

Міністерство освіти і науки України
Донецький національний університет імені Василя Стуса Факультет
інформаційних і прикладних технологій
Кафедра інформаційних технологій

ЗВІТ

з лабораторної роботи № 5
з дисципліни «Основи програмування»
на тему:
«Організація циклів у мові Python»

Виконав: студент гр. Б25_д/Ф3

Кручківський Ю.О.

Перевірів: доц. Бабаков Р. М.

Завдання 1.

7	Задані два дійсні числа A і B , $A < B$. У діапазоні $[A; B]$ знайти кількість чисел, що є дільниками числа $A*B$
---	--

Лістинг до завдання 1.

```
while True:
    print("Введіть два дійсні числа,  $A < B$ ")
    try:
        A = int(input("Введіть A: "))
        B = int(input("Введіть B: "))

        if not A < B:
            raise ValueError

        break

    except ValueError:
        print("Спробуйте знову")

AB_product = A * B # Знаходимо добуток A та B
count = 0 # ініціалізуємо змінну лічильник

for i in range(A, B + 1): # Створюємо цикл для перебору всіх значень в
    діапазоні (0; B]
    if i != 0 and AB_product % i == 0: # перевіряємо дільники числа
        AB_product
        count += 1 # додаємо 1 в випадку успішного ділення

print(f"У діапазоні [{A}, {B}] {count} чисел є дільниками {AB_product}")
```

Розглянемо приклад використання програми до завдання 1.

```
Введіть два дійсні числа,  $A < B$ 
Введіть A: 45
Введіть B: 234
У діапазоні [45, 234] 12 чисел є дільниками 10530
```

Розглянемо блок схеми до завдання 1.

