МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Український державний університет імені Михайла Драгоманова

Факультет математики, інформатики та фізики

Кафедра інформаційних технологій та програмування

**Звіт**

з лабораторної роботи №1

«Лінійні алгоритми мовою Python»

з дисципліни «Програмування»

Виконав:

студент ІІІ курсу групи 31І

Гладкович Я.В

Перевірила:

викладач Устименко О.Б.

Оцінка\_\_\_\_\_\_\_\_\_\_\_

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_

Київ - 2023

**Зміст**

[Мета роботи 3](#_Toc150170352)

[1 Постановка задачі 4](#_Toc150170353)

[2 Основна частина 5](#_Toc150170354)

[2.1 Опис вхідних та вихідних даних 5](#_Toc150170355)

[2.1 Блок-схема 7](#_Toc150170356)

[Висновки 13](#_Toc150170357)

[Список літератури 16](#_Toc150170358)

[Додатки 17](#_Toc150170359)

# Мета роботи

Мета лабораторної роботи – скласти програми мовою Python для вирішення задач.

# 1 Постановка задачі

1. Напишіть програму для перетворення введених секунд у кількість днів, годин, хвилин та секунд.
2. Дано два моменти часу в межах однієї доби. Для кожного моменту вказано години, хвилини і секунди. Відомо, що другий момент часу настав не раніше першого. Визначте скільки секунд пройшло між двома моментами часу. Програма на вхід отримує шість цілих чисел в окремих рядках. Перші три цілих числа відповідають годинам, хвилинам і секундам першого моменту, наступні три числа відповідають другому моменту. Години задаються числом від 0 до 23 включно. Хвилини і секунди - від 0 до 59.
3. Напишіть програму для друку літери Ж висотою 5 рядків за допомогою введеного користувачем символу.
4. Напишіть програму, щоб у введеному користувачем рядку виконати обмін місцями першого та останнього символів.

# 

# 2 Основна частина

## 2.1 Опис вхідних та вихідних даних

***Завдання №1***

Команди,які були використані для цього завдання:

1. **input("Уведіть число: "):** Запит користувача ввести число з клавіатури
2. **float():** Перетворення введеного користувачем рядка на число з плаваючою точкою
3. **//:** Оператор цілочисельного ділення, який повертає цілу частину від ділення.
4. **%:** Оператор залишку від ділення, який повертає залишок від цілочисельного ділення.
5. **print():** Вивід результату на екран.

Програма працює наступним чином:

**Введення користувача:**

час = float(input("Уведіть число: ")): Користувач вводить кількість секунд у змінну час.

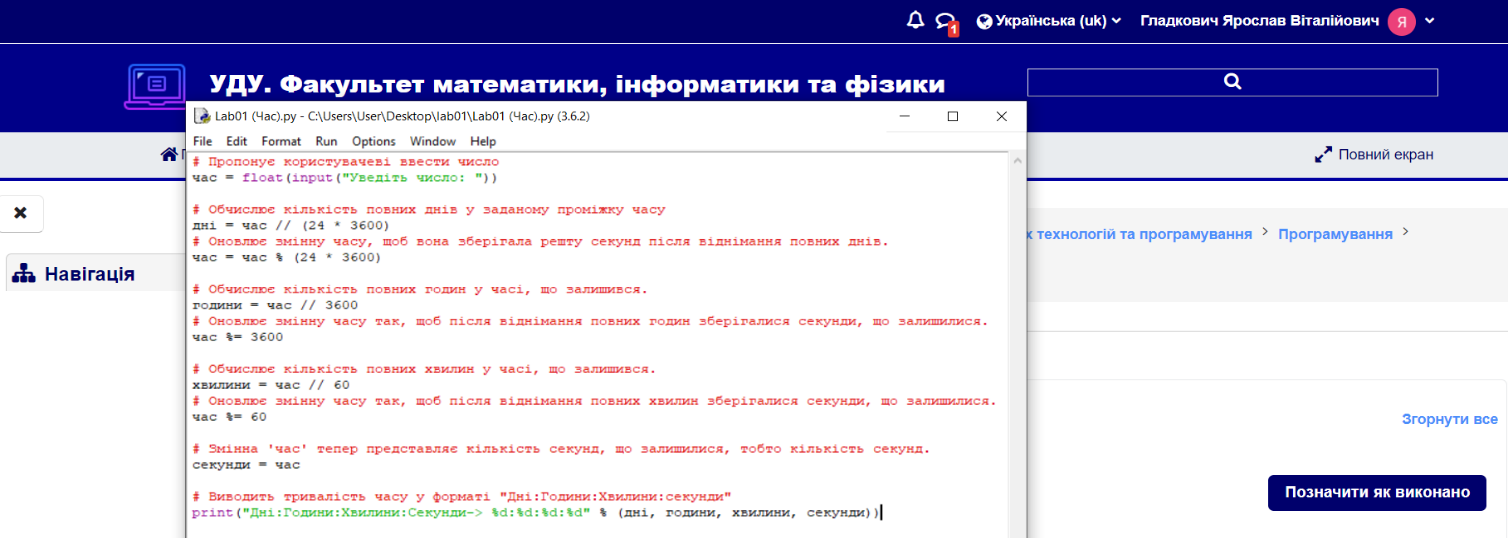
**Обчислення днів:** дні = час // (24 \* 3600): Розділяємо загальну кількість секунд на кількість секунд у добі, отримуючи повні дні. час = час % (24 \* 3600): Оновлюємо змінну час, щоб вона містила залишкові секунди після віднімання повних днів.

