

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 5 з дисципліни
«Алгоритми та структури даних-1.
Основи алгоритмізації»

«Дослідження складних циклічних
алгоритмів»

Варіант 27

Виконав студент ІІ-13 Пархомчук Ілля Вікторович
(шифр, прізвище, ім'я, по батькові)

Перевірив _____
(прізвище, ім'я, по батькові)

Київ 2021

Лабораторна робота 5

Дослідження складних циклічних алгоритмів

Мета – дослідити особливості роботи складних циклів та набути практичних навичок їх використання під час складання програмних специфікацій.

Варіант 27.

Знайти всі числа, що представляють собою повторювані фрагменти послідовності цифр (наприклад, 3434, 23452345) з інтервалу [1000,1000000]

Постановка задачі

Потрібно перебрати числа від 1000 до 1000000, перевіряючи усі цифри відповідного числа, чи не є вони повторюваними фрагментами деякої послідовності. Знайдене число вивести.

Побудова математичної моделі

Таблиця змінних

Змінна	Тип	Ім'я	Призначення
Лічильник чисел проміжку	Ціличисельний	i	Проміжні дані, Вихідні дані
Кількість цифр у числі	Ціличисельний	num_of_digits	Проміжні дані
Спеціальне число на яке буде ділитися відповідне число з інтервалу	Ціличисельний	div_num	Проміжні дані
Кількість цифр у послідовності	Ціличисельний	num_in_sequence	Проміжні дані
Степінь у яку піднесеться 10	Ціличисельний	power_10	Проміжні дані

Таблиця констант

Змінна	Тип	Ім'я	Значення
Початок проміжку	Цілочисельний	FROM	1000
Кінець проміжку	Цілочисельний	TO	1000000

Таблиця функцій

Оператор	Назва	Синтаксис	Опис
log10	Логарифм за основою 10	log10(a)	Повертає Логарифм за основою 10 числа a
pow	Піднесення до степеню	pow(a,b)	Піднести a в степінь b
mod	Ділення з остачею	a mod b	Остача від ділення a на b

Задача зводиться до пошуку чисел виду $\overline{a_1a_2a_1a_2 \dots a_1a_2}$, $\overline{a_1a_2a_3 \dots a_1a_2a_3}$ і т.п., де повторюється частина $\overline{a_1a_2}$ і $\overline{a_1a_2a_3}$ відповідно. Можна помітити, що цифра a_1 повторюється кожний n -ий порядок, де n – це кількість цифр, у послідовності що повторюються.

Якщо поділити відповідне число на повторюваний фрагмент, то отримаємо число виду $(10 \dots) \dots 1$, де фрагмент $10 \dots$ має стільки нулів, скільки цифр в послідовності - 1, а сам фрагмент повторюється стільки раз, скільки кількість цифр у послідовності вміщується в числі цифр відповідного числа - 1.

Перебираючи числа з вказаного інтервалу потрібно створити всі можливі числа виду $(10 \dots) \dots 1$ для кількості чисел відповідного числа та перевірити, чи ділиться відповідне число на створене націло. Якщо так – число знайдено.

Для числа найменша кількість цифр у послідовності не може бути меншою 2 і більшою половини кількості цифр цього числа (у протилежному випадку кількість цифр утвореного числа була б більшою ніж у даного)

Розв'язання

Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді блок-схеми.

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо перебір чисел на заданому проміжку з використанням арифметичного оператору повторення.

Крок 3. Деталізуємо знаходження кількості повторюваних елементів в послідовності з використанням арифметичного оператору повторення та умовної форми вибору

Крок 4. Деталізуємо створення спеціального числа з використанням арифметичного оператору повторення.

Крок 5. Деталізуємо перевірку приналежності поточного числа з проміжку до множини чисел з повторюваними фрагментами послідовності цифр з використанням умовної форми вибору.

