

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра конструювання електронно обчислювальної апаратури

**Лабораторна Робота 1**

з дисципліни

«Високорівнева розробка програмного забезпечення»

Виконала студентка:

Харчук К. М.

Навчальної групи: ФЕЛ, ДК-21

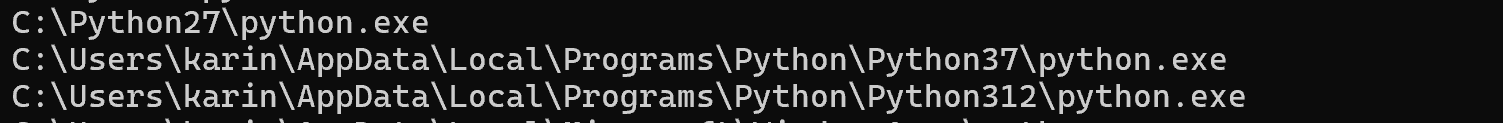
Перевірив: Мушта І. А.

                                                         Київ 2024

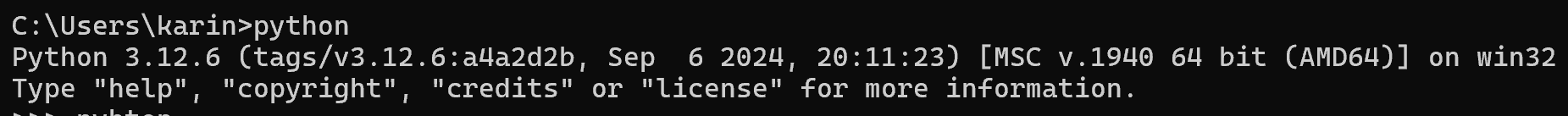
**Завдання**

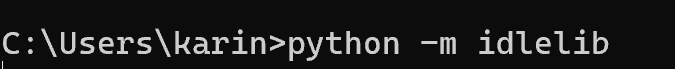
#### Частина 1: Установка Python

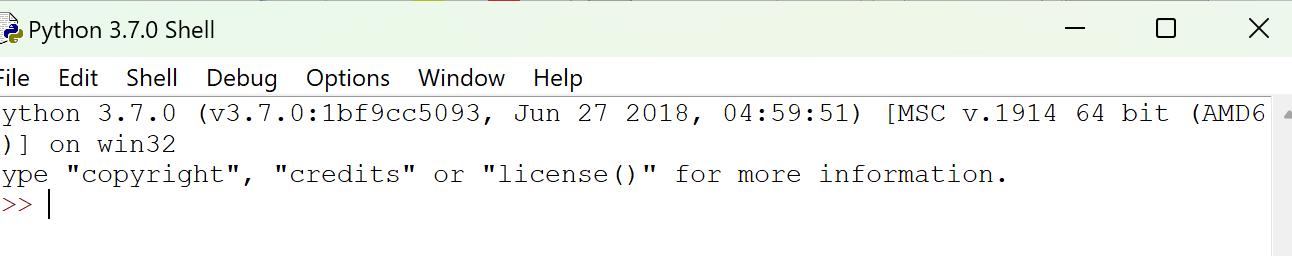
1. **Встановіть три версії Python**:
   * Python 3.12
   * Python 2.x (будь-яка версія)
   * Python 3.x (будь-яка версія, крім 3.12)



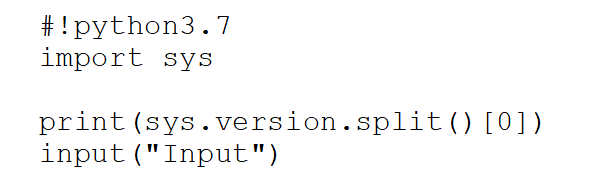
1. **Вивчіть варіанти запуску Python**:
   * Через командний рядок (команди python, py)



* + З текстового редактора або IDE (наприклад, IDLE) 

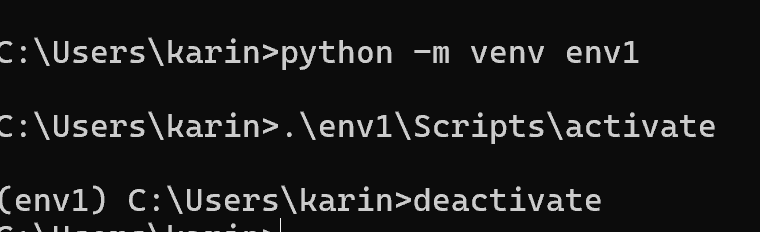


* + Використовуючи shebang (рядок виду #!/usr/bin/python3).