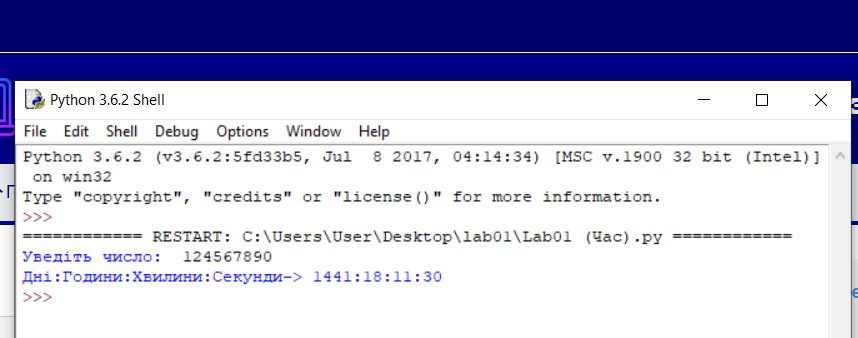
**Обчислення годин:** години = час // 3600: Розділяємо залишковий час на кількість секунд у годині, отримуючи повні години. час %= 3600: Оновлюємо змінну час, щоб вона містила залишкові секунди після віднімання повних годин.

**Обчислення хвилин:** хвилини = час // 60: Розділяємо залишковий час на кількість секунд у хвилині, отримуючи повні хвилини. час %= 60: Оновлюємо змінну час, щоб вона містила залишкові секунди після віднімання повних хвилин.

**Останні секунди:** секунди = час: Змінна час тепер містить залишкові секунди.

**Вивід результату:** print("Дні:Години:Хвилини:Секунди-> %d:%d:%d:%d" % (дні, години, хвилини, секунди)): Виводимо результат у зручному форматі "Дні:Години:Хвилини:Секунди".





## 

***Завдання №2***

Команди, використані для цієї програми:

**1. `int(input())`:** Запит користувача ввести ціле число з клавіатури та перетворення його на ціле число.

**2. `\*`:** Оператор множення.

**3. `+`:** Оператор додавання.

**4. `-`:** Оператор віднімання.

**5. `print()`:** Вивід результату на екран.

1. Введення часу для першого моменту:

години1 = int(input())

хвилини1 = int(input())

секунди1 = int(input())

Користувач вводить години, хвилини і секунди для першого моменту часу.

2. Введення часу для другого моменту:

години2 = int(input())

хвилини2 = int(input())

секунди2 = int(input())

Користувач вводить години, хвилини і секунди для другого моменту часу.

3. Підрахунок кількості секунд для обох моментів:

всього\_секунд1 = години1 \* 3600 + хвилини1 \* 60 + секунди1

всього\_секунд2 = години2 \* 3600 + хвилини2 \* 60 + секунди2

Кількість секунд для обох моментів обчислюється шляхом перетворення годин, хвилин і секунд у секунди і їх додавання.