Псевдокод алгоритму

кrok 1

початок

перебір чисел на заданому проміжку

знаходження кількості повторюваних елементів в послідовності

створення спеціального числа

перевірка приналежності поточного числа до множини шуканих

вивід і

кінець

крок 2

початок

для i від FROM до TO , з кроком 1 повторити

знаходження кількості повторюваних елементів в послідовності

створення спеціального числа

перевірка приналежності поточного числа до множини шуканих

вивід i

все повторити

кінець

крок 3

початок

для i від FROM до TO, з кроком 1 повторити

num_of_digits := $\log_{10}(i) + 1$

для num_in_sequence від 2 до num_of_digits / 2, з кроком 1 повторити

якщо num_of_digits mod num_in_sequence == 0

то

створення спеціального числа

перевірка приналежності поточного числа до множини шуканих

вивід i

все якщо

все повторити

все повторити

кінець

крок 4

початок

для i від FROM до TO, з кроком 1 повторити

 num_of_digits := *log10(i)* + 1

для num_in_sequence від 2 до num_of_digits / 2, з кроком 1 повторити

якщо num_of_digits mod num_in_sequence == 0

то

 div_num := 1

для power_10 від num_in_sequence до num_of_digits,

з кроком num_in_sequence повторити

 div_num := div_num + *pow(10, power_10)*

все повторити

перевірка приналежності поточного числа до множини шуканих

вивід i

все якщо

все повторити

все повторити

кінець

крок 5

початок

для i від FROM до TO, з кроком 1 повторити

 num_of_digits := *log10*(i) + 1

для num_in_sequence від 2 до num_of_digits / 2, з кроком 1 повторити

якщо num_of_digits mod num_in_sequence == 0

то

 div_num := 1

для power_10 від num_in_sequence до num_of_digits,

з кроком num_in_sequence повторити

 div_num := div_num + *pow*(10, power_10)