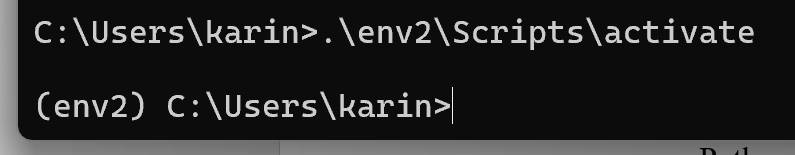
Частина 2: Віртуальні середовища

1. **Встановіть наступні віртуальні середовища** для роботи з різними версіями Python:
   * Venv

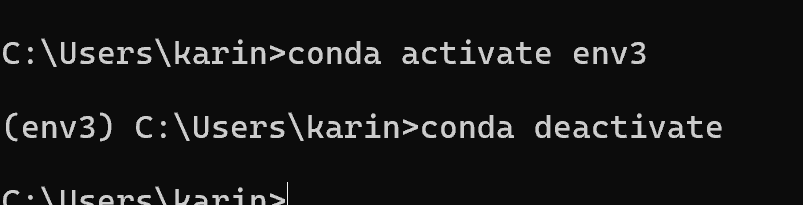


* + Virtualenv





* + Conda



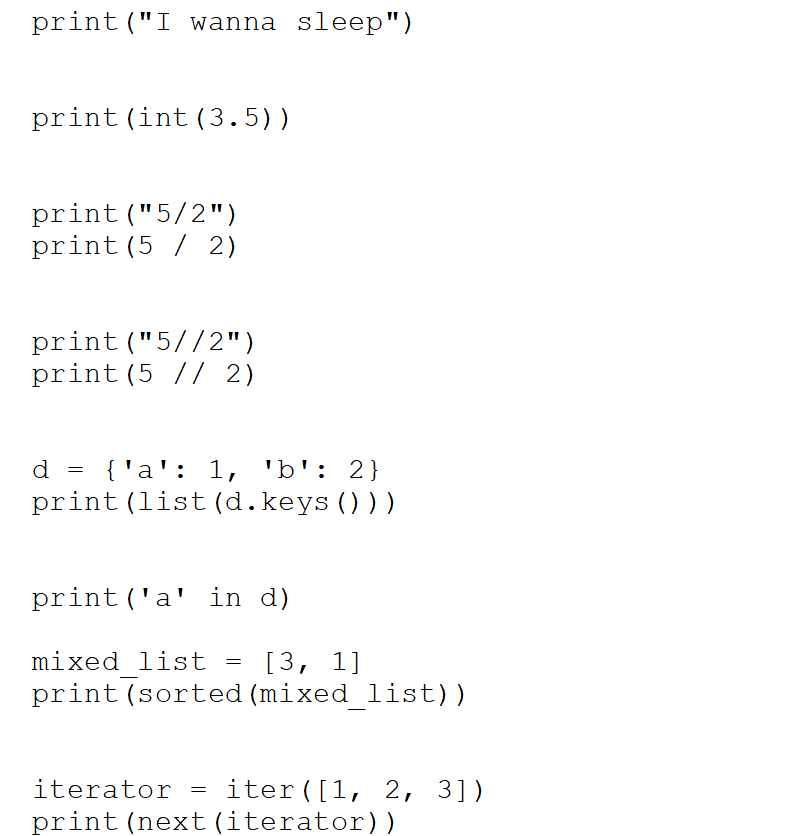
**Створіть одне віртуальне середовище** з Python 2.x, інше з Python 3.x.

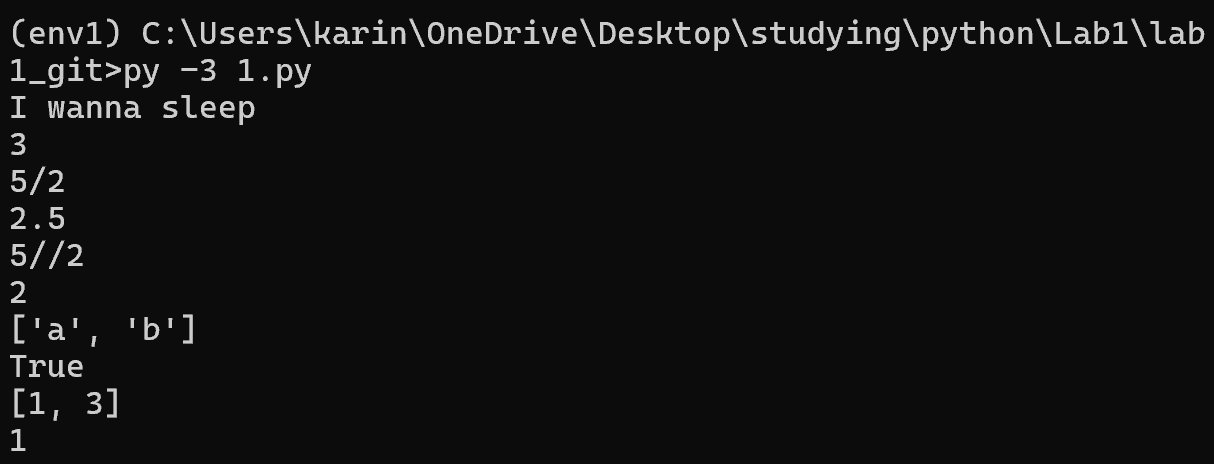
1. **Напишіть код** для демонстрації 10 основних відмінностей між Python 2 і Python 3.
   * Використовуйте для цього створені віртуальні середовища з Python 2.x та та Python 3.х.

