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 №3

«Цикли»

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 10](#_Toc150170357)

[Список літератури 12](#_Toc150170358)

[Додатки 13](#_Toc150170359)

# Мета роботи

Мета лабораторної роботи – скласти програми мовою Python для вирішення задач з теми «Цикли»

# 1 Постановка задачі

1. Визначте суму усіх елементів послідовності, яка завершується числом 0. Вводиться послідовність цілих чисел, що закінчується числом 0 (саме число 0 в послідовність не входить, а використовується як ознака її закінчення).
2. Дано натуральне число n. Визначте, яким за рахунком числом Фібоначчі воно є. Якщо n не є числом Фібоначчі, виведіть значення -1.

# 

# 2 Основна частина

## 2.1 Опис вхідних та вихідних даних

***Завдання №1***

Команди, використані для цієї програми:

**1. `input("Введіть послідовність цілих чисел")`:** Запит користувача ввести послідовність цілих чисел.

**2. `list(map(int, input\_str.split()))`:** Розбиває введену стрічку на список цілих чисел.

**3. `for num in numbers:`:** Цикл для обходження кожного числа в списку.

**4. `if num == 0: break`:** Умова виходу з циклу, якщо зустрічено число 0.

**5. `total\_sum += num`:** Додавання поточного числа до загальної суми.

**6. `print("Загальна сума:", total\_sum)`:** Виведення загальної суми на екран.

1. Запит введення послідовності від користувача:

input\_str = input("Введіть послідовність цілих чисел")

Користувач вводить послідовність цілих чисел, розділених пробілами.

2. Розбивка введеної стрічки на список чисел:

numbers = list(map(int, input\_str.split()))

Введена стрічка розбивається на окремі числа та створюється список цілих чисел.

3. Обчислення загальної суми:

total\_sum = 0

for num in numbers:

if num == 0:

break

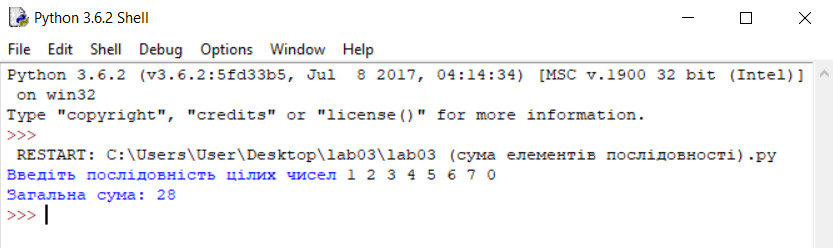
total\_sum += num

Запускається цикл, який обходить кожне число в списку. Якщо зустрічається число 0, цикл завершується. В іншому випадку числа додаються до загальної суми.

4. Виведення результату на екран:

print("Загальна сума:", total\_sum)

Загальна сума виводиться на екран.



***Завдання №2***

Команди, використані для цієї програми:

**1. `int(input("Введіть натуральне число: "))`:** Запит користувача ввести натуральне число.

**2. `isPerfectSquare(x)`:** Функція для перевірки, чи є число повним квадратом.

**3. `isFibonacci(n)`:** Функція для перевірки, чи є число числом Фібоначчі.

4**. `print("Число", num, "є числом Фібоначчі.")`** та `**print("Число", num, "не є числом Фібоначчі, а тому значення -1.")`:** Виведення результату на екран.

1. Запит введення натурального числа від користувача:

num = int(input("Введіть натуральне число: "))

Користувач вводить натуральне число для подальшої перевірки.

2. Перевірка, чи є число повним квадратом:

def isPerfectSquare(x):

s = int(x 0.5)

return s \* s == x

Визначається функція `isPerfectSquare(x)`, яка перевіряє, чи є число `x` повним квадратом.

3. Перевірка, чи є число числом Фібоначчі:

def isFibonacci(n):

return isPerfectSquare(5 \* n \* n + 4) or isPerfectSquare(5 \* n \* n - 4)

Створюється функція `isFibonacci(n)`, яка використовує функцію `isPerfectSquare(x)` для перевірки, чи є число `n` числом Фібоначчі.

4. Виведення результату на екран:

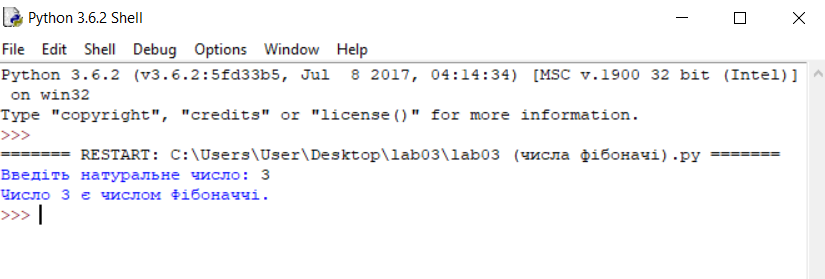
if isFibonacci(num):

print("Число", num, "є числом Фібоначчі.")

else:

print("Число", num, "не є числом Фібоначчі, а тому значення -1.")

Здійснюється перевірка введеного числа на належність до чисел Фібоначчі, та виведення відповідного результату на екран. Якщо число є числом Фібоначчі, виводиться повідомлення, в іншому випадку виводиться `-1`.



## 2.1 Блок-схема завдання №1

Початок

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

Кінець

print("Загальна сума:", total\_sum)

total\_sum = 0 for num in numbers: if num == 0: break total\_sum += num

numbers = list(map(int, input\_str.split()))

**Блок-схема завдання №2**

Початок

num = int(input("Введіть натуральне число: "))

Кінець

if isFibonacci(num): print("Число", num, "є числом Фібоначчі.") else: print("Число", num, "не є числом Фібоначчі.")

def isPerfectSquare(x): s = int(x 0.5) return s \* s == x

# Висновки

У ході виконання лабораторної роботи з теми "Цикли", було розглянуто різноманітні завдання, що використовують циклічні конструкції у мові програмування Python. Кожна програма вирішувала конкретне завдання, використовуючи цикли для обробки даних, введених користувачем або переданих в програму.

У результаті виконання лабораторної роботи створено програми, які вирішують різноманітні завдання, використовуючи циклічні конструкції. Це дозволило поглибити розуміння роботи циклів у мові програмування Python та їхнє практичне застосування для розв'язання конкретних задач.

# Список літератури

Список літератури

# Додатки

https://github.com/YH31I/Lab03