все повторити

якщо i mod div_num == 0

то

вивід i

все якщо

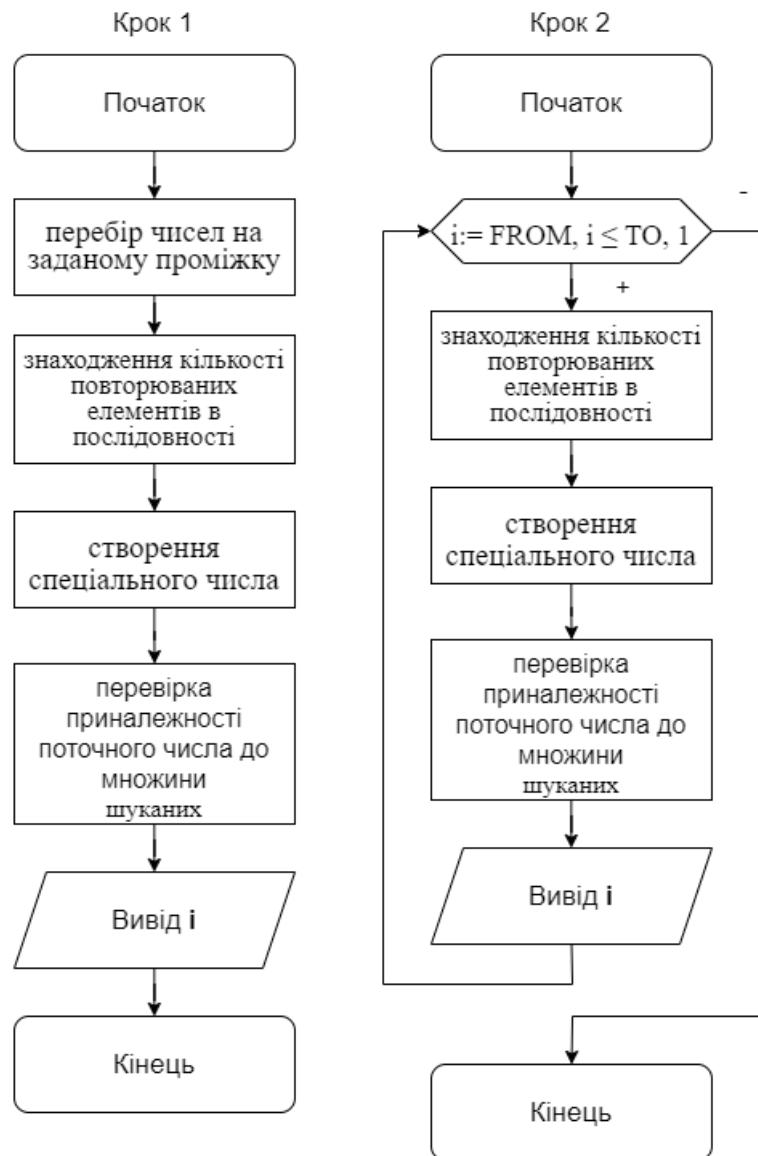
все якщо

все повторити

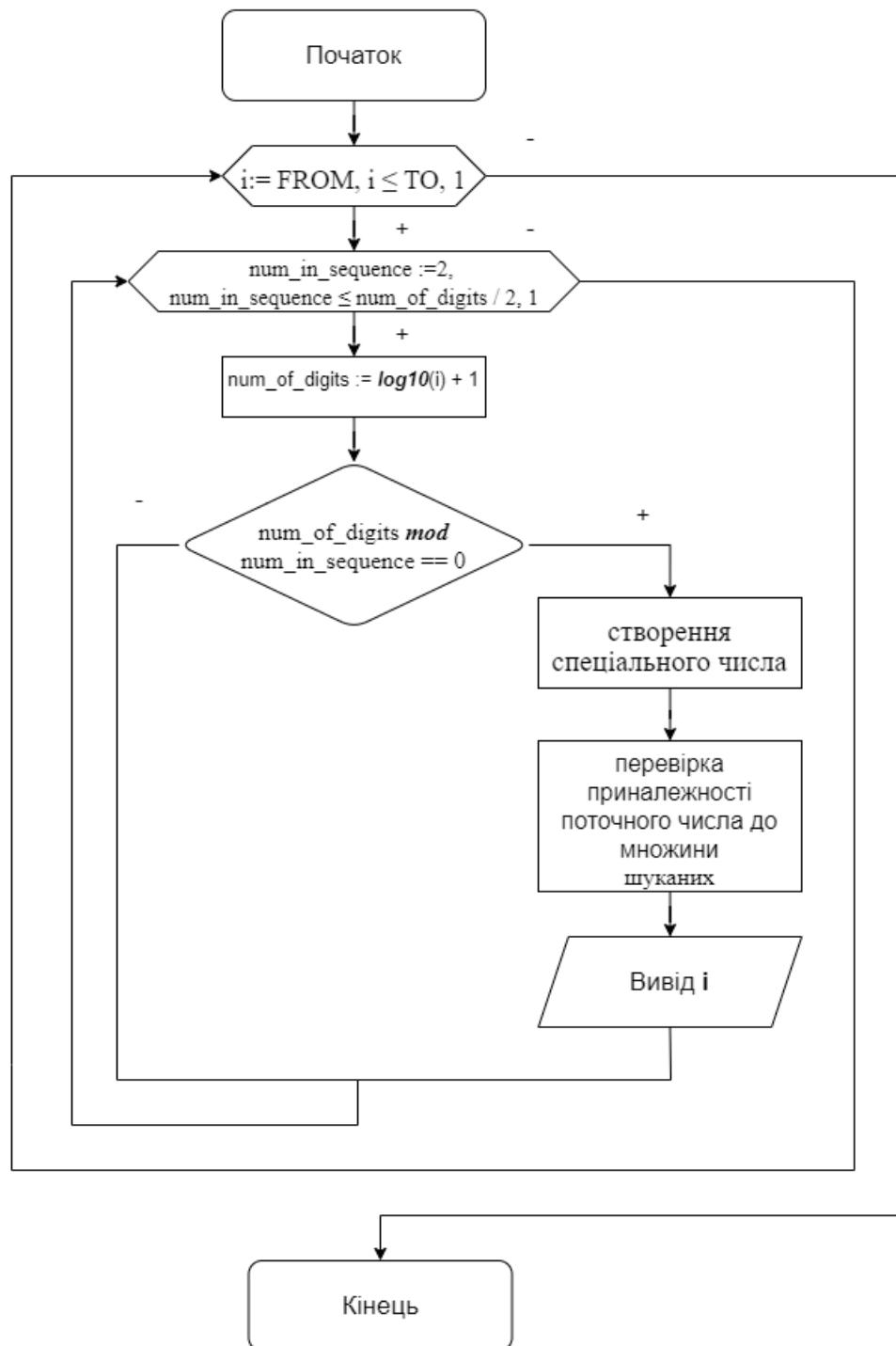
все повторити

кінець

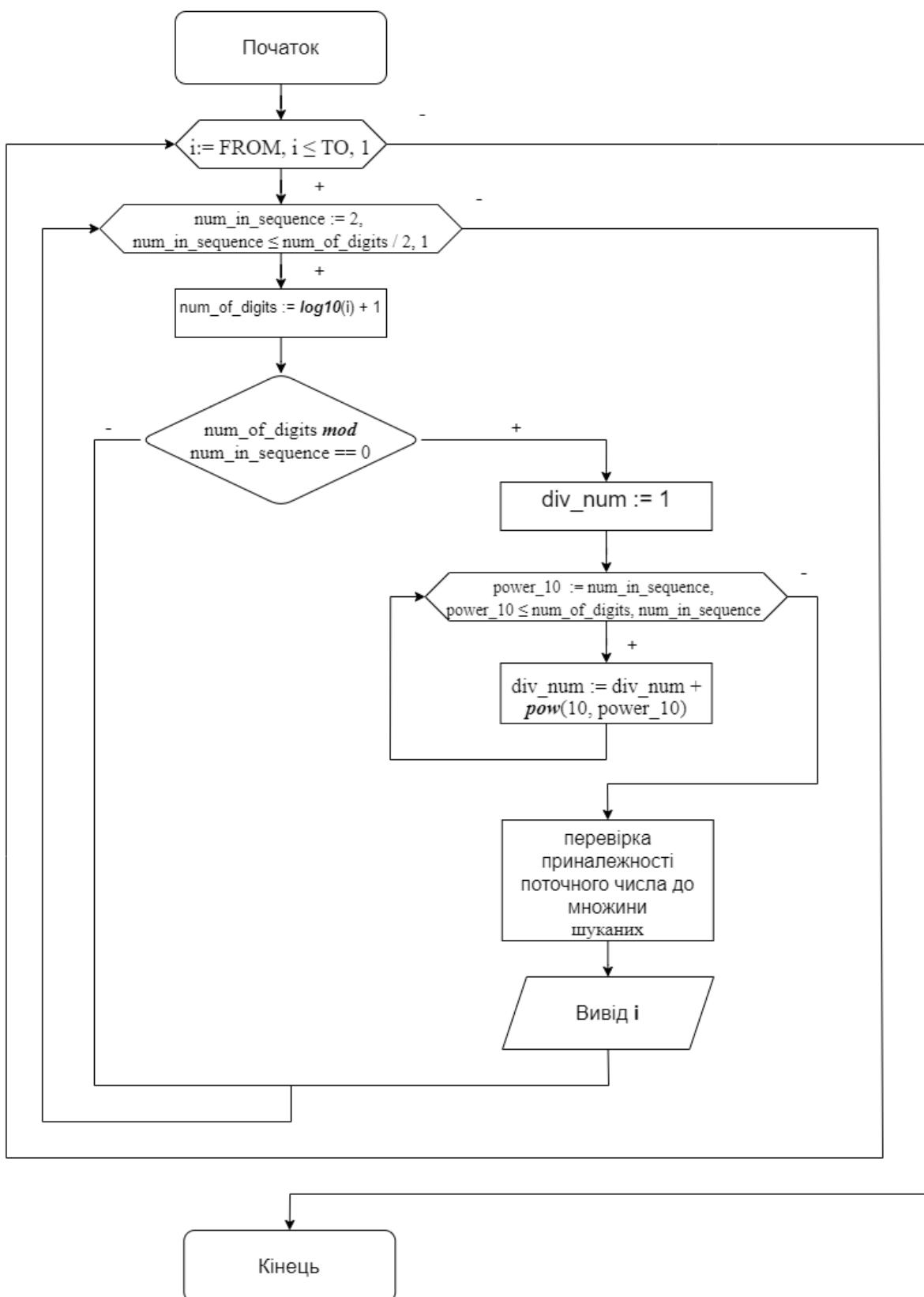
Блок-схема



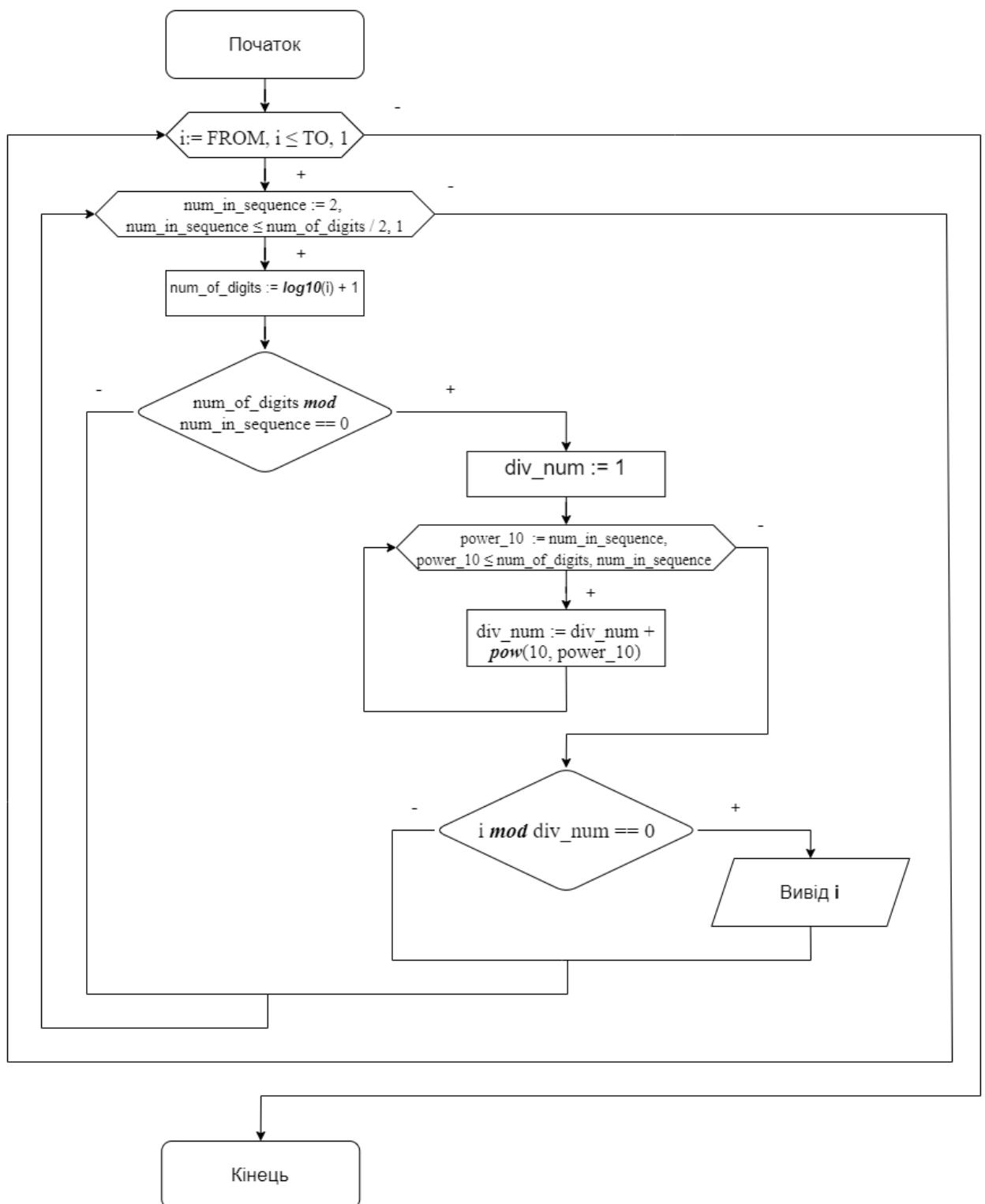
Крок 3



Крок 4



Крок 5



Випробування алгоритму

Блок	Дія
Початок	
1	i = 1000, i <= TO = true
2	num_of_digits = 4
3	num_in_sequence = 2
4	(num_in_sequence <= num_of_digits / 2) = true 2 <= 2 = true
5	(num_of_digits mod num_in_sequence == 0) = true 4 mod 2 == 0 = true
6	div_num = 1
7	power_10 = 2
8	power_10 < num_of_digits = true 2 < 4 = true
9	div_num = 101