Для використання print у третій версії потрібні (), у другій версії – ні

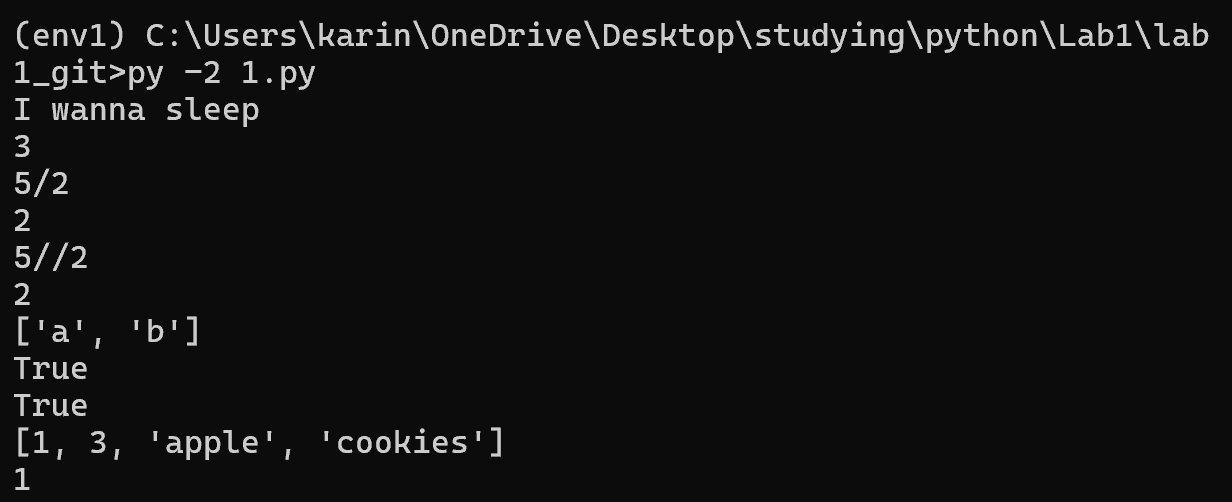
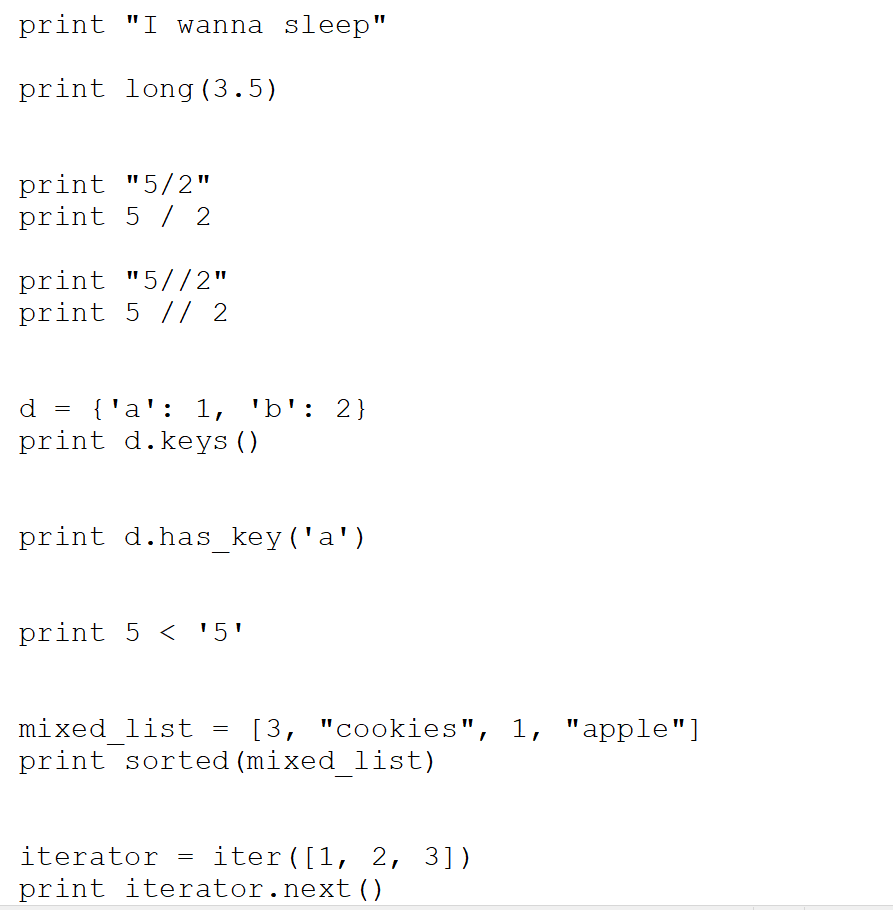
|  |  |
| --- | --- |
| Python 2 | Python 3 |
| Print без () | Для використання print потрібні () |
| Для великих чисел потрібно юзати long | Не потрібно long, бо int підтримує великі числа |
| Оператор / повертає ціле число для цілих чисел | Оператор / повертає число з плаваючою комою |
| Має has\_key() | Не має has\_key(), замість оператор in |
| Функція next() викликається як метод .next() | Окрема функція next() |
| Можна порівнювати числа та рядки | Порівнює тільки змінні одного типу |
| Можна сортувати списки, які містять різні типи змінних | Не можна сортувати списки, які містять різні типи змінних |
| d.keys() повертає список ключів | d.keys() повертає ітератор, тому для перетворення на список треба викликати list(d.keys()) |