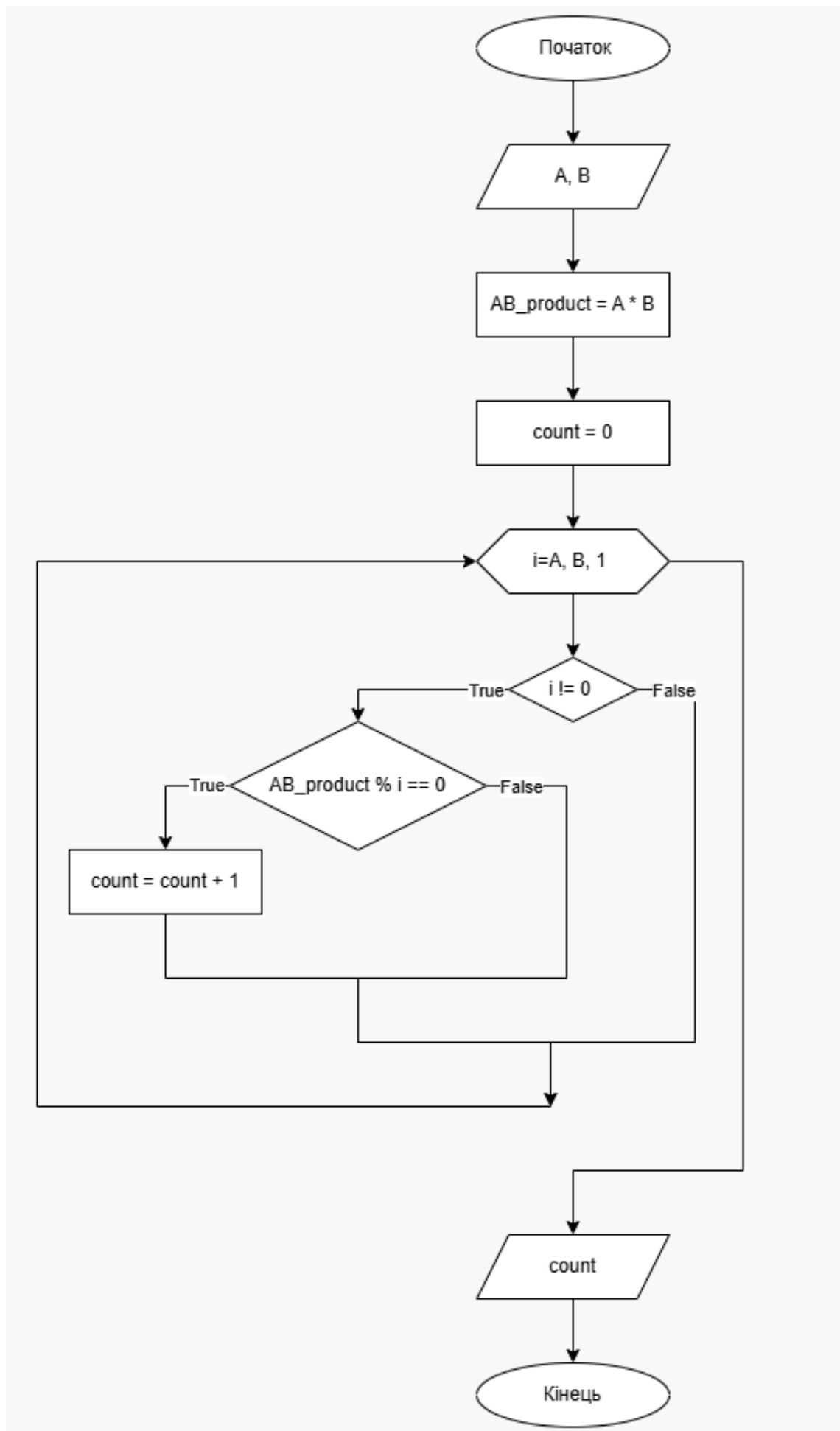


Рисунок 1 – Блок схема до завдання 1

Завдання 2.

7	Натуральне тризначне число A записане у десятковій системі числення у вигляді окремих розрядів a1, a2, a3. Натуральне тризначне число A записане у десятковій системі числення у вигляді окремих розрядів a1, a2, a3.
---	---

В завдання 2 присутні альтернативні рішення (позначені 2.1-2.2):

Лістинг до завдання 2.1.

```
while True:
    print("Введіть сотні десятки і десятки до числа A")
    try:
        a3 = int(input("Введіть сотні: ")[0])
        a2 = int(input("Введіть десятки: ")[0])
        a1 = int(input("Введіть одиниці: ")[0])
        break

    except ValueError:
        print("Спробуйте знову")

A = a3 * 100 + a2 * 10 + a1 # складаємо задані цифри в число A

A_range = range(0, A + 1) # Створимо діапазон чисел (0; A]

first_number = None # ініціалізуємо змінні для збереження чисел
задовільняючих умову
last_number = None

def is_contain_same_digits(number: int) -> bool:
    digits = list(map(int, str(number))) # перетворимо число в масив
цифр
    for digit in set(digits): # створимо цикл для перебору унікальних
цифр числа
        if digit >= 5 and digits.count(digit) == 2: # перевіримо чи
задовільняє число умову
            return True
    return False

for i in A_range: # створимо цикл для перебору чисел діапазону
    if is_contain_same_digits(i): # перевіримо число
        first_number = i # у випадку успіху запишемо число
        break

for i in reversed(A_range): # створимо цикл для перебору чисел діапазону
з кінця
    if is_contain_same_digits(i):
        last_number = i
        break

print(f"Перше число в діапазоні (0; {A}] з двома однаковими цифрами
{first_number}, а останнє {last_number}")
```

Для вирішення завдання 2.1 було використано клас Counter з стандартного пакету collections. Що спростило реалізацію підрахунку цифр в числі.

Розглянемо приклад використання програми до завдання 2.1.

Введіть сотні десятки і десятки до числа A
Введіть сотні: 5
Введіть десятки: 2
Введіть одиниці: 7
Перше число в діапазоні [0; 527] з двома однаковими цифрами 55, а останнє 499

Розглянемо блок схеми до завдання 2.1.

