a computer programming language often used to build websites and software, automate
7 What is Python used for?
8 Python is commonly used for developing websites and software, task automation, data analysis,
9 "Writing programs is a very creative and rewarding activity," says University of Michigan and
10 What can you do with python? Some things include:
11 Data analysis and machine learning
12 Web development
13 Automation or scripting
14 Software testing and prototyping
15 Everyday tasks
16 Here's a closer look at some of these common ways Python is used.
17 Data analysis and machine learning
18 Python has become a staple in data science, allowing data analysts and other professionals to
19 Python can build a wide range of different data visualizations, like line and bar graphs, pie
20

```

```

lab6.py × post_about_python.txt × word_frequency.txt ×
1 File was created: October 07, 2022 07:24:41
2
3 "what" repeats "7" times. Write time: 7.85e-05.
4 "is" repeats "16" times. Write time: 9.95e-05.
5 "python" repeats "35" times. Write time: 9.86e-05.
6 "used" repeats "17" times. Write time: 9.92e-05.
7 "for" repeats "21" times. Write time: 9.94e-05.
8 "a" repeats "28" times. Write time: 9.69e-05.
9 "beginner's" repeats "1" times. Write time: 9.53e-05.
10 "guide" repeats "1" times. Write time: 9.75e-05.
11 "python" repeats "8" times. Write time: 0.0001006.
12 "has" repeats "2" times. Write time: 9.64e-05.
13 "become" repeats "3" times. Write time: 9.74e-05.
14 "one" repeats "18" times. Write time: 9.89e-05.
15 "of" repeats "14" times. Write time: 9.86e-05.
16 "the" repeats "2" times. Write time: 9.82e-05.
17 "most" repeats "5" times. Write time: 9.68e-05.

```

		Бабушко А.С.			«Житомирська політехніка».22.121.01.000 – Лр6	Арк.
		Морозов Д.С.				12
Змн.	Арк.	№ докум.	Підпис	Дата		

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 отримували 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:

```
# task 7
"""
    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 отримували 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.
"""

def task_7():
    print('\nTASK 7!')

    if not os.path.isdir(r'./task7'):
        os.mkdir(r'./task7')

    with io.open(r'./task7/marks.lab6.csv', 'rt', encoding='utf-8') as marks_lab6_csv:
        marks_csv = csv.reader(marks_lab6_csv)
        marks = [line for line in marks_csv]

    students_count = len(marks)
    print(f'Total students count who wrote a test: {students_count}')

    students_marks = []
    for i in range(students_count):
        mark_string = marks[i][4].split(',')
        mark_number = float(mark_string[0]) + float(int(mark_string[1]) / 100)
        students_marks.append(mark_number)

    print()
    iterator = 1
    for mark in students_marks:
        print(f'{iterator} student received {mark} mark.')
        iterator += 1

    students_time = []
    for i in range(students_count):
        time = marks[i][3].split(' ')
        if len(time) > 2:
            minutes = int(time[0])
            seconds = int(time[2]) + minutes * 60
        else:
            minutes = int(time[0])
            seconds = minutes * 60
        students_time.append(seconds)

    students_mark_per_min = []
    for i in range(students_count):
```

		Бабушко А.С.			«Житомирська політехніка».22.121.01.000 – Лр6	Арк.
		Морозов Д.С.				13
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        students_mark_per_min.append(round((students_marks[i] / students_time[i]) * 60, 2))

    print()
    iterator = 1
    for average_mark in students_mark_per_min:
        print(f'{iterator} student received {average_mark}/min.')
        iterator += 1

    file_create_date = date.today().strftime('%B %d, %Y')
    file_create_time = datetime.now().strftime('%H:%M:%S')
    with io.open(r'./task7/statistics.txt', 'wt', encoding='utf-8') as statistics_txt:
        statistics_txt.write(f'File was created: {file_create_date}
{file_create_time}\n\n')

    student_answers = []
    for answers in marks:
        student_answers.append(answers[5:])

    print()
    iterator = 1
    for answer in student_answers:
        correct_answers = 0
        incorrect_answers = 0

        for i in range(0, len(answer)):
            if answer[i] == '0,50':
                correct_answers += 1
            else:
                incorrect_answers += 1

        correct_answers_percentage = round((correct_answers / len(answer)) * 100)
        incorrect_answers_percentage = round((incorrect_answers / len(answer)) * 100)
        statistics_txt.write(f'{iterator} student has {correct_answers_percentage}% of
correct answers '
                                f'and {incorrect_answers_percentage}% of incorrect
answers.\n')
        print(f'{iterator} student has {correct_answers_percentage}% of correct answers
'
              f'and {incorrect_answers_percentage}% of incorrect answers.')
        iterator += 1

    print()
    statistics_txt.write('\n')
    top_5_results = []
    students_mark_per_min_copy = students_mark_per_min[:]
    for i in range(0, 5):
        max_average_mark = max(students_mark_per_min_copy)
        top_5_results.append(max_average_mark)

    students_mark_per_min_copy.pop(students_mark_per_min_copy.index(max_average_mark))
    index_of_student = students_mark_per_min.index(top_5_results[i])
    statistics_txt.write(f'Top {i + 1}! {index_of_student + 1} student has
{top_5_results[i]}/min mark.\n')
    print(f'Top {i + 1}! {index_of_student + 1} student has {top_5_results[i]}/min
mark.')

    last_changes_date = date.today().strftime('%B %d, %Y')
    last_changes_time = datetime.now().strftime('%H:%M:%S')
    statistics_txt.write(f'\nLast changes time: {last_changes_date}
{last_changes_time}.\n')

task 7()