**Python 3:**



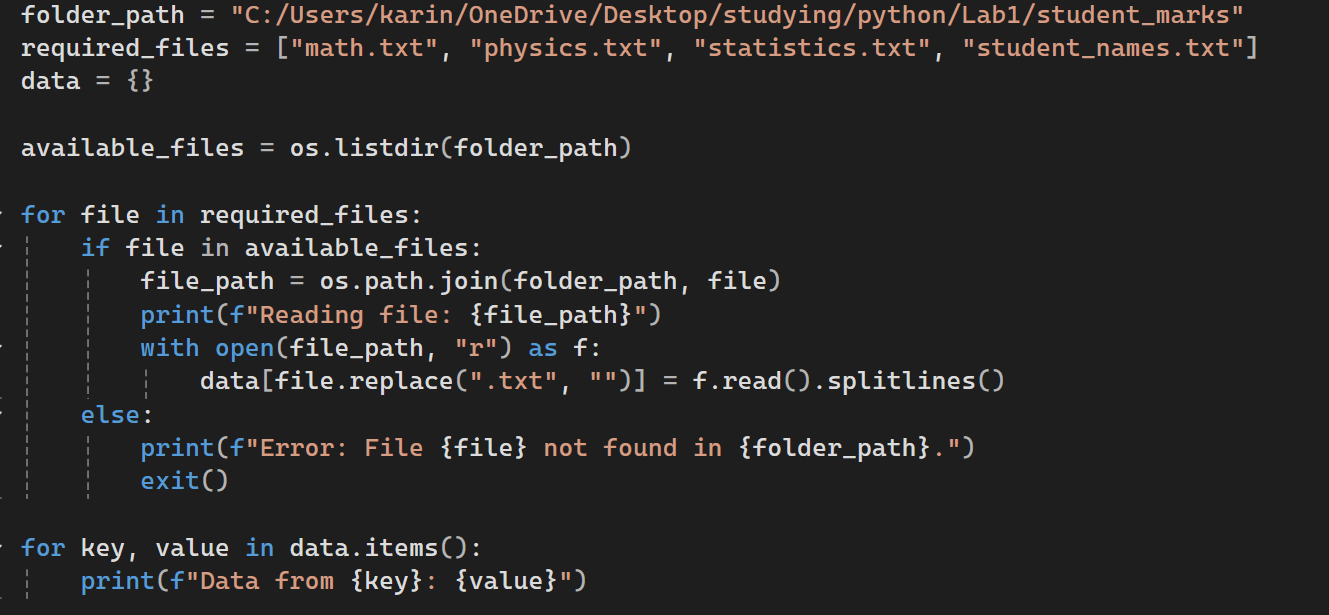


**Python 2**



Частина 3: Списки, словники та цикли

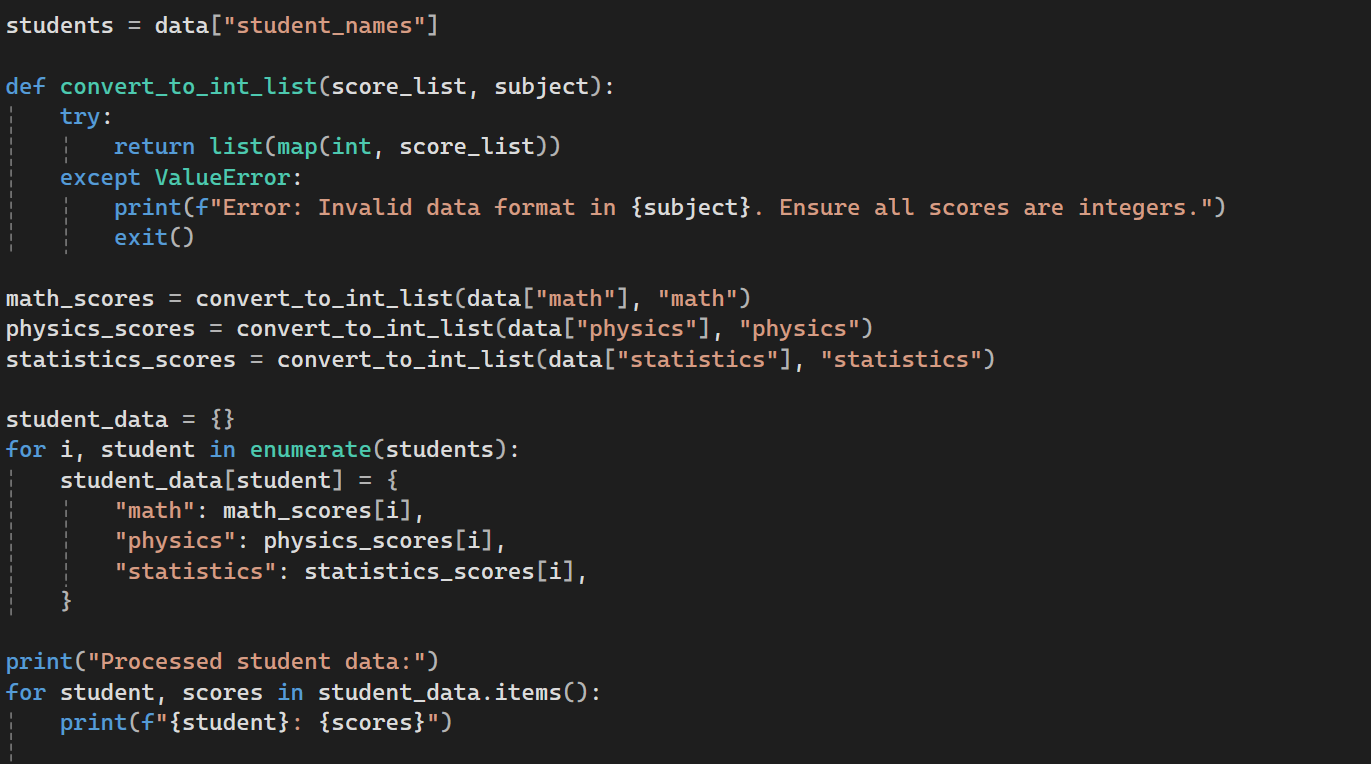
1. **Зчитати файли**:
   * Використовуючи os.listdir(), пройдіться по файлах math.txt, statistics.txt, physics.txt, та student\_names.txt.
   * Перевірте, що всі файли існують, та відкрийте їх для зчитування.



Прописуємо шлях до папки де наш файл лежить. Визначаємо які саме файли потрібні і робимо пустий словник, де буде збережено дані з цих файлів. Функція os.listdir(folder\_path) повертає список файлів, які є в директорії folder\_path. За допомогою циклу і іфу в ньому перевіряємо чи є файл у списку доступних файлів, створюємо до нього шлях, після чого відкриваємо цей файл у режимі читання і додаємо його до словника. Якщо файл не знайдено, то виведеться помилка і ми вийдемо з процесу.

Наступним фором виводимо вміст кожного файлу.  


1. **Обробка даних**:
   * Зчитайте оцінки з кожного файлу, перетворивши їх у список цілих чисел для кожного предмета. Також створіть список з іменами студентів.