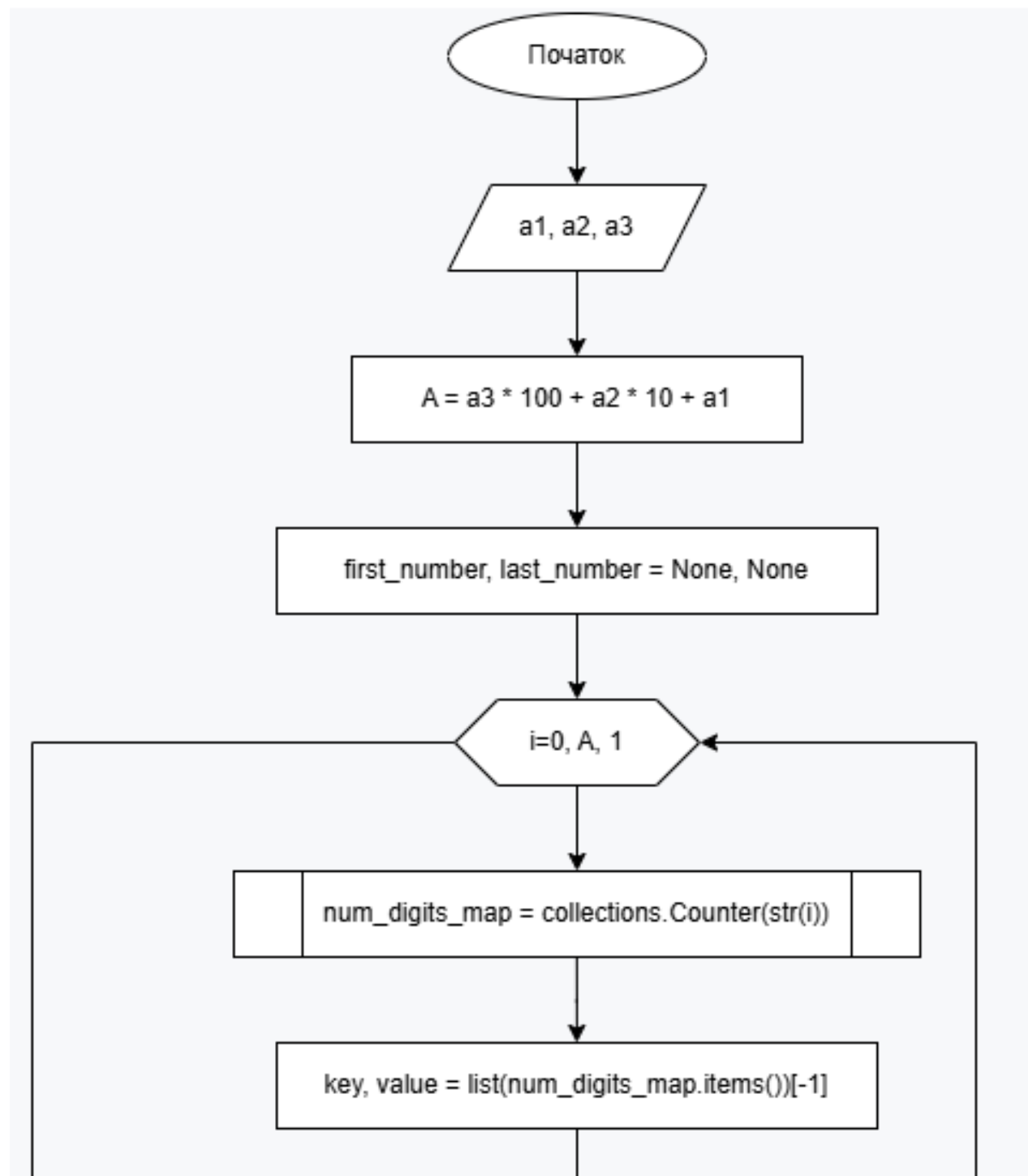


Рисунок 2 – Блок схема до завдання 2.1 (Частина 1)

