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 №4

«»

з дисципліни «Програмування»

Виконав:

студент ІІІ курсу групи 31І

Бритвич О.В.

Перевірила:

викладач Устименко О.Б.

Оцінка

Дата \_\_\_\_\_\_\_\_\_\_\_\_\_

Київ – 2023

Зміст

Мета роботи 3

1 Постановка задачі 4

2 Основна частина 5

2.1 Опис вхідних та вихідних даних 7

2.1 Блок-схема 8

2.2 Опис вхідних та вихідних даних 14

2.2 Блок-схема 15

Висновки 18

Список літератури 19

Додатки 20

**Мета роботи**

Скласти програми які мають надавати правильні результати для різних вхідних значень і бути легкими для розуміння та використання.

**1 Постановка задачі**

1. Задача №1.

Напишіть програму, яка знайде максимально можливе число, яке можна отримати після видалення однієї цифри. Перший рядок містить одне ціле число *n* *(10 ≤ n ≤ 2025)*. Виведіть у другому рядку максимально можливе число, яке можна отримати після видалення однієї цифри.

1. Задача №2.

Дано список країн і міст кожної країни. Потім подані назви міст. Для кожного міста вкажіть, в якій країні воно знаходиться. Програма отримує на вхід кількість країн *n*. Далі йде *n* рядків, кожен рядок починається з назви країни, потім йдуть назви міст цієї країни. У наступному рядку записано число *m*, далі йдуть *m* запитів – назви якихось m міст, перерахованих вище. Для кожного із запиту виведіть назву країни, в якій знаходиться дане місто.

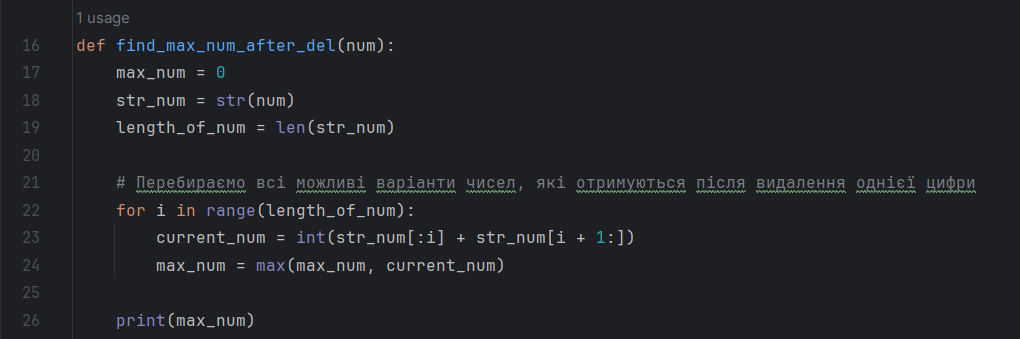
**2 Основна частина**

Задача №1

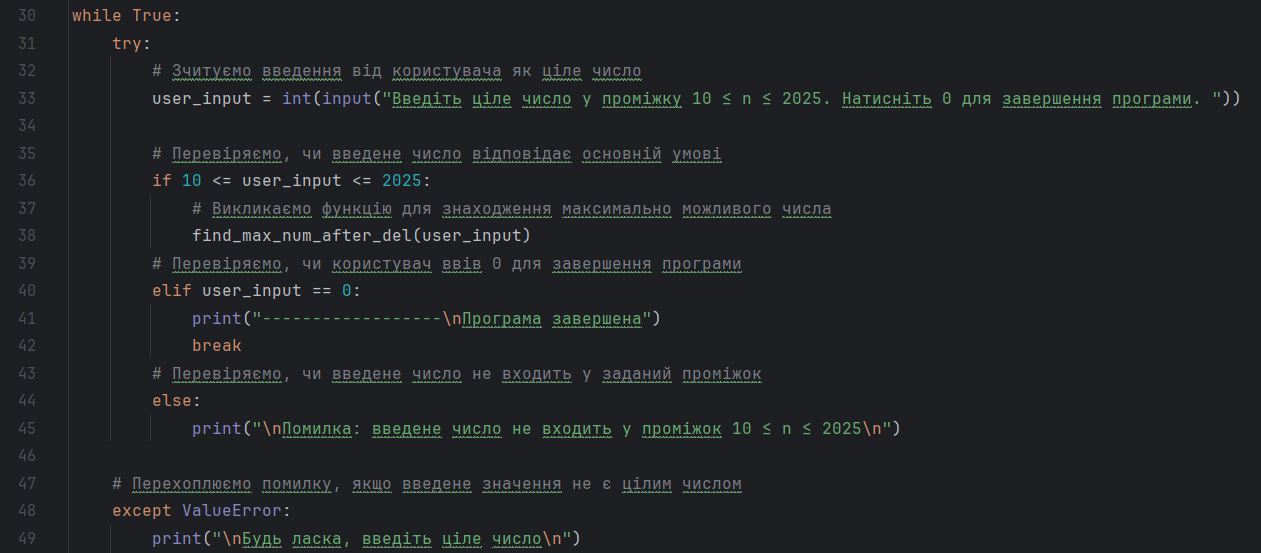
Код даної програми починається з ініціалізації функції, *find\_max\_num\_after\_del*, яка приймає ціле число *num* і виводить на екран максимально можливе число, яке можна отримати, видаливши одну цифру з введеного числа. Ініціалізація змінної *max\_num* використовується для збереження максимального знайденого числа після видалення однієї цифри. Метод *str(num)* перетворює введені числа *num* в рядок для зручності роботи з цифрами. Ділі йде метод *len(str\_num)*, який використовується для знаходження кількості цифр у введеному числі. Цикл *for i in range(length\_of\_num)* перебирає всі можливі позиції для видалення цифр у введеному числі за допомогою двох нових змінних:

* *current\_num* створює нове число, видаляючи одну цифру на позиції *i*.
* *max\_num* порівнює поточні числа з максимальним числом. Якщо поточне число більше, воно стає новим максимальним числом.

Функція закінчується виводом максимального числа за умовою задачі:



Цикл *while True* використовується для безперервного введення чисел від користувача. У разі введення числа, яке задовольняє умові *10 ≤ n ≤ 2025*, викликається функція *find\_max\_num\_after\_del*. Якщо користувач введе *0*, програма завершується, інакше виводиться повідомлення про помилку, якщо введене число не входить у вказаний проміжок, або якщо введене значення не є цілим числом:



2.1 Опис вхідних та вихідних даних

Вхідні дані:

*user\_input* – змінна для зберігання введеного користувачем цілого числа.

Вихідні дані:

Результат роботи функції *find\_max\_num\_after\_del*.

2.1 Блок-схема

Початок

find\_max\_num\_after\_del(num)

while True

for i in range(length\_of\_num)

Вивести max\_num

max\_num = max(max\_num, current\_num)

length\_of\_num = len(str\_num)

str\_num = str(num)

current\_num = int(str\_num[:i] + str\_num[i + 1:])

max\_num = 0

try

Вивести текст про завершення програми

elif user\_input == 0

if 10 <= user\_input <= 2025

Увести user\_input

find\_max\_num\_after\_del(user\_input)

break

else

Вивести текст про невідповідність умові задачі

except ValueError

Вивести текст про некороктно-введені дані

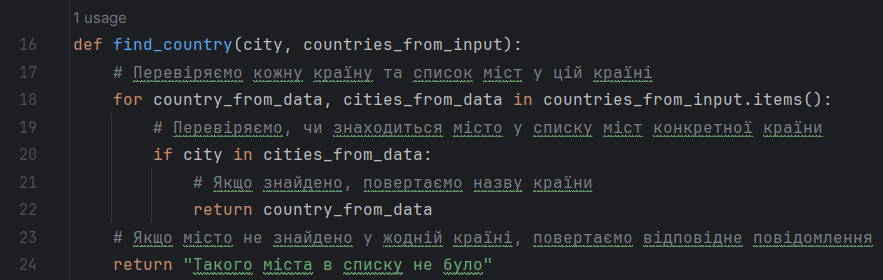
Кінець

Задача №2

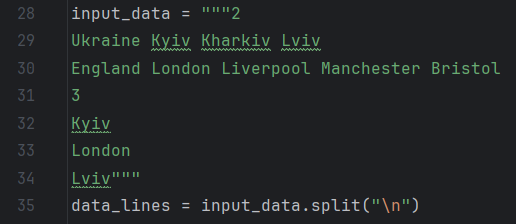
Код починається з функції *find\_country*, яка приймає два аргументи:

* *city* (місто, для якого потрібно знайти країну)
* *countries\_from\_input* (словник, де ключі – це країни, а значення – це списки їхніх міст).