10	power_10 = 4
11	power_10 < num_of_digits = false 4 < 4 = false
12	i mod div_num == 0 = false 1000 div 101 == 0 = false
13	num_in_sequence = 3
14	num_in_sequence <= num_of_digits / 2 = false 3 <= 2 = false
Кінець	

Блок	Дія
Початок	
1	$i = 1010, i \leq TO = true$
2	$num_of_digits = 4$
3	$num_in_sequence = 2$
4	$(num_in_sequence \leq num_of_digits / 2) = true$ $2 \leq 2 = true$
5	$(num_of_digits \bmod num_in_sequence == 0) = true$ $4 \bmod 2 == 0 = true$
6	$div_num = 1$
7	$power_10 = 2$
8	$power_10 < num_of_digits = true$ $2 < 4 = true$
9	$div_num = 101$
10	$power_10 = 4$
11	$power_10 < num_of_digits = false$ $4 < 4 = false$
12	$i \bmod div_num == 0 = true$ $1010 \bmod 101 == 0 = true$
13	Вивід: i
Кінець	

Висновки

На лабораторній роботі я дослідив особливості роботи складних циклів та набув практичних навичок їх використання під час складання програмних специфікацій. Спочатку я використав арифметичний оператор повторення для перебору значень на інтервалі. Далі я використав вкладений арифметичний оператор повторення та умовну форму вибору для визначення кількості можливих членів у послідовності. За допомогою ще одного арифметичного оператора повторення я створив спеціальне число і за його допомогою в умовній формі вибору визначав, чи належить воно до чисел з повторюваними фрагментами цифр.