

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

## **З В І Т**

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

Виконав: студент гр. Б25\_д/F3 (Б)

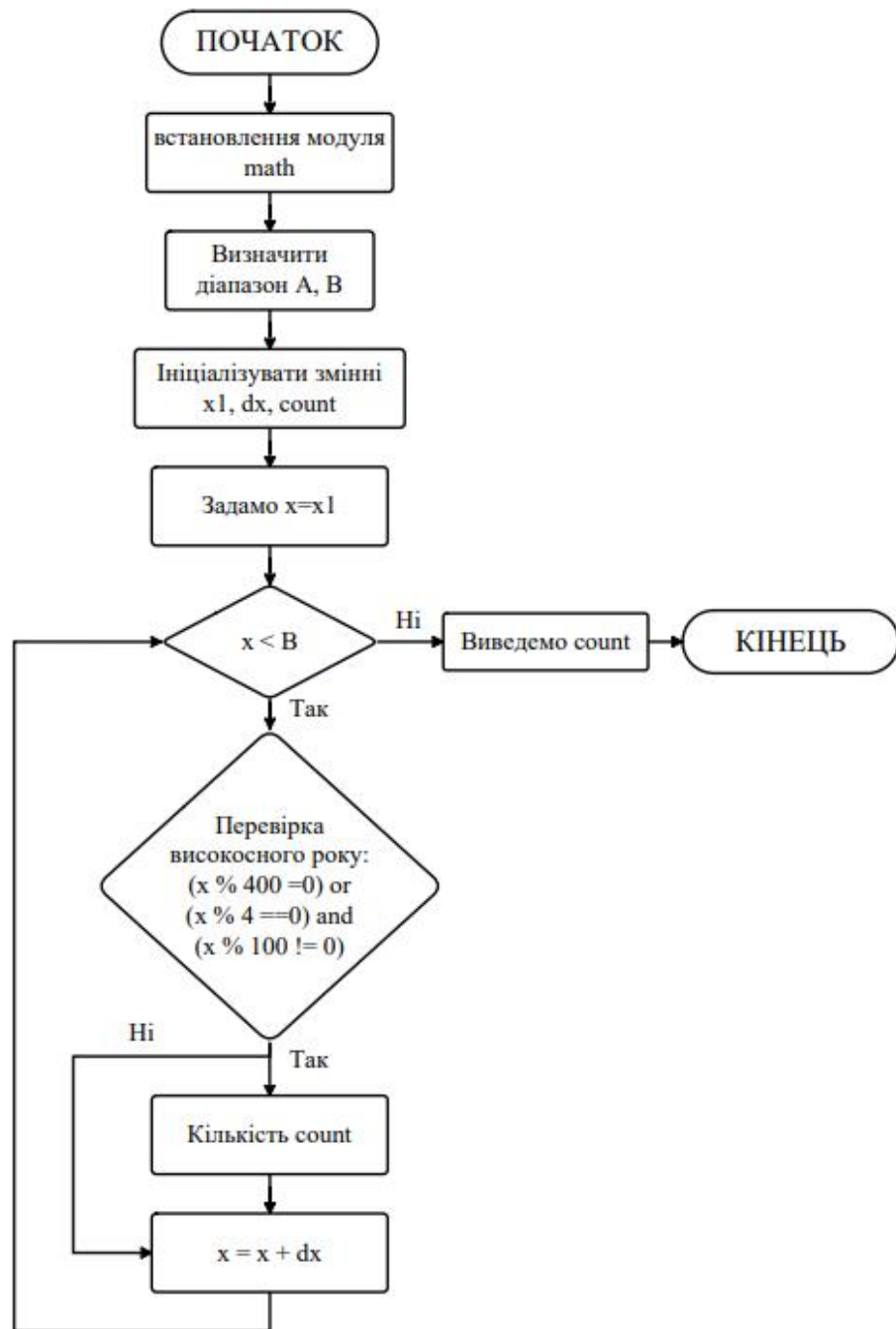
Сауляк Н. Б.

Перевірив: доц. Бабаков Р. М.