Далі проходиться по ключам та значенням у словнику *countries\_from\_input*. Якщо задане місто (*city*) знаходиться в одному зі списків міст, повертає відповідний ключ (*країну*). Якщо місто не знайдено в жодній країні, повертає рядок «*Такого міста в списку не було»*:



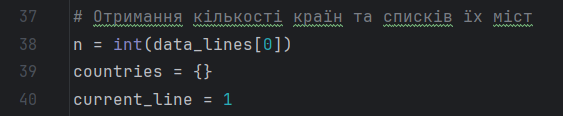
Далі йде рядок, який містить введені дані (*input\_data*). Спочатку задається кількість країн (у тестовому випадку 2), а потім для кожної країни задається назва країни та її міста:



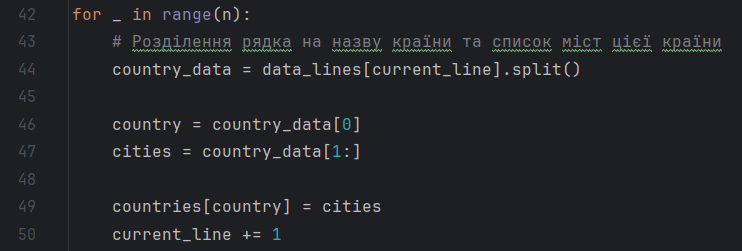
Наступний рядок *data\_lines = input\_data.split("\n")* розбиває рядок *input\_data* на список рядків за символом нового рядка (*\n*):



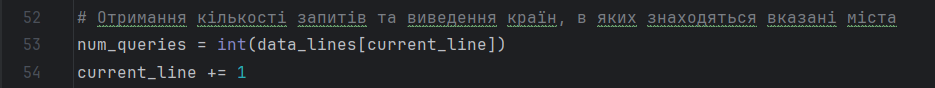
Створюємо пустий словник *countries*. Рядок *n = int(data\_lines[0])* отримує кількість країн з першого рядка. Словник *countries* створюється шляхом зчитування країн та їхніх міст з вхідних даних та додавання їх до словника:



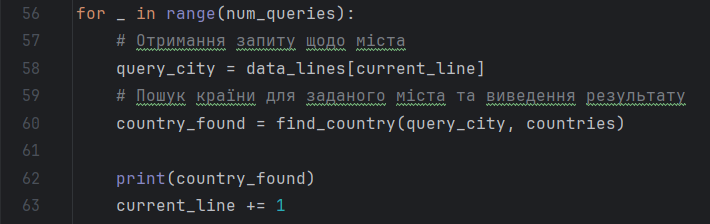
Проходження циклом *for \_ in range(n)* через кожен рядок, починаючи з *current\_line*, розбиття рядка на слова та присвоєння першого слова як ключа (назви країни), а решти служать значенням (список міст для цієї країни):



Зчитується наступний рядок після інформації про країни, і його числове значення використовується як кількість наступних запитів:



Проходження циклом через кожен запит. Кожен запит – це назва міста (*query\_city*), і за допомогою функції *find\_country*, знаходиться країна, якій належить це місто. Результат виводиться на екран. Потім *current\_line* збільшується для обробки наступного запиту:



2.2 Опис вхідних та вихідних даних

Вхідні дані:

*input\_data* – змінна для зберігання інформації про кількість країн, їх міста та кількість запитів міст.

Вихідні дані:

*country\_found* – змінна для зберігання країни, відповідній введеним містам.

2.2 Блок-схема

Початок

def find\_country(city, countries\_from\_input)

n = int(data\_lines[0])

countries = {}

return "Такого міста в списку не було"

data\_lines = input\_data.split("\n")

return country\_from\_data

if city in cities\_from\_data

input\_data

for country\_from\_data, cities\_from\_data in countries\_from\_input.items()

current\_line = 1

cities = country\_data[1:]

country\_data = data\_lines[current\_line].split()

for \_ in range(n)

country = country\_data[0]

countries[country] = cities

current\_line += 1

num\_queries = int(data\_lines[current\_line])

current\_line += 1

for \_ in range(num\_queries)

query\_city = data\_lines[current\_line]

country\_found = find\_country(query\_city, countries)

current\_line += 1

Вивести знайдену країну

Кінець

**Висновки**

Лабораторна робота включала розв’язання двох задач.

Перша задача передбачала написання програми для знаходження максимально можливого числа, що залишається після видалення однієї цифри з заданого числа *n*.

Друга задача полягала у створенні програми, яка визначала країну, в якій розташовані введені міста, відповідно до вхідних даних щодо кількості країн, їх назв та переліку міст.

Обидві програми були успішно реалізовані та протестовані на різних тестових випадках, що дозволило підтвердити їх коректну роботу. Ефективність роботи програм була підтверджена в межах поставлених завдань.

Результати лабораторної роботи можуть бути використані у сферах, де необхідно опрацьовувати та аналізувати дані щодо чисел та відомостей про країни та міста. Наприклад, це може бути використано для оптимізації даних у соціальних мережах, геолокаційних сервісах та аналізі географічних даних.