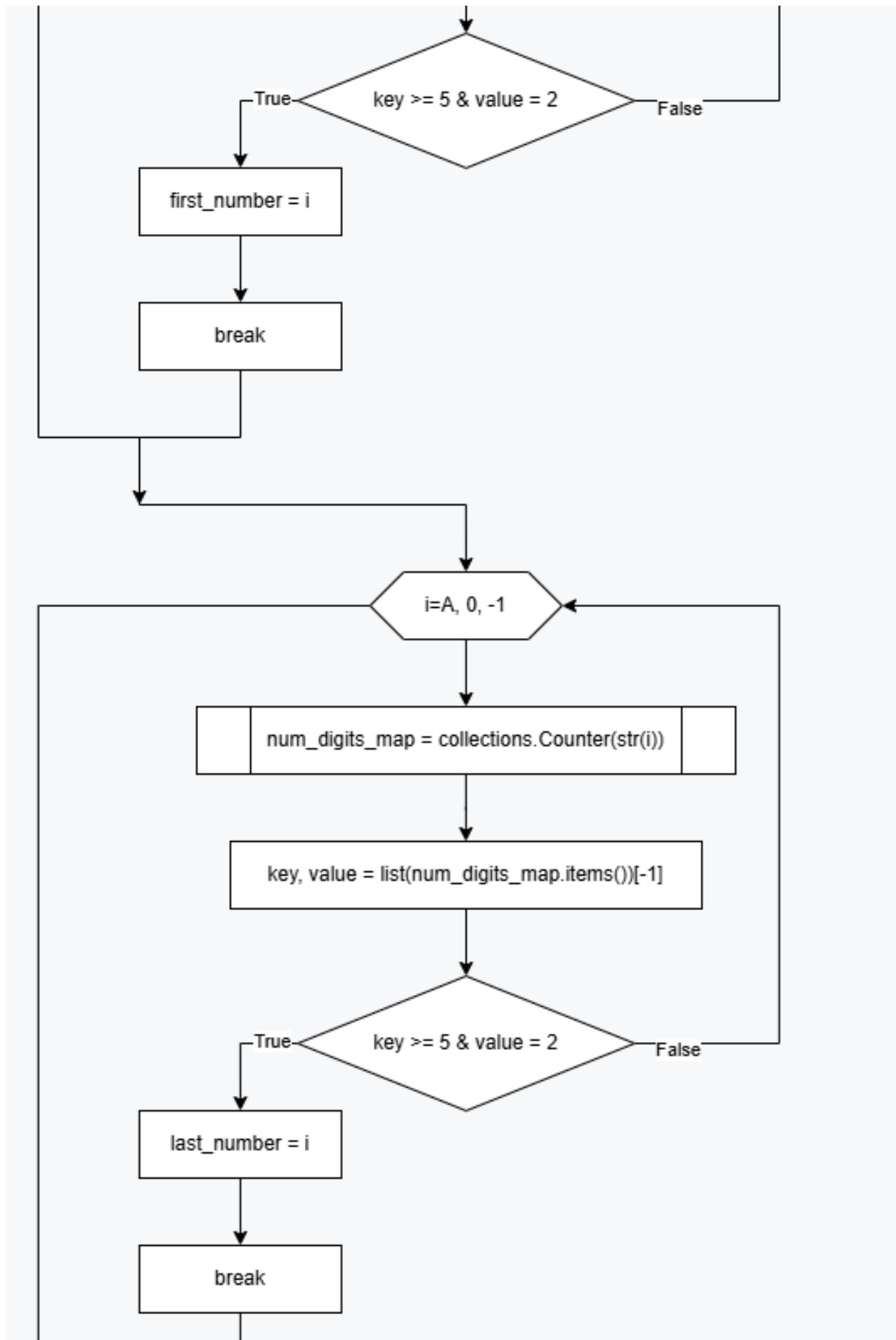


Рисунок 3 – Блок схема до завдання 2.1 (Частина 2)