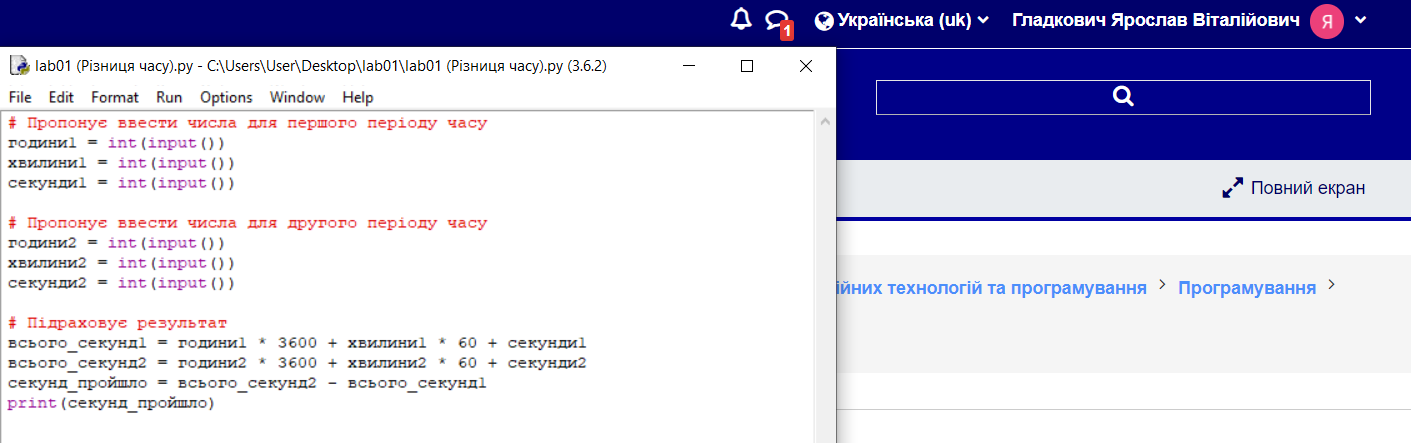
4. Визначення різниці між двома моментами:

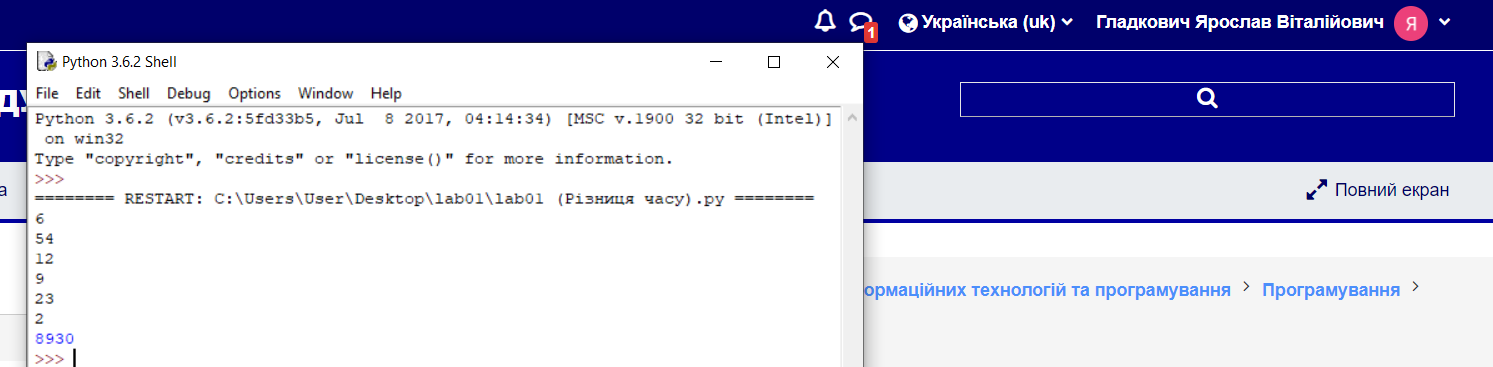
секунд\_пройшло = всього\_секунд2 - всього\_секунд1

Різниця між кількістю секунд для другого та першого моментів визначається.

5. Виведення результату:

print(секунд\_пройшло) Кількість пройдених секунд виводиться на екран.





***Завдання №3***

Команди, використані для цієї програми:

**1. `input()`:** Запит користувача ввести символ з клавіатури.

**2. `print()`:** Вивід результату на екран.

**3. `f-string`:** Форматування рядка, що дозволяє вставляти значення змінних прямо в рядок.

1. Введення символу користувачем:

s = input("Уведіть символ: ")

Користувач вводить символ, який буде використовуватися для створення літери "Ж".