**Список літератури**

1Вивчаємо Python, В двох томах. 5-е видання. Марк Лутц (комплект з двох книг) – [Режим доступу] : <https://leader-books.com.ua/komp-yuterna-literatura/vivchayemo-python-v-dvoh-tomah-5-e-vidannya-mark-lutc-komplekt-z-dvoh-knig?gclid=CjwKCAiAgeeqBhBAEiwAoDDhnxzTW_ZFWceIMmEQLdDM0XPtB2uG9tUdKMSjvmw6BX7zHEoS49h0gxoC_skQAvD_BwE>

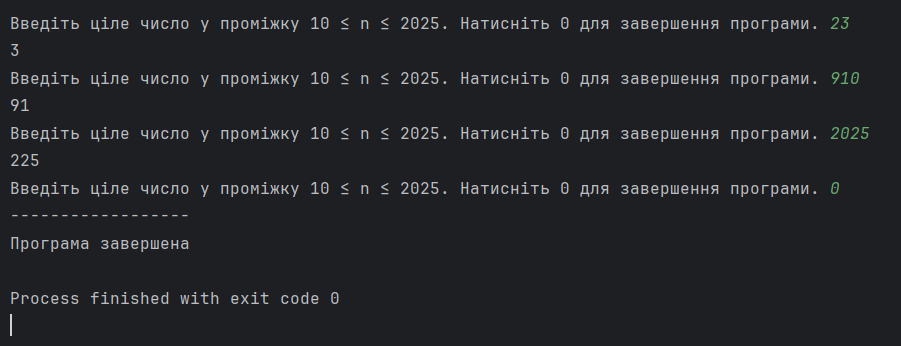
2 Форум з технічних питань – [Режим доступу] : <https://stackoverflow.com/>

**Додатки**

Текст програми №1

# Функція для знаходження максимально можливого числа  
def find\_max\_num\_after\_del(num):  
 max\_num = 0  
 str\_num = str(num)  
 length\_of\_num = len(str\_num)  
  
 # Перебираємо всі можливі варіанти чисел, які отримуються після видалення однієї цифри  
 for i in range(length\_of\_num):  
 current\_num = int(str\_num[:i] + str\_num[i + 1:])  
 max\_num = max(max\_num, current\_num)  
  
 print(max\_num)  
  
  
# Безкінечний цикл для введення  
while True:  
 try:  
 # Зчитуємо введення від користувача як ціле число  
 user\_input = int(input("Введіть ціле число у проміжку 10 ≤ n ≤ 2025. Натисніть 0 для завершення програми. "))  
  
 # Перевіряємо, чи введене число відповідає основній умові  
 if 10 <= user\_input <= 2025:  
 # Викликаємо функцію для знаходження максимально можливого числа  
 find\_max\_num\_after\_del(user\_input)  
 # Перевіряємо, чи користувач ввів 0 для завершення програми  
 elif user\_input == 0:  
 print("------------------\nПрограма завершена")  
 break  
 # Перевіряємо, чи введене число не входить у заданий проміжок  
 else:  
 print("\nПомилка: введене число не входить у проміжок 10 ≤ n ≤ 2025\n")  
  
 # Перехоплюємо помилку, якщо введене значення не є цілим числом  
 except ValueError:  
 print("\nБудь ласка, введіть ціле число\n")

Результат роботи програми №1



Текст програми №2

# Функція для знаходження країни за назвою міста в списку країн та міст  
def find\_country(city, countries\_from\_input):  
 # Перевіряємо кожну країну та список міст у цій країні  
 for country\_from\_data, cities\_from\_data in countries\_from\_input.items():  
 # Перевіряємо, чи знаходиться місто у списку міст конкретної країни  
 if city in cities\_from\_data:  
 # Якщо знайдено, повертаємо назву країни  
 return country\_from\_data  
 # Якщо місто не знайдено у жодній країні, повертаємо відповідне повідомлення  
 return "Такого міста в списку не було"  
  
  
# Вхідні дані для тестування програми  
input\_data = """2  
Ukraine Kyiv Kharkiv Lviv  
England London Liverpool Manchester Bristol  
3  
Kyiv  
London  
Lviv"""  
data\_lines = input\_data.split("\n")  
  
# Отримання кількості країн та списків їх міст  
n = int(data\_lines[0])  
countries = {}  
current\_line = 1  
  
for \_ in range(n):  
 # Розділення рядка на назву країни та список міст цієї країни  
 country\_data = data\_lines[current\_line].split()  
  
 country = country\_data[0]  
 cities = country\_data[1:]  
  
 countries[country] = cities  
 current\_line += 1  
  
# Отримання кількості запитів та виведення країн, в яких знаходяться вказані міста  
num\_queries = int(data\_lines[current\_line])  
current\_line += 1  
  
for \_ in range(num\_queries):  
 # Отримання запиту щодо міста  
 query\_city = data\_lines[current\_line]  
 # Пошук країни для заданого міста та виведення результату  
 country\_found = find\_country(query\_city, countries)  
  
 print(country\_found)  
 current\_line += 1

Результат роботи програми №2

