МІНІСТЕРСТВО НАУКИ ТА ОСВІТИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ «ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

КАФЕДРА СИСТЕМНОГО АНАЛІЗУ ТА ІНФОРМАЦІЙНО-АНАЛІТИЧНИХ ТЕХНОЛОГІЙ

ЛАБОРАТОРНА РОБОТА №3

з дисципліни «**Інструменти Python для AI та Data Science**»

|  |  |
| --- | --- |
|  | Виконала:  студентка групи КН-1223б  Дар’я КОЗЛОВА  Перевірила:  доцент каф. ПІІТУ  Світлана КОВАЛЕНКО |

Харків – 2024

**ЗМІСТ**

[Тема: 3](#_Toc851)

[Мета роботи: 3](#_Toc30527)

[ВПРАВА 1 3](#_Toc32207)

[Код скрипта: 3](#_Toc19605)

[Робота коду: 4](#_Toc23701)

[ВПРАВА 2 4](#_Toc3219)

[Код скрипта: 4](#_Toc7628)

[Робота коду: 5](#_Toc757)

[ВПРАВА 3 5](#_Toc21423)

[Код скрипта: 6](#_Toc16588)

[Робота коду: 6](#_Toc31687)

[ВИСНОВОК 7](#_Toc9043)

Тема: Робота зі списками Python

Мета роботи: **:** Отримати базові знання та навички роботи зі списками Python**.**

**Варіант №1**

# ВПРАВА 1

Введіть список, що складається з *n* цілих елементів. Знайдіть

кількість парних елементів. *n* вводиться з клавіатури.

# Код скрипта:

print("Парні елементи списку")

print("-------------------------")

def count\_even\_numbers():

n = int(input("Введіть кількість елементів у списку: "))

elements = []

for i in range(n):

element = int(input(f"Введіть елемент {i+1}: "))

elements.append(element)

count\_even = 0

for x in elements:

if x % 2 == 0:

count\_even += 1

print(f"Кількість парних елементів: {count\_even}")

while True:

count\_even\_numbers()

choice = input("Повторити? (1 так/ 2 ні): ")

if choice != '1':

print("Вихід із програми.")

break

# Робота коду:

Реалізація логічного виразу і скріншот графіку приведено нижче:

# ВПРАВА 2

Введіть список, в якому є лише два однакових елемента.

Визначте їх індекси.

# Код скрипта:

print("Знаходження однакових елементів")

print("-------------------------")

def find\_duplicate():

n = int(input("Введіть кількість елементів у списку: "))

elements = []

for i in range(n):

element = int(input(f"Введіть елемент {i+1}: "))

elements.append(element)

print("Ваш список:", elements)

for i in range(len(elements)):

for j in range(i + 1, len(elements)):

if elements[i] == elements[j]:

print(f"Однакові елементи знайдено за індексами: {i} і {j}")

return

print("Не знайдено двох однакових елементів.")

while True:

find\_duplicate()

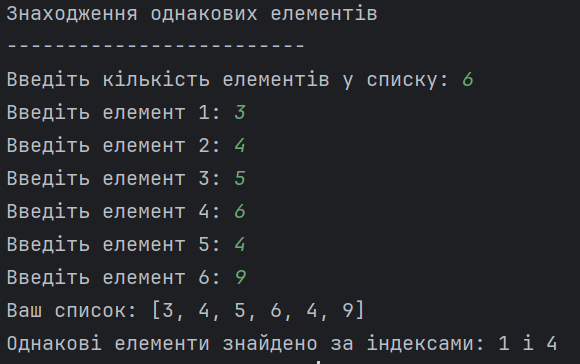
choice = input("Повторити? (1 так/ 2 ні): ")

if choice != '1':

print("Вихід із програми.")

break

# Робота коду:

Результат роботи коду наведена нижче у скріншотах:

# ВПРАВА 3

Видаліть зі списку всі елементи перед тим, що задається в якості

параметра.

Для ілюстрації у нас є список [1, 2, 3, 4, 5], і нам потрібно видалити всі

елементи, які йдуть перед 3, тобто 1 і 2.

У нас є два граничних випадки: (1) якщо елемент, що вводиться, неможливо

знайти, то список не слід змінювати. (2) якщо список порожній, він повинен

залишатися порожнім.