```

Результат програми:

		Бабушко А.С.			«Житомирська політехніка».22.121.01.000 – Лр6	Арк.
		Морозов Д.С.				14
Змн.	Арк.	№ докум.	Підпис	Дата		

TASK 7!

Total students count who wrote a test: 170

1 student received 8.0 mark.
 2 student received 8.0 mark.
 3 student received 7.0 mark.
 4 student received 8.0 mark.
 5 student received 4.0 mark.
 6 student received 8.0 mark.
 7 student received 9.5 mark.
 8 student received 10.0 mark.
 9 student received 5.5 mark.
 10 student received 10.0 mark.
 11 student received 7.5 mark.

165 student received 8.0 mark.
 166 student received 5.5 mark.
 167 student received 5.5 mark.
 168 student received 8.5 mark.
 169 student received 9.0 mark.
 170 student received 8.0 mark.

1 student received 0.52/min.
 2 student received 1.05/min.
 3 student received 0.81/min.
 4 student received 0.4/min.
 5 student received 0.38/min.
 6 student received 0.55/min.
 7 student received 0.73/min.
 8 student received 0.68/min.

165 student received 0.74/min.
 166 student received 0.27/min.
 167 student received 0.32/min.
 168 student received 0.42/min.
 169 student received 0.77/min.
 170 student received 0.56/min.

1 student has 80% of correct answers and 20% of incorrect answers.
 2 student has 80% of correct answers and 20% of incorrect answers.
 3 student has 70% of correct answers and 30% of incorrect answers.
 4 student has 80% of correct answers and 20% of incorrect answers.
 5 student has 40% of correct answers and 60% of incorrect answers.
 6 student has 80% of correct answers and 20% of incorrect answers.
 7 student has 95% of correct answers and 5% of incorrect answers.
 8 student has 100% of correct answers and 0% of incorrect answers.


```

165 student has 80% of correct answers and 20% of incorrect answers.
166 student has 55% of correct answers and 45% of incorrect answers.
167 student has 55% of correct answers and 45% of incorrect answers.
168 student has 85% of correct answers and 15% of incorrect answers.
169 student has 90% of correct answers and 10% of incorrect answers.
170 student has 80% of correct answers and 20% of incorrect answers.

Top 1! 147 student has 1.99/min mark.
Top 2! 95 student has 1.68/min mark.
Top 3! 89 student has 1.54/min mark.
Top 4! 163 student has 1.51/min mark.
Top 5! 106 student has 1.42/min mark.

Process finished with exit code 0

```

Увесь лістинг програми:

```

""" Lab 6. Python. Andrii Babushko. Repository: https://github.com/AndriiBabushko/Python
"""
import csv
import os
import io
from datetime import datetime
from datetime import date
from time import perf_counter

# task 1
"""
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 файлу і обчислює їх суму, виводить цю
суму на екран і, водночас,
записує цю суму у інший файл з назвою sum_numbers.txt.
"""

def task_1():
    print('\nTASK 1!')
    if not os.path.isdir(r'./task1'):
        os.mkdir(r'./task1')

    with io.open(r'./task1/numbers.txt', 'wt', encoding='utf-8') as numbers_txt:
        numbers_txt.write(create_numbers_string())

    with io.open(r'./task1/numbers.txt', 'rt', encoding='utf-8') as numbers_txt:
        task_1_sum = 0
        task_1_number_list = []

        for number in numbers_txt:
            try:
                task_1_sum += int(number)
                task_1_number_list.append(int(number))
            except ValueError as err:
                print(f'ERROR! {ValueError}')

        with io.open(r'./task1/sum_numbers.txt', 'wt', encoding='utf-8') as
sum_numbers_txt:
            sum_numbers_txt.write(str(task_1_sum))