   * Створіть словник, який містить оцінки для кожного студента за всі три предмети. Використовуйте ім'я студента як ключ, а словник з предметами – як значення. Словник з предметами використовує назву предмету як ключ, а оцінку - як значення.



Перший рядок зчитує список імен студентів зі словника data та зберігає в змінну students. Далі оголошується функція, яка перетворює список оцінок у число для заданого предмету. У блоці try перетворюємо кожен елемент списку score\_list на ціле число (return повертає список цілих чисел, які зробила функція map).

Виклик функції convert\_to\_int\_list перетворює дані оцінок із файлу для відповідного предмета (math, physics, statistics) на список цілих чисел.

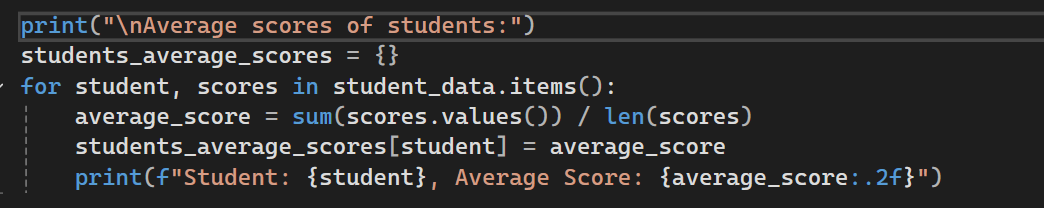
Ініціалізуємо порожній словник, де будуть оцінки кожного студента. Перший цикл проходить по кожному студенту та його індексу у списку students. Для кожного студента створюється словник із його оцінками і туди вже додаються оцінки по предметах.

Виводимо повідомлення про завершення обробки даних і наступним циклом проходимося по кожному студенту та його оцінкам зі словника і воводимо ці дані.