# Код скрипта:

print("Видалення елементів перед заданим параметром")

print("--------------------------------------------")

def remove\_before\_element():

n = int(input("Введіть кількість елементів у списку: "))

elements = []

for i in range(n):

element = int(input(f"Введіть елемент {i + 1}: "))

elements.append(element)

print("Ваш список:", elements)

param = int(input("Введіть число, перед яким потрібно видалити всі елементи: "))

if param in elements:

index = elements.index(param)

result = elements[index:]

print("Результат:", result)

else:

print("Елемент не знайдено. Список не змінюється.")

print("Ваш список:", elements)

while True:

remove\_before\_element()

choice = input("Повторити? (1 так/ 2 ні): ")

if choice != '1':

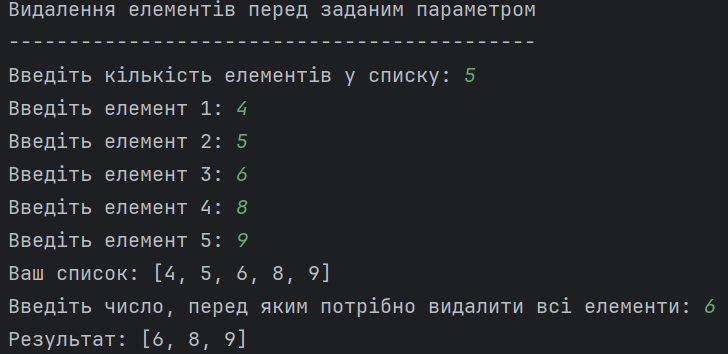
print("Вихід із програми.")

break

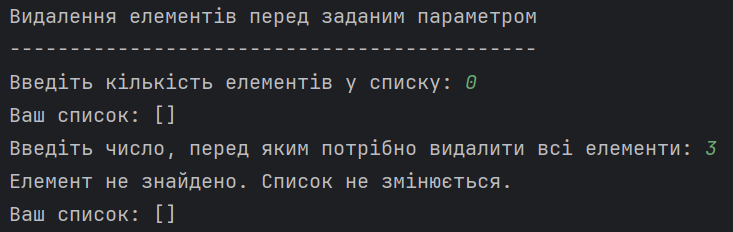
# Робота коду:

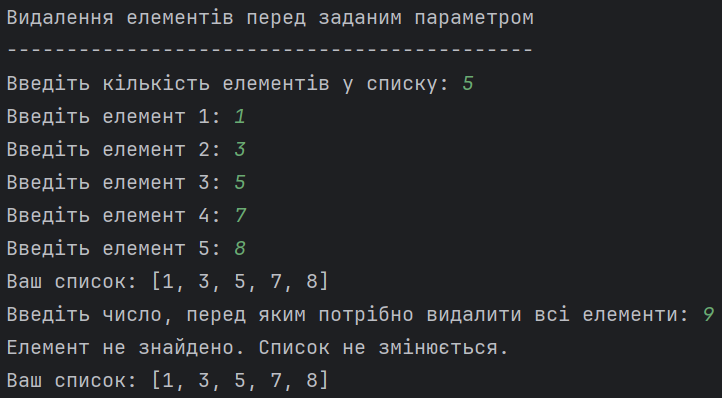
Результат роботи коду наведена нижче у скріншотах:

Перший випадок:



Другий випадок, коли список пустий:

Третій випдак, коли список не має значення вказаного параметру:



# ВИСНОВОК

У процесі виконання лабораторної роботи було отримано практичні знання щодо роботи зі списками в мові програмування Python, що є важливим елементом у роботі з даними. Робота зі списками дозволяє ефективно зберігати, обробляти та маніпулювати даними, що робить цей інструмент одним із ключових у програмуванні.

У першому завданні було розглянуто методи пошуку та підрахунку парних елементів у списку. Це дало можливість закріпити навички використання циклів і умовних операторів для перевірки елементів на відповідність певній умові (парні елементи). Такі операції є необхідними для аналізу даних, пошуку закономірностей або статистичних обчислень у великих наборах даних.

У другому завданні було вивчено пошук однакових елементів і їхніх індексів у списку. Це завдання підкреслило важливість розуміння роботи з індексами в списках, а також ефективного алгоритму пошуку. Це вміння стане корисним у більш складних алгоритмах, де необхідно швидко знаходити або порівнювати елементи даних.

Третє завдання було присвячене видаленню елементів списку до певного значення з урахуванням граничних випадків (порожній список або відсутність шуканого елемента). Це допомогло закріпити розуміння операцій над списками, включаючи видалення елементів та роботу з виключеннями. Обробка виняткових ситуацій є важливою частиною роботи з будь-якими структурами даних, що допомагає забезпечити стабільність і надійність програми.

Загалом, виконання лабораторної роботи дозволило значно поглибити розуміння базових операцій зі списками, що є фундаментальними для подальшого вивчення структур даних і алгоритмів. Ці знання закладають основу для вирішення складніших задач, пов'язаних з обробкою великих обсягів інформації, оптимізацією роботи з даними та розробкою ефективних програмних рішень.