```

		Бабушко А.С.			«Житомирська політехніка».22.121.01.000 – Лр6	Арк.
		Морозов Д.С.				16
Змн.	Арк.	№ докум.	Підпис	Дата		


```

print(f'Received numbers from file: {task_1_number_list}')
print(f'Sum of numbers in file = {task_1_sum}')

def create_numbers_string():
    from random import randint
    new_string = ''

    for new_number in range(10):
        new_string += f'{randint(-100, 100)}\n'

    return new_string

task_1()

# task 2
"""
    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.
"""

def task_2():
    print('\nTASK 2!')
    task_2_enter_count = int(input('Enter how much numbers you want to enter: '))
    task_2_number_list = task_2_enter_some_numbers(task_2_enter_count)

    if not os.path.isdir(r'./task2'):
        os.mkdir(r'./task2')

    with io.open(r'./task2/even_or_odd_numbers.txt', 'wt', encoding='utf-8') as
even_or_odd_numbers_txt:
        for number in task_2_number_list:
            if number % 2 == 0:
                even_or_odd_numbers_txt.write(f'{number} is even number!\n')
            else:
                even_or_odd_numbers_txt.write(f'{number} is odd number!\n')

    with io.open(r'./task2/even_or_odd_numbers.txt', 'rt', encoding='utf-8') as
even_or_odd_numbers_txt:
        for line in even_or_odd_numbers_txt:
            print(line, end="")

def task_2_enter_some_numbers(count):
    numbers_list = []

    while count != 0:
        try:
            int_number = int(input('Enter value: '))
            numbers_list.append(int_number)
            count -= 1
        except ValueError as value_error:
            numbers_list.pop()
            print('ERROR:', value_error)

    return numbers_list

task_2()

# task 3
"""
    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 ...» .
"""

```

		Бабушко А.С.			«Житомирська політехніка».22.121.01.000 – Лр6	Арк.
		Морозов Д.С.				17
Змн.	Арк.	№ докум.	Підпис	Дата		

```

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.
"""

def task_3():
    print('\nTASK 3!')

    if not os.path.isdir(r'./task3'):
        os.mkdir('./task3')

    with io.open(r'./task3/learning_python.txt', 'wt', encoding='utf-8') as
learning_python_txt:
        learning_python_txt.writelines(
            ["Python можна використати для machine learning.\n",
            "Python можна використати для написання простих скриптів.\n",
            "Python можна використати для програмування веб додатків.\n",
            "Python можна використати для написання особистого телеграм боту.\n"])

    with io.open(r'./task3/learning_python.txt', 'rt', encoding='utf-8') as
learning_python_txt:
        python_string_list = []

        for line in learning_python_txt:
            python_string_list.append(line)

        python_string_sorted_list = python_string_list[:]
        python_string_sorted_list.sort(key=len)
        print('\nSorted list:')
        for string in python_string_sorted_list:
            print(string, end="")

        python_string_reversed_sorted_list = python_string_list[:]
        python_string_reversed_sorted_list.sort(key=len, reverse=True)
        print('\nReversed sorted list:')
        for string in python_string_reversed_sorted_list:
            print(string, end="")

task_3()

# task 4
"""
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 C, а які ні. Всі хибні твердження
запишіть в інший файл, а істинні
- в поточний.
"""

def task_4():
    print('\nTASK 4!')

    if not os.path.isdir(r'./task4'):
        os.mkdir(r'./task4')

    with io.open(r'./task3/learning_python.txt', 'rt', encoding='utf-8') as
learning_python_txt:
        python_strings = []

        for string in learning_python_txt:

```

		Бабушко А.С.			«Житомирська політехніка».22.121.01.000 – Лр6	Арк.
		Морозов Д.С.				18
Змн.	Арк.	№ докум.	Підпис	Дата		

```

python_strings.append(string.split(' '))

for i in range(0, len(python_strings)):
    python_strings[i][0] = 'C'
    python_strings[i] = " ".join(python_strings[i])

with io.open(r'./task4/learning_c.txt', 'wt', encoding='utf-8') as true_about_c_txt:
    for c_string in python_strings:
        true_about_c_txt.write(c_string)

if os.path.exists(r'./task4/true_about_c.txt'):
    os.remove(r'./task4/true_about_c.txt')
if os.path.exists(r'./task4/false_about_c.txt'):
    os.remove(r'./task4/false_about_c.txt')

with io.open(r'./task4/learning_c.txt', 'rt', encoding='utf-8') as learning_c_txt:
    for c_string in learning_c_txt:
        print(c_string, end='')
        answer = enter_true_or_false()
        if answer:
            with io.open(r'./task4/true_about_c.txt', 'at', encoding='utf-8') as
true_about_c_txt:
                true_about_c_txt.write(c_string)
            else:
                with io.open(r'./task4/false_about_c.txt', 'at', encoding='utf-8') as
false_about_c_txt:
                    false_about_c_txt.write(c_string)

    print(f'True statements about C:')
    with io.open(r'./task4/true_about_c.txt', 'rt', encoding='utf-8') as
true_about_c_txt_read:
        for c_string in true_about_c_txt_read:
            print(c_string, end='')

    print(f'False statements about C:')
    with io.open(r'./task4/false_about_c.txt', 'rt', encoding='utf-8') as
false_about_c_txt_read:
        for c_string in false_about_c_txt_read:
            print(c_string, end='')

def enter_true_or_false():
    while True:
        try:
            answer = input("Enter 'true' or 'false' about this statement: ")
            if answer != 'true' and answer != 'false':
                raise ValueError('Value was entered incorrectly!')
            else:
                pass
                if answer == 'true':
                    return True
                elif answer == 'false':
                    return False
        except ValueError as value_err:
            print(f'ERROR! {value_err}')

task_4()

# task 5
"""
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

```

		Бабушко А.С.			«Житомирська політехніка».22.121.01.000 – Лр6	Арк.
		Морозов Д.С.				19
Змн.	Арк.	№ докум.	Підпис	Дата		

```

        вказання в ньому часу останніх внесених змін
        """

def task_5():
    print('\nTASK 5!')

    if not os.path.isdir(r'./task5'):
        os.mkdir(r'./task5')

    task_5_count = enter_count()
    file_create_time = date.today().strftime('%B %d, %Y')

    with io.open(r'./task5/guest_book.txt', 'wt', encoding='utf-8') as guest_book_txt:
        guest_book_txt.write(f'File was created: {file_create_time}\n\n')

        for counter in range(0, task_5_count):
            name_greeting = f'{counter + 1}) Hello, ' + enter_name() + '! '
            message_time = datetime.now().strftime('%H:%M:%S')
            name_greeting += f'Name was entered in time: {message_time}.\n'
            print(name_greeting, end="")
            guest_book_txt.write(name_greeting)

        last_changes_date = date.today().strftime('%B %d, %Y')
        last_changes_time = datetime.now().strftime('%H:%M:%S')
        guest_book_txt.write(f'\nLast changes time: {last_changes_date}
{last_changes_time}.\n')

def enter_name():
    while True:
        try:
            name = input('Enter name: ')
            if len(name) < 2:
                raise ValueError('Name\'s length is less than 2!')
            else:
                pass
            return name
        except ValueError as value_err:
            print(f'ERROR! {value_err}')

def enter_count():
    while True:
        try:
            count = int(input('Enter count of names: '))
            if count <= 0:
                raise ValueError('Count names was entered incorrectly!')
            else:
                pass
            return count
        except ValueError as value_err:
            print(f'ERROR! {value_err}')

task_5()

# task 6
"""
    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 роботи 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.
    """

```

		Бабушко А.С.			«Житомирська політехніка».22.121.01.000 – Лр6	Арк.
		Морозов Д.С.				20
Змн.	Арк.	№ докум.	Підпис	Дата		

```

def task_6():
    import re
    print('\nTASK 6!')

    with io.open(r'./task6/post_about_python.txt', 'rt', encoding='utf-8') as
post_about_python_txt:
        words = re.split(r' |\n| |? |\.\|, |\: |\;|', post_about_python_txt.read().lower())

        file_create_date = date.today().strftime('%B %d, %Y')
        file_create_time = datetime.now().strftime('%H:%M:%S')
        with io.open(r'./task6/word_frequency.txt', 'wt', encoding='utf-8') as
word_frequency_txt:
            word_frequency_txt.write(f'File was created: {file_create_date}
{file_create_time}\n\n')
            print(f'File was created: {file_create_time}\n')

            words_without_duplicates = []
            for string in words:
                if string not in words_without_duplicates:
                    words_without_duplicates.append(string)

            for i in range(0, len(words_without_duplicates)):
                time_started = perf_counter()
                counter = 0
                for j in range(0, len(words)):
                    if words_without_duplicates[i] == words[j]:
                        counter += 1
                time_finished = perf_counter()
                write_time = round(time_finished - time_started, 8)
                word_frequency_txt.write(f'"{words[i]}" repeats "{counter}" times. Write time:
{write_time}.\n')
                print(f'"{words[i]}" repeats "{counter}" times. Write time: {write_time}.')

            last_changes_date = date.today().strftime('%B %d, %Y')
            last_changes_time = datetime.now().strftime('%H:%M:%S')
            word_frequency_txt.write(f'\nLast changes time: {last_changes_date}
{last_changes_time}.\n')
            print(f'\nLast changes time: {last_changes_date} {last_changes_time}.')

task_6()

# task 7
"""
    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 отримували 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.
"""

def task_7():
    print('\nTASK 7!')

    if not os.path.isdir(r'./task7'):
        os.mkdir(r'./task7')

    with io.open(r'./task7/marks.lab6.csv', 'rt', encoding='utf-8') as marks_lab6_csv:
        marks_csv = csv.reader(marks_lab6_csv)
        marks = [line for line in marks_csv]