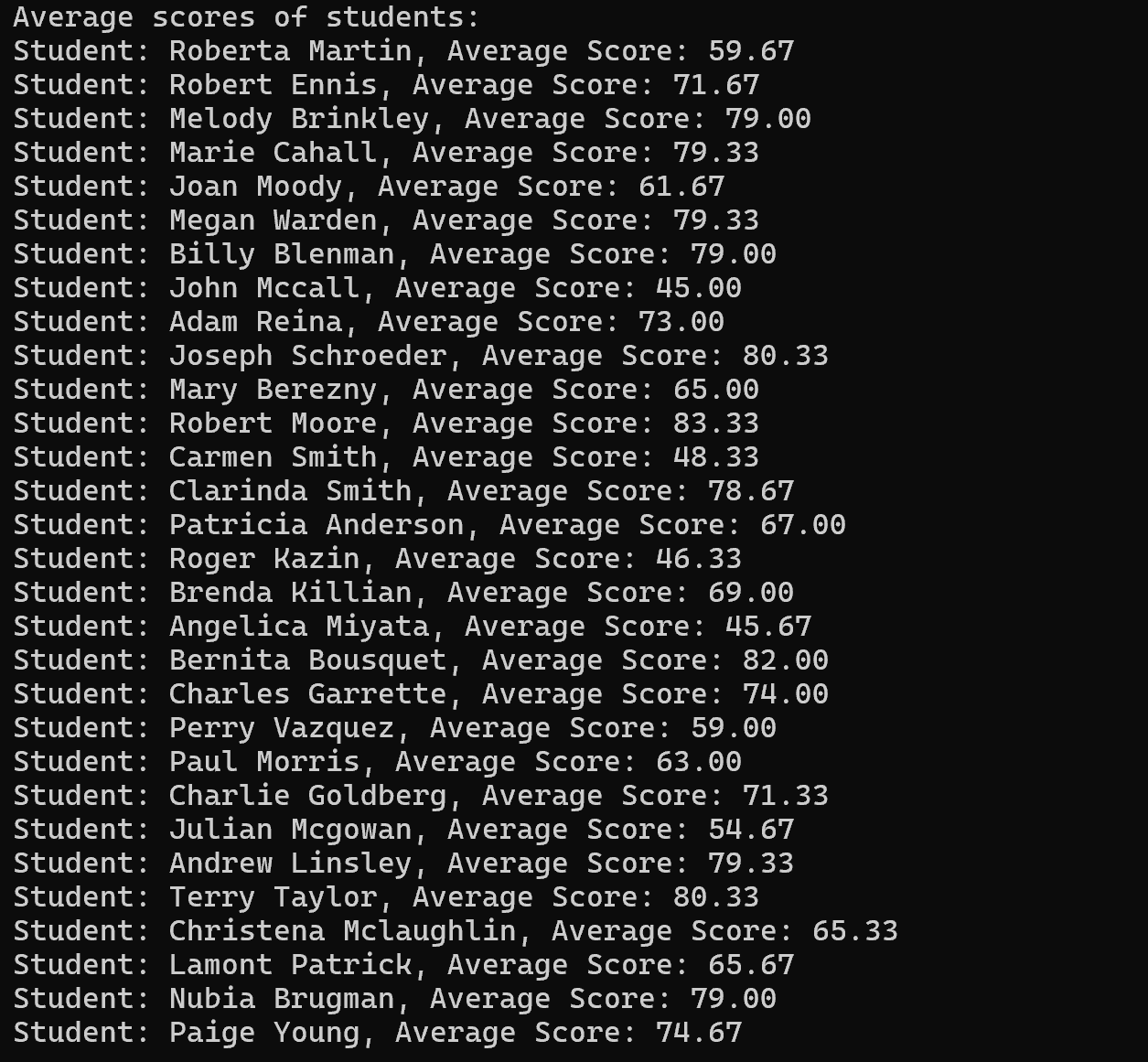


1. **Виведіть результати**:
   * Виведіть статистику для кожного студента: його ім'я та середню оцінку по всіх трьох предметах.

Приклад виводу:   
*студент: … , середня оцінка: …*

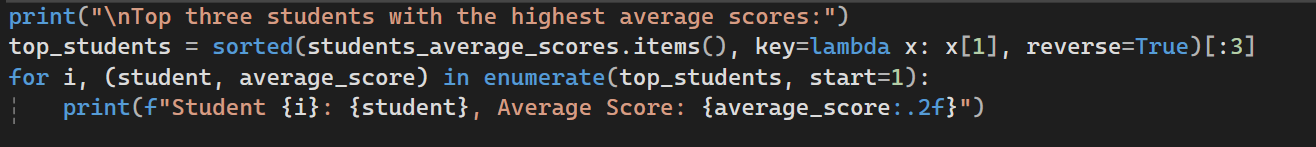
**

Спочатку ініціалізуємо порожній масив, де будуть зберігатися середні оцінки. У циклі проходимо по кожному студенту та його оцінках (student\_data). Обчислюється середня оцінка (сума оцінок поточного студента ділиться на кількість предметів). Результат додається до словника students\_average\_scores. Оцінка виводиться з двома числами після коми.

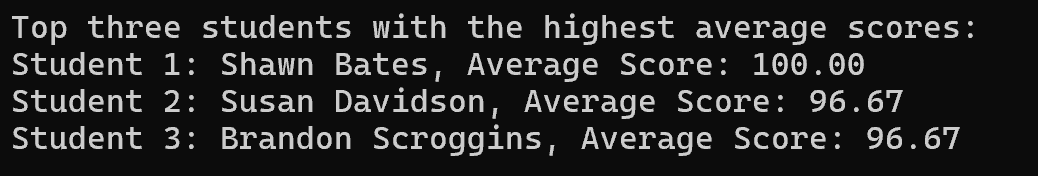


* + Знайдіть трьох студентів з найвищими середніми оцінками.