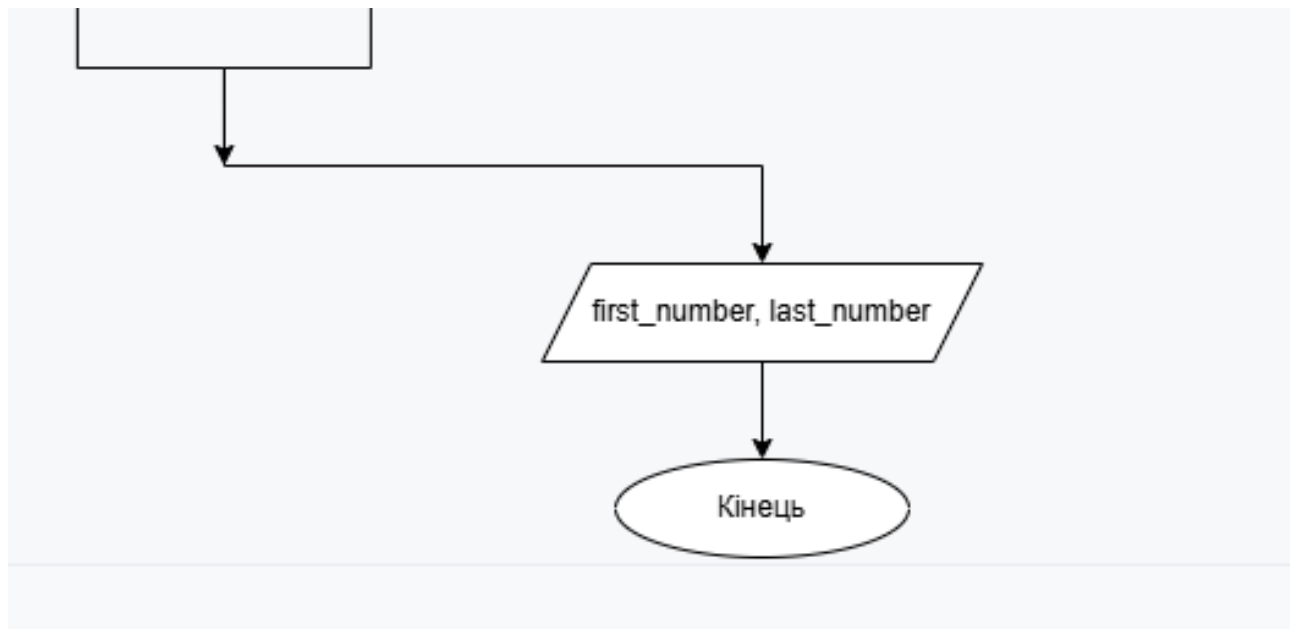


Рисунок 4 – Блок схема до завдання 2.1 (Частина 3)

Лістинг до завдання 2.2.

```

while True:
    print("Введіть сотні десятки і десятки до числа A")
    try:
        a3 = int(input("Введіть сотні: ")[0])
        a2 = int(input("Введіть десятки: ")[0])
        a1 = int(input("Введіть одиниці: ")[0])
        break

    except ValueError:
        print("Спробуйте знову")

A = a3 * 100 + a2 * 10 + a1 # складаємо задані цифри в число A

A_range = range(0, A + 1) # Створимо діапазон чисел (0; A]

first_number = None # ініціалізуємо змінні для збереження чисел
задовільняючих умову
last_number = None

def is_contain_same_digits(number: int) -> bool:
    digits = list(map(int, str(number))) # перетворимо число в масив
цифр
    for digit in set(digits): # створимо цикл для перебору унікальних
цифр числа
        if digit >= 5 and digits.count(digit) == 2: # перевіримо чи
задовільняє число умову
            return True
    return False

for i in A_range: # створимо цикл для перебору чисел діапазону
    if is_contain_same_digits(i): # перевіримо число
        first_number = i # у випадку успіху запишемо число
        break
  
```

```

for i in reversed(A_range): # створимо цикл для перебору чисел діапазону
    з кінця
    if is_contain_same_digits(i):
        last_number = i
        break

print(f"Перше число в діапазоні (0; {A}] з двома однаковими цифрами
{first_number}, а останнє {last_number}")

```

Для вирішення завдання 2.2 було реалізовано інший метод валідації чисел в основі якого лежить метод count(). Також було зменшено к-сть ітерацій завдяки перетворення масиву цифр на множину, що у випадку однакових цифр в числі відкидає заяву ітерацію.

Розглянемо приклад використання програми до завдання 2.2.

```

Введіть сотні десятки і десятки до числа A
Введіть сотні: 4
Введіть десятки: 5
Введіть одиниці: 6
Перше число в діапазоні (0; 456] з двома однаковими цифрами 55, а останнє
455

```

Розглянемо блок схему до завдання 2.2.