Вінниця – 2025

## Завдання 1

№	Завдання
12	У діапазоні [A; B) знайти кількість цілих чисел, які можуть вважатися високосними роками



Потрібно визначити, скільки високосних років міститься у проміжку від A до B (включно з A, але без B).

Оскільки роки — це цілі числа, будемо перебирати лише цілі значення з цього діапазону.

Для цього треба правильно знайти:

- початкове ціле число  $x_1$ ,
- кінцеве ціле число  $x_2$ ,
- крок  $dx$ .

Оскільки ми перебираємо цілі числа, відстань між двома послідовними значеннями дорівнює одиниці:  $dx = 1$

Оскільки діапазон  $[A; B)$  включає  $A$ , то перше число, яке нас цікавить, повинно бути найменшим цілим, не меншим за  $A$ .

Тобто:

- якщо  $A$  — ціле число, то  $x_1 = A$ ;
- якщо  $A$  — дробове число, треба відкинути дробову частину й додати 1, щоб отримати найближче більше ціле.

```
A, B = 1999.5, 2025 # заданий діапазон
```

```
x1 = int(A) + 1 # початкове значення (ціле, не менше A)
print(x1)
```

Результат:

```
2000
```

Оскільки діапазон має вигляд  $[A; B)$ , значення  $B$  не входить у діапазон, тому:

- $x_2$  — це найбільше ціле число, яке менше за  $B$ .

Для цього скористаємося `int()`:

```
x2 = int(B) - 1
```

Але зручніше буде просто перевіряти у циклі умову  $x < B$ , щоб не хвилюватися про включення межі.

Побудова циклу `while`

Перед початком циклу потрібно створити лічильник `count = 0`, у якому накопичуватиметься кількість високосних років.

Початкове значення змінної циклу  $x$  дорівнює  $x_1$ .

Умова циклу: поки  $x < B$ .

Таблиця 1. Лістинг програми (1)

```
A, B = 1999.5, 2025 # заданий діапазон

x1 = int(A) + 1 # початкове значення (ціле, не менше A)
dx = 1 # крок зміни
count = 0 # лічильник високосних років

x = x1 # початкове значення змінної циклу

while x < B: # поки не досягнемо межі B
    if (x % 400 == 0) or (x % 4 == 0 and x % 100 != 0): #
        перевірка високосного року
        count = count + 1
    x = x + dx # перехід до наступного числа

print("Кількість високосних років у діапазоні [", A, ";", B,
      ") =", count)
```

Результат:

```
Кількість високосних років у діапазоні [ 1999.5 ; 2025 ) = 7
```

### Висновок

Програма правильно визначає кількість високосних років у довільному діапазоні  $[A; B)$ , використовуючи цикл `while`, цілі значення змінної циклу та логічні умови для перевірки високосності року.

## Завдання 2

№	Завдання
12	Серед усіх чисел діапазону $[0; A]$ знайти кількість чисел, із цифр яких можна скласти число, яке ділиться одночасно на 2, 5 та 7

### Побудова «каркасу» програми:

Спочатку побудуємо загальний каркас програми, який дозволить перебирати всі числа від 0 до  $A$  за допомогою окремих розрядів — сотень, десятків і одиниць.

У першому рядку задаються розряди числа  $A$ , а у другому — обчислюється саме число  $A$ :