```

		Бабушко А.С.			«Житомирська політехніка».22.121.01.000 – Лр6	Арк.
		Морозов Д.С.				21
Змн.	Арк.	№ докум.	Підпис	Дата		

```

students_count = len(marks)
print(f'Total students count who wrote a test: {students_count}')

students_marks = []
for i in range(students_count):
    mark_string = marks[i][4].split(',')
    mark_number = float(mark_string[0]) + float(int(mark_string[1]) / 100)
    students_marks.append(mark_number)

print()
iterator = 1
for mark in students_marks:
    print(f'{iterator} student received {mark} mark.')
    iterator += 1

students_time = []
for i in range(students_count):
    time = marks[i][3].split(' ')
    if len(time) > 2:
        minutes = int(time[0])
        seconds = int(time[2]) + minutes * 60
    else:
        minutes = int(time[0])
        seconds = minutes * 60
    students_time.append(seconds)

students_mark_per_min = []
for i in range(students_count):
    students_mark_per_min.append(round((students_marks[i] / students_time[i]) * 60, 2))

print()
iterator = 1
for average_mark in students_mark_per_min:
    print(f'{iterator} student received {average_mark}/min.')
    iterator += 1

file_create_date = date.today().strftime('%B %d, %Y')
file_create_time = datetime.now().strftime('%H:%M:%S')
with io.open(r'./task7/statistics.txt', 'wt', encoding='utf-8') as statistics_txt:
    statistics_txt.write(f'File was created: {file_create_date}
{file_create_time}\n\n')

student_answers = []
for answers in marks:
    student_answers.append(answers[5:])

print()
iterator = 1
for answer in student_answers:
    correct_answers = 0
    incorrect_answers = 0

    for i in range(0, len(answer)):
        if answer[i] == '0,50':
            correct_answers += 1
        else:
            incorrect_answers += 1

    correct_answers_percentage = round((correct_answers / len(answer)) * 100)
    incorrect_answers_percentage = round((incorrect_answers / len(answer)) * 100)
    statistics_txt.write(f'{iterator} student has {correct_answers_percentage}% of
correct answers '
                        f'and {incorrect_answers_percentage}% of incorrect
answers.\n')
    print(f'{iterator} student has {correct_answers_percentage}% of correct answers
'
          f'and {incorrect_answers_percentage}% of incorrect answers.')
    iterator += 1

```

		Бабушко А.С.			«Житомирська політехніка».22.121.01.000 – Лр6	Арк.
		Морозов Д.С.				22
Змн.	Арк.	№ докум.	Підпис	Дата		

```

print()
statistics_txt.write('\n')
top_5_results = []
students_mark_per_min_copy = students_mark_per_min[:]
for i in range(0, 5):
    max_average_mark = max(students_mark_per_min_copy)
    top_5_results.append(max_average_mark)

students_mark_per_min_copy.pop(students_mark_per_min_copy.index(max_average_mark))
index_of_student = students_mark_per_min.index(top_5_results[i])
statistics_txt.write(f'Top {i + 1}! {index_of_student + 1} student has
{top_5_results[i]}/min mark.\n')
print(f'Top {i + 1}! {index_of_student + 1} student has {top_5_results[i]}/min
mark.')

last_changes_date = date.today().strftime('%B %d, %Y')
last_changes_time = datetime.now().strftime('%H:%M:%S')
statistics_txt.write(f'\nLast changes time: {last_changes_date}
{last_changes_time}.\n')

task_7()

```

Висновок: під час виконання лабораторної роботи було отримано навички роботи з текстовими та csv файлами на мові Python.

		Бабушко А.С.			«Житомирська політехніка».22.121.01.000 – Лр6	Арк.
		Морозов Д.С.				23
Змн.	Арк.	№ докум.	Підпис	Дата		