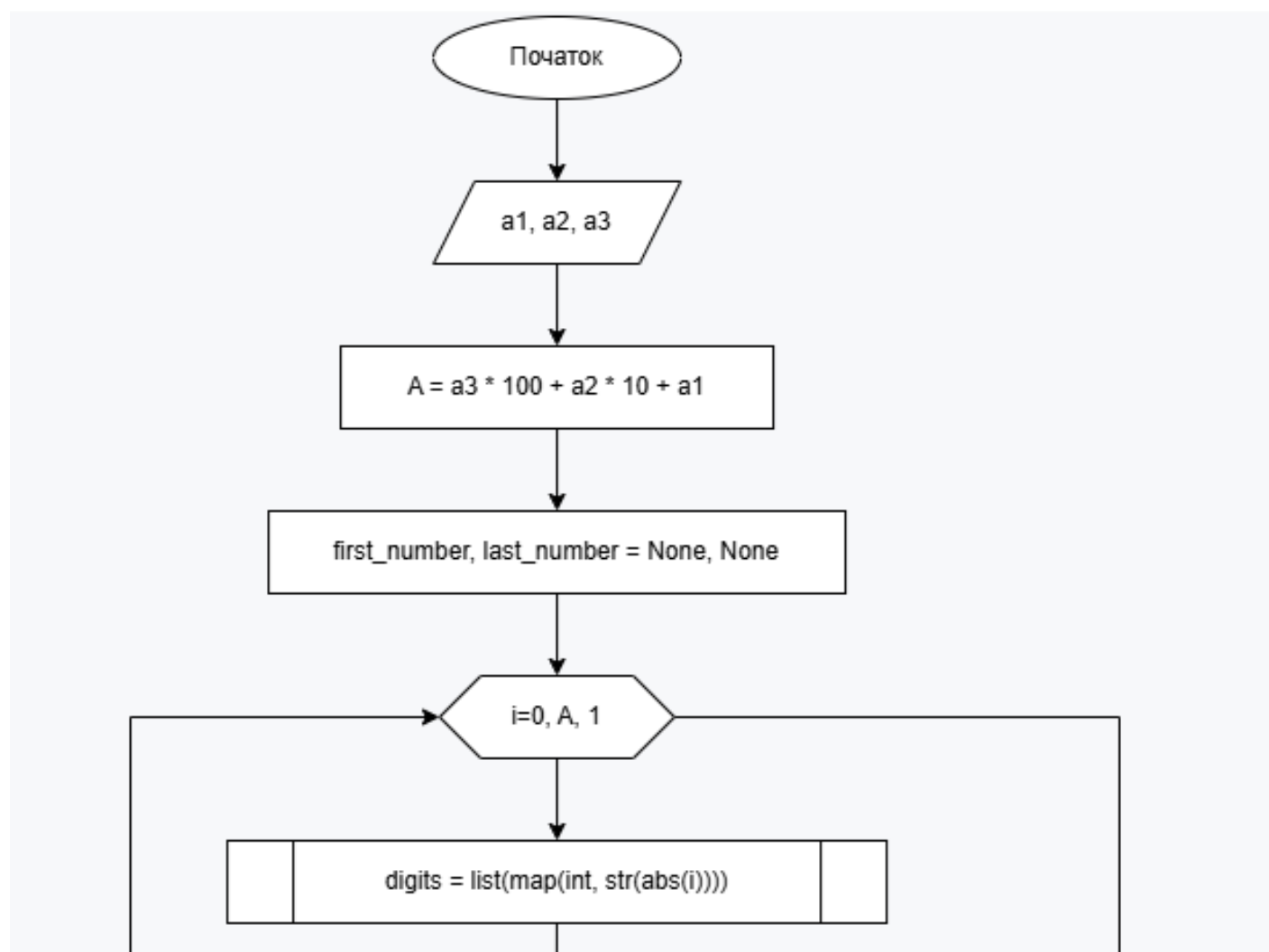


Рисунок 5 – Блок схема до завдання 2.2 (Частина 1)