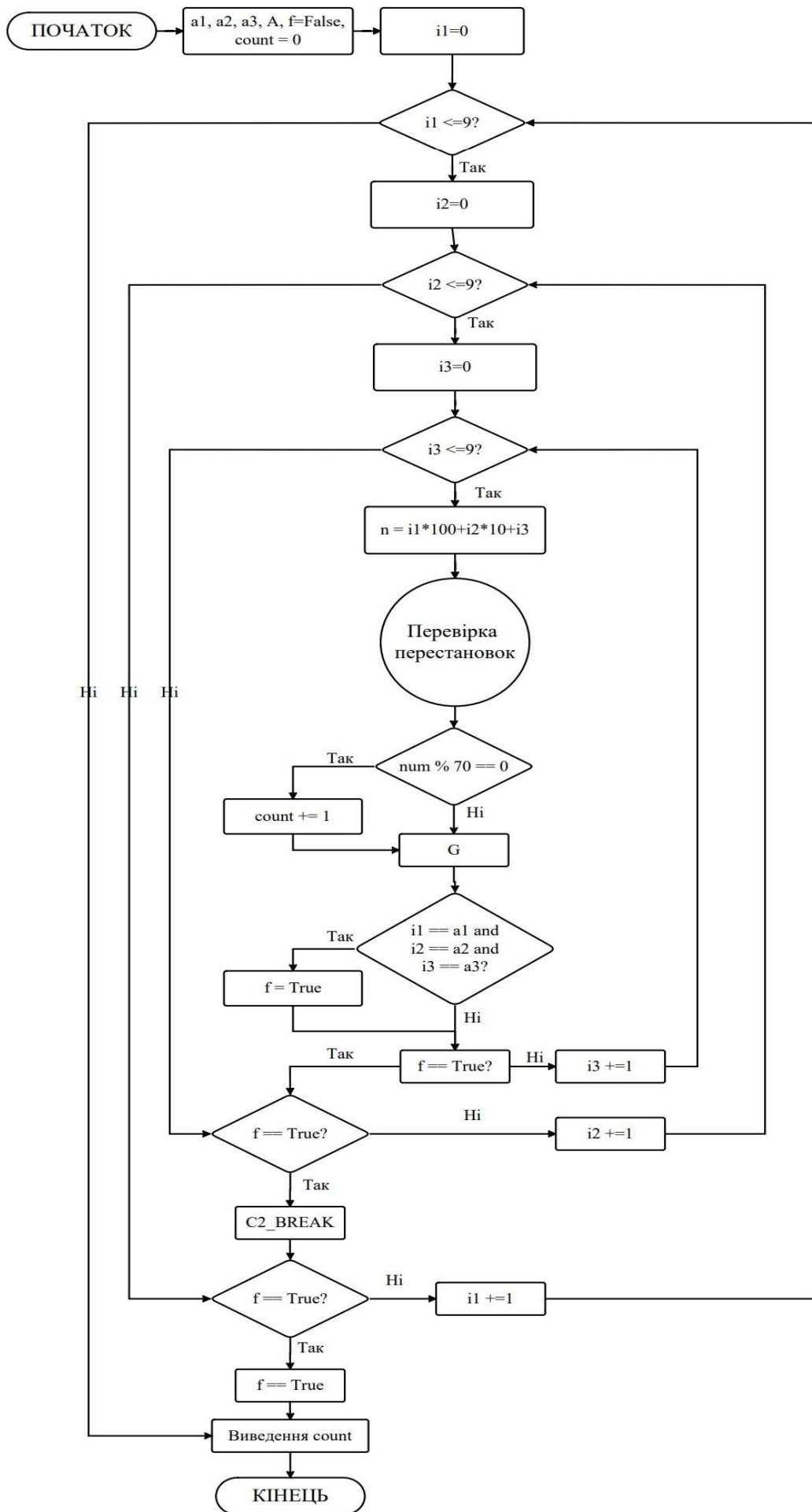
```
a1, a2, a3 = 2, 1, 4 # Задані розряди числа A
A = a1*100 + a2*10 + a3 # Саме число A
```

Програма складатиметься із трьох вкладених циклів:

- зовнішній цикл перебирає сотні ( $i1$ ),
- середній — десятки ( $i2$ ),
- внутрішній — одиниці ( $i3$ ).

```
i1 = 0
while i1 <= 9:
    i2 = 0
    while i2 <= 9:
        i3 = 0
        while i3 <= 9:
            print(i1, i2, i3)
            i3 = i3 + 1
        i2 = i2 + 1
    i1 = i1 + 1
```

Внутрішній цикл відповідає за одиниці, середній — за десятки, а зовнішній — за сотні. Таким чином, програма послідовно перебирає всі можливі тризначні комбінації чисел від 000 до 999.