2. Виведення літери «Ж»:

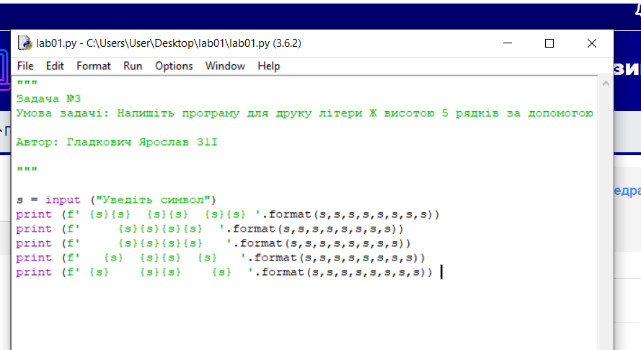
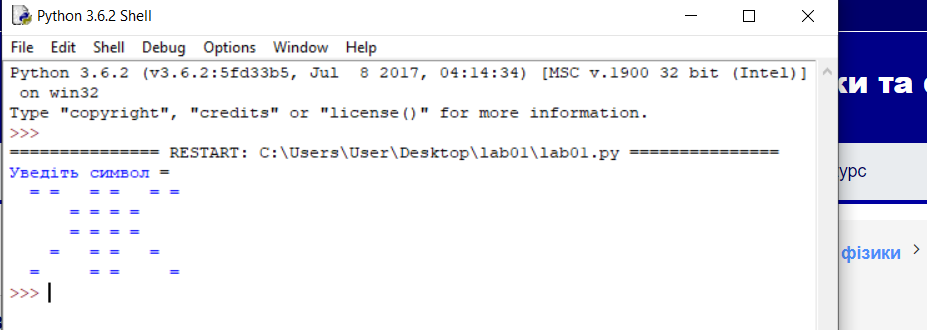
print (f' {s}{s} {s}{s} {s}{s} '.format(s,s,s,s,s,s,s,s))

print (f' {s}{s}{s}{s} '.format(s,s,s,s,s,s,s,s))

print (f' {s}{s}{s}{s} '.format(s,s,s,s,s,s,s,s))

print (f' {s} {s}{s} {s} '.format(s,s,s,s,s,s,s,s))

print (f' {s} {s}{s} {s} '.format(s,s,s,s,s,s,s,s))

## Завдання №4

## Команди, використані для цієї програми:

## 1. `input()`: Запит користувача ввести рядок з клавіатури.

## 2. `len()`: Функція для визначення довжини рядка.

## 3. `if `та `else` : Умовні конструкції для визначення, чи рядок складається з одного символу чи більше.

## 4. `[-1]`: Індексація для отримання останнього елемента рядка.

## 5. `+`: Оператор конкатенації для об'єднання рядків.

## 6. `print()`: Вивід результату на екран.

## 1. Введення рядка користувачем:

## user\_input = input("Введіть рядок: ")

## 

## Користувач вводить рядок, з яким буде виконуватися обмін.

## 2. Перевірка довжини рядка:

## if len(user\_input) > 1:

## 

## Перевіряється, чи рядок має більше одного символу.

## 3. Обмін першого та останнього символу:

## swapped\_string = user\_input[-1] + user\_input[1:-1] + user\_input[0]

## 

## Створюється новий рядок, у якому перший символ замінюється останнім, а останній символ - першим. Середня частина залишається незмінною.

## 4. Виведення обміненого рядка:

## print(swapped\_string)

## 

## Обмінений рядок виводиться на екран.

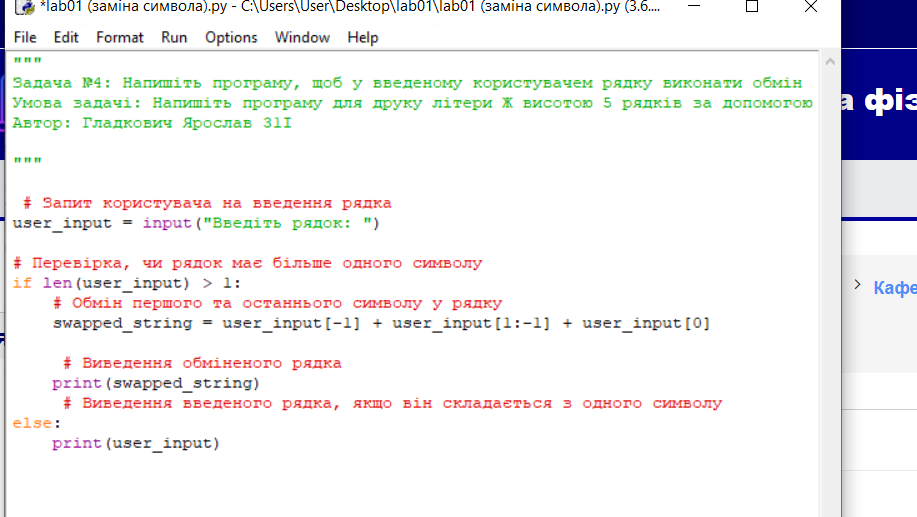
## 5. Виведення введеного рядка, якщо він складається з одного символу:

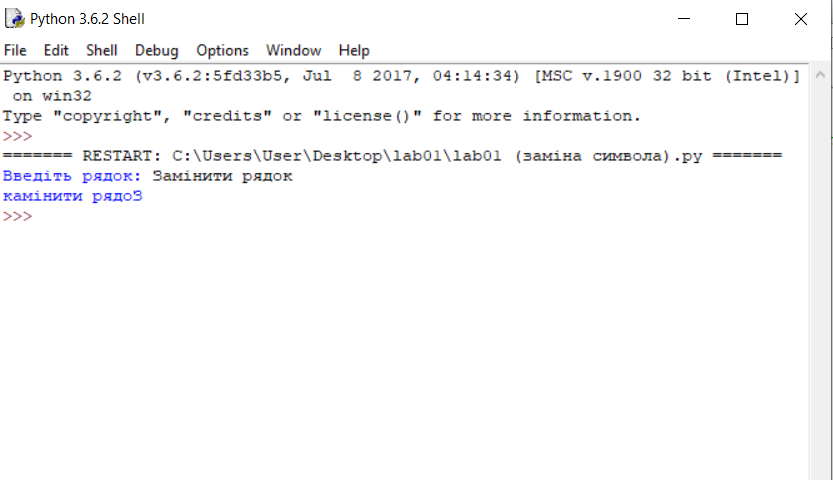
## else:

## print(user\_input)

## 

## Якщо рядок складається лише з одного символу, виводиться той самий рядок без змін.





## 2.1 Блок-схема завдання №1

Початок

Увести час = float(input("Уведіть число: "))

print("Дні:Години:Хвилини:Секунди-> %d:%d:%d:%d" %

Кінець

секунди = час

хвилини = час // 60

час %= 60

години = час // 3600

час %= 3600

дні = час // (24 \* 3600)

час = час % (24 \* 3600)

**Блок-схема завдання №2**

Початок

Години = int(input())

хвилини = int(input())

секунди = int(input())

Кінець

print(секунд\_пройшло)

всього\_секунд1 = години1 \* 3600 + хвилини1 \* 60 + секунди1

всього\_секунд2 = години2 \* 3600 + хвилини2 \* 60 + секунди2

секунд\_пройшло = всього\_секунд2 - всього\_секунд1)

**Блок-схема завдання №4**

Початок

user\_input = input("Введіть рядок: ")

Кінець

print(swapped\_string)

print(user\_input)

if len(user\_input) > 1:

swapped\_string = user\_input[-1] + user\_input[1:-1] + user\_input[0]

# Висновки

У ході виконання лабораторної роботи з теми "Лінійні алгоритми мовою Python" були розглянуті різноманітні завдання, які демонструють використання базових конструкцій мови програмування Python для розв'язання різних завдань. Програми були написані для вирішення практичних задач, таких як перетворення часу, робота з рядками та символами.Кожна програма включала в себе введення даних від користувача, обробку цих даних за допомогою різних операцій та виведення результату. Використані команди були ретельно розписані для кращого розуміння, а також надані пояснення кожного етапу виконання програми.В ході роботи були використані основні концепції мови програмування Python, такі як умовні конструкції, цикли, операції з рядками та числами, введення та виведення даних. Це дозволило здійснити вирішення різноманітних завдань з ефективністю та зрозумілістю коду.

Загалом, лабораторна робота допомогла поглибити розуміння лінійних алгоритмів, а також дозволила отримати практичні навички використання мови програмування Python для розв'язання конкретних завдань.

# Список літератури

Список літератури

# Додатки

Лістинги програм