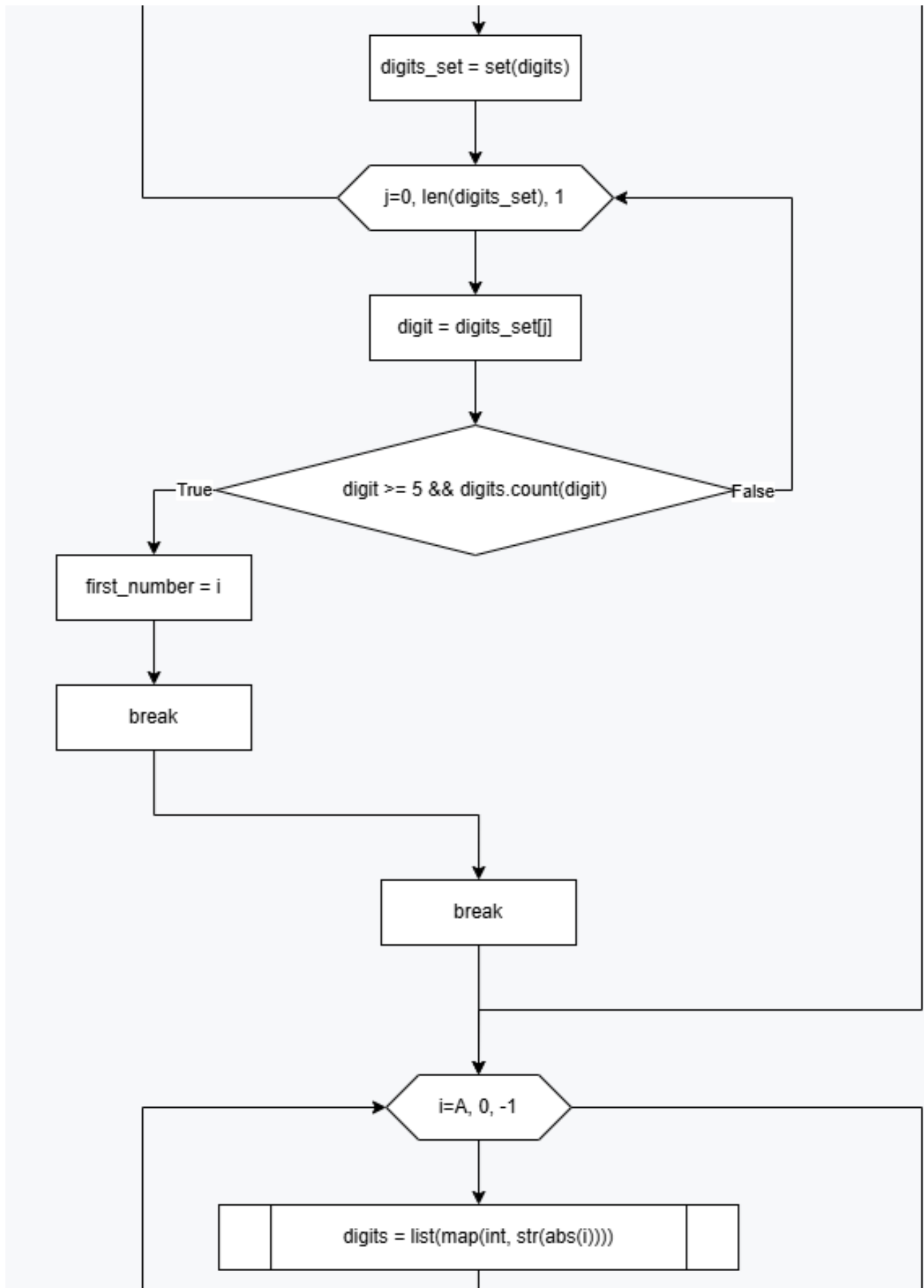


Рисунок 6 – Блок схема до завдання 2.2 (Частина 2)