## Обмеження діапазону

Оскільки за умовою треба розглядати лише числа від 0 до A, додамо прапорець (булеву змінну) f, який показуватиме, коли досягнуто значення A.

Коли  $i1 == a1$ ,  $i2 == a2$ ,  $i3 == a3$ , прапорець f набуває значення True, і виконання циклів треба припинити.

<code>f = False # прапорець завершення</code>
---

Після кожного циклу додаємо перевірку: якщо  $f == True$ , виходимо з циклу командою break.

```
a1, a2, a3 = 2, 1, 4
A = a1*100 + a2*10 + a3
f = False

i1 = 0
while i1 <= 9:
    i2 = 0
    while i2 <= 9:
        i3 = 0
        while i3 <= 9:
            print(i1, i2, i3)
            if i1 == a1 and i2 == a2 and i3 == a3:
                f = True
            if f == True: break
            i3 = i3 + 1
        if f == True: break
        i2 = i2 + 1
    if f == True: break
    i1 = i1 + 1
```

Тепер програма виводить усі числа в діапазоні від 0 до A.

## Підготовка до розв'язання задачі

Замість простого виведення цифр на екран, у тілі внутрішнього циклу потрібно:

1. Сформуувати саме число  $n = i1*100 + i2*10 + i3$ .
2. Перевірити, чи з цифр числа  $n$  можна скласти інше число, яке ділиться на 2, 5 і 7 одночасно.
3. Якщо так — збільшити лічильник на 1.

## Аналіз умови кратності

Число ділиться одночасно на 2, 5 і 7 тоді, коли воно ділиться на 70 (оскільки  $\text{НСК}(2, 5, 7) = 70$ ).

Отже, перевірка матиме вигляд:

```
if num % 70 == 0:
```

## Перевірка перестановок цифр

Щоб перевірити, чи з цифр  $i1$ ,  $i2$ ,  $i3$  можна скласти число, кратне 70, треба створити всі можливі перестановки цифр.

Для цього використаємо ще три вкладені цикли, які вибиратимуть порядок розташування цифр.

Таблиця 2. Лістинг програми (2)

```
a1, a2, a3 = 2, 1, 4 # Задані розряди числа A
A = a1*100 + a2*10 + a3 # Саме число A
f = False # Прапорець завершення циклів
count = 0 # Лічильник підходящих чисел

i1 = 0
while i1 <= 9:
    i2 = 0
    while i2 <= 9:
        i3 = 0
        while i3 <= 9:
```



```
n = i1*100 + i2*10 + i3 # Поточне число
# Перевіряємо усі перестановки цифр n
j1 = 1
found = False
while j1 <= 3:
    j2 = 1
    while j2 <= 3:
        j3 = 1
        while j3 <= 3:
            if j1 != j2 and j1 != j3 and j2 != j3:
                # Вибір цифр за позиціями
                if j1 == 1: d1 = i1
                elif j1 == 2: d1 = i2
                else: d1 = i3

                if j2 == 1: d2 = i1
                elif j2 == 2: d2 = i2
                else: d2 = i3

                if j3 == 1: d3 = i1
                elif j3 == 2: d3 = i2
                else: d3 = i3

                num = d1*100 + d2*10 + d3
                if num % 70 == 0:
                    found = True

            j3 += 1
        j2 += 1
    j1 += 1
```

```
        if found:
            count += 1 # збільшуємо лічильник

        if i1 == a1 and i2 == a2 and i3 == a3:
            f = True
            if f == True: break
            i3 += 1
        if f == True: break
        i2 += 1
    if f == True: break
    i1 += 1

print("Кількість чисел, із цифр яких можна скласти число,
кратне 70:", count)
```

Результат:

```
Кількість чисел, із цифр яких можна скласти число, кратне 70:
36
```

### Висновок

Створено програму, що використовує кілька вкладених циклів while для перебору всіх чисел у діапазоні [0; A].

Програма перевіряє усі перестановки цифр кожного числа та визначає, чи можна з них утворити число, кратне 2, 5 і 7 одночасно.

В результаті виводиться кількість таких чисел, що задовольняють умову.