Приклад виводу: *Студент 1  
Студент 2  
Студент 3*

**

Сортуються студенти за середніми оцінками від найвищої до найнижчої і вибираються три перші студенти. Цикл проходиться по цих трьох студентах і додає індексацію їм. Далі виводимо порядковий номер студента, ім’я та оцінку.

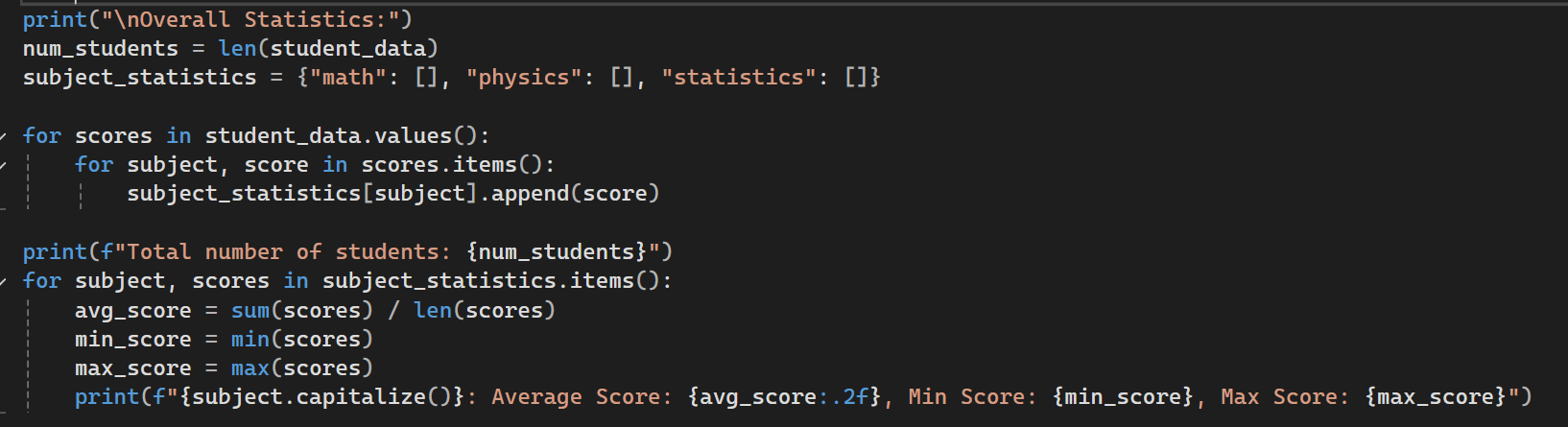


* + Підрахуйте загальну кількість студентів, середні оцінки по кожному предмету, максимальні та мінімальні оцінки.

Приклад виводу:

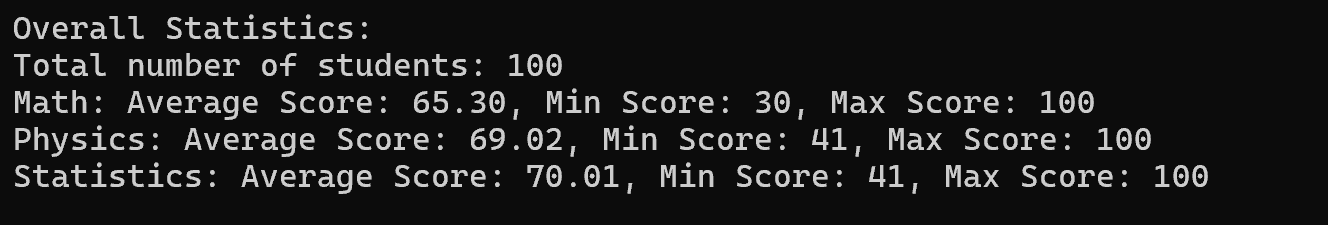
*Загальна кількість студентів: …*

*предмет: … , середня оцінка … , min оцінка: … , max оцінка: …*

**

Отримуємо довжину словника student\_data, щоб знати кількість студентів. Ініціалізуємо словник, що зберігатиме всі оцінки для кожного предмета. Зовнішнім циклом проходимося по всіх оцінках, а внутрішнім по кожному предмету і його оцінці. Оцінка додається до відповідного списку в словнику subject\_statistics.

У for’і розраховується середня оцінка для кожного предмета, знаходиться мінімальна та максимальна оцінка по предмету і ці дані виводимо.