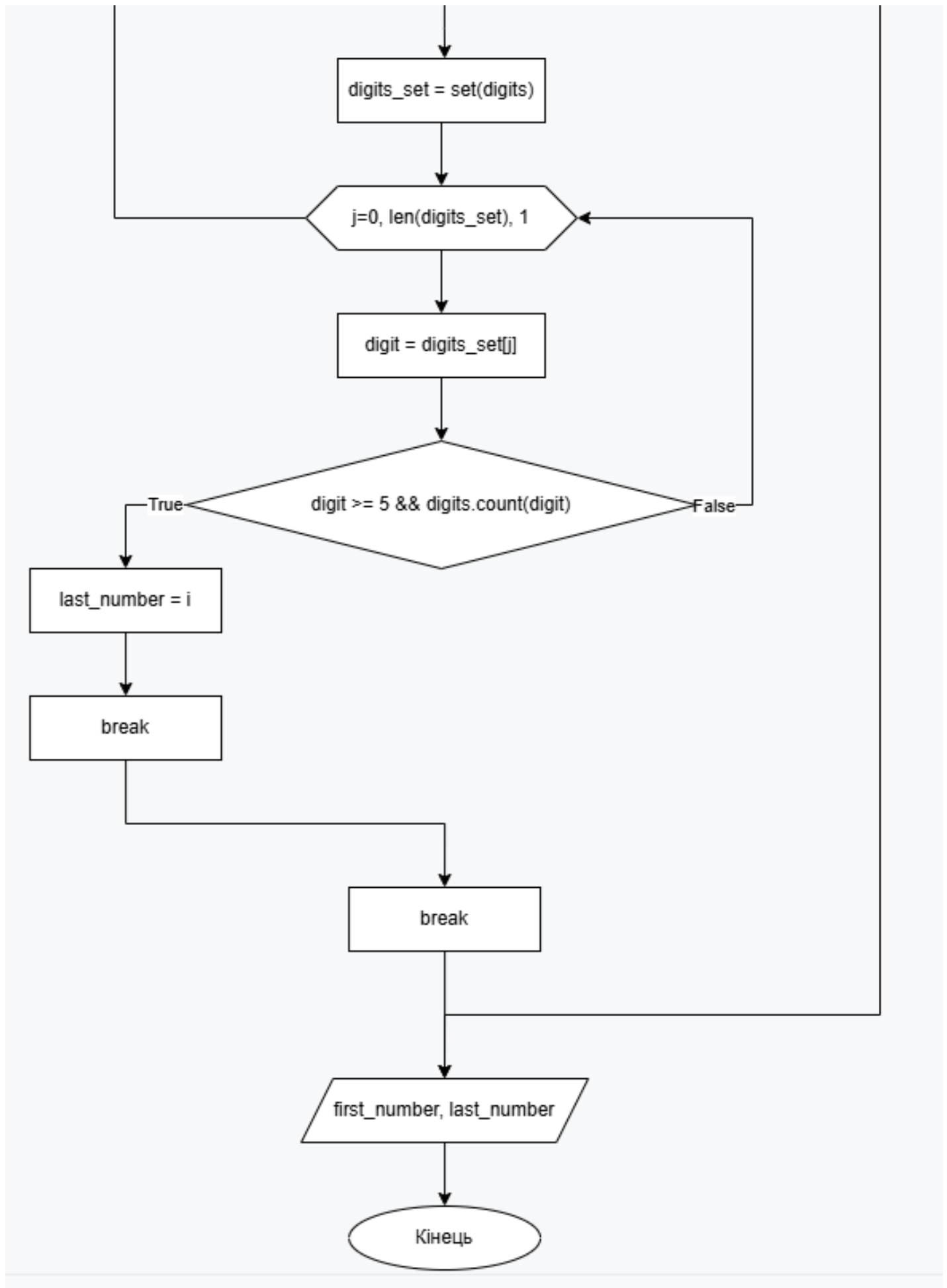


Рисунок 7 – Блок схема до завдання 2.2 (Частина 3)

Висновок до лабораторної роботи №4

Під час виконання лабораторної роботи було розглянуто принципи роботи циклів. Було розглянуто стандартний пакет `collections` залучений у програмі до завдання 2. Проведені дискусії щодо рішення завдання 2, розглянуто декілька варіантів та обрано найефективніший варіант реалізації для заданої умови. Покращені навички з використання циклів на мові Python. Також під час виконання лабораторної роботи було розглянуто виконання інших варіантів завдання 2.