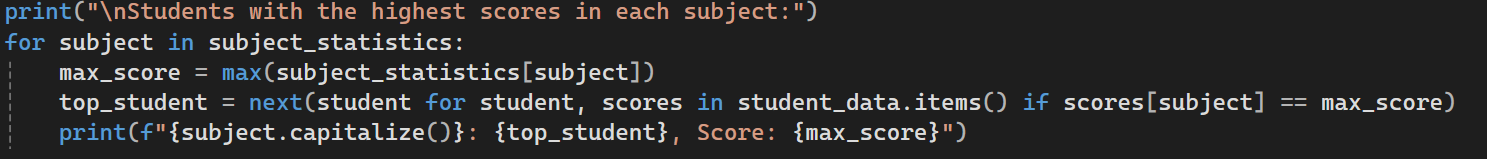


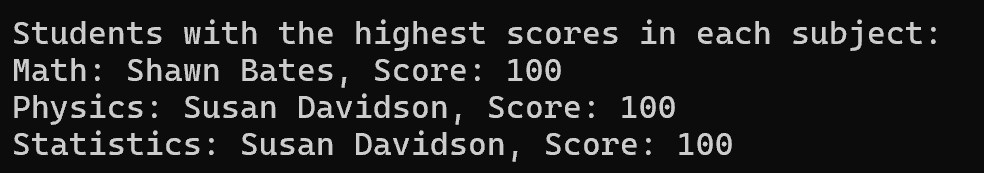
* + Для кожного предмету виведіть ім'я студента, який отримав найвищий бал, а також оцінку

Приклад виводу:  
*Предмет 1: … | студент: … , оцінка: …*

*Предмет 2: … | студент: … , оцінка: …*

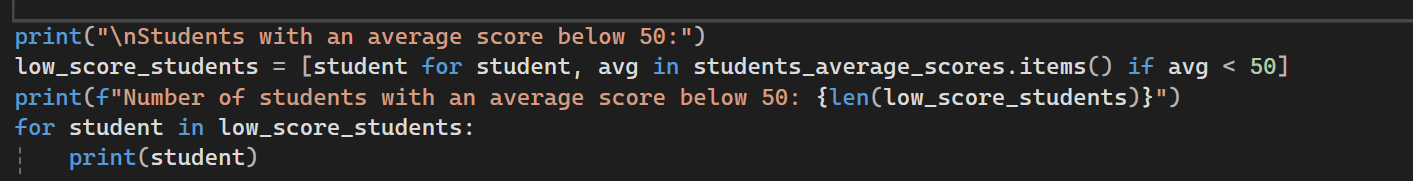
*Предмет 3: … | студент: … , оцінка: …*

**Цикл проходиться по предметах, знаходить максимальну оцінку для поточного предмету і знаходить ім’я студента з цією оцінкою. Виводимо результат

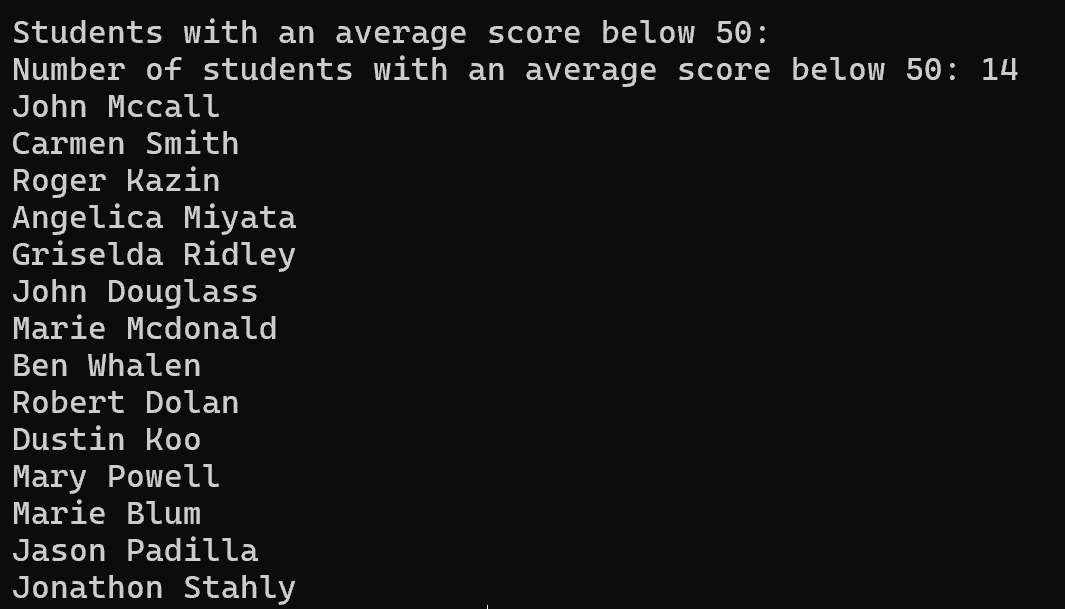
**

* + Виведіть загальну кількість студентів, які мають середню оцінку нижче 50, а також список таких студентів

Приклад виводу:  
*Кількість студентів з оцінкою нижче 50: …  
Студент 1  
Студент 2  
Студент 3  
…*

**

Створюється список студентів low\_score\_students у яких бал менший за 50 і виводимо їхню кількість. Фором проходимося по цьому списку і виводимо ім’я кожного.

**