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 28

Виконав студент ІП-11 Сідак Кирил Ігорович
(шифр, прізвище, ім'я, по батькові)

Перевірив Мартинова Оксана Петрівна
(прізвище, ім'я, по батькові)

Київ 2021

Лабараторна робота№5

Дослідження складних циклічних алгоритмів

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

Індивідуальне завдання:

Варіант 28

Отримати всі піфагорові трійки натуральних чисел, кожне з яких не перевищує n, тобто всі такі трійки натуральних чисел a, b, c, що a^2+ b^2= c^2 (a ≤, b ≤, c ≤ n).

Постановка задачі: для кожного натурального числа a та b, кожне з яких не більше заданого числа n, треба знайти таке число c (c не більше n), що a^2 + b^2 = c^2. Тобто, якщо c = sqrt(a^2+b^2)– ціле число, то a, b, c – одна з шуканих піфагорових трійок чисел.

Побудова математичної моделі

Складемо таблицю змінних

Змінна	Тип	Ім’я	Призначення
Задане натуральне число, яке обмежує значення піфагорових трійок	цілий	n	Вхідне дане
Перше число з піфагорової трійки	цілий	a	Проміжне дане
Друге число з піфагорової трійки	цілий	b	Проміжне дане
Число, квадрат якого є сумою квадратів першого і другого числа піфагорової трійки.	дійсний	c	Результат

Таким чином, формування задачі зводиться до обчислення значення c для кожного натурального a від 1 до n включно та відповідного йому b від a+1 до n включно, c = sqrt(a^2+b^2). Якщо c – ціле число ([c] == c) та c <= n, то значення a, b, для яких це виконується є шуканими, тобто знайдені числа a, b, c утворюють піфагорову трійку. Отже, в результаті перевірки для кожного можливого a та b отримаємо всі можливі піфагорові трійки, в яких жодне з чисел не перевищує n.

[] – ціла частина від числа

sqrt() – квадратний корінь числа

Розв’язання

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

Крок 1. Визначимо основні дії

Крок 2. Деталізуємо перебір першого числа

Крок 3. Деталізуємо перебір другого числа

Крок 4. Деталізуємо визначення шуканої піфагорової трійки.

Псевдокод

Крок 1

Початок

        <u>Перебір першого числа</u>

        Перебір другого числа

        Визначення шуканої піфагорової трійки

Кінець

Крок 2

Початок

        повторити

          для a **від** 1 **до** n

            <u>Перебір другого числа</u>

            Визначення шуканої піфагорової трійки

        все повторити

Кінець

Крок 3

Початок

повторити

для a **від** 1 **до** n

повторити

для b **від** a+1 **до** n

Визначення шуканої піфагорової трійки

все повторити

все повторити

Кінець

Крок 4

Початок

повторити

для a **від** 1 **до** n

повторити

для b **від** a+1 **до** n

c = sqrt(a^2 + b^2)

якщо c == [c] **і** c <= n

то

виведення "a = ", a, " b = ", b," c = ", c, "\n"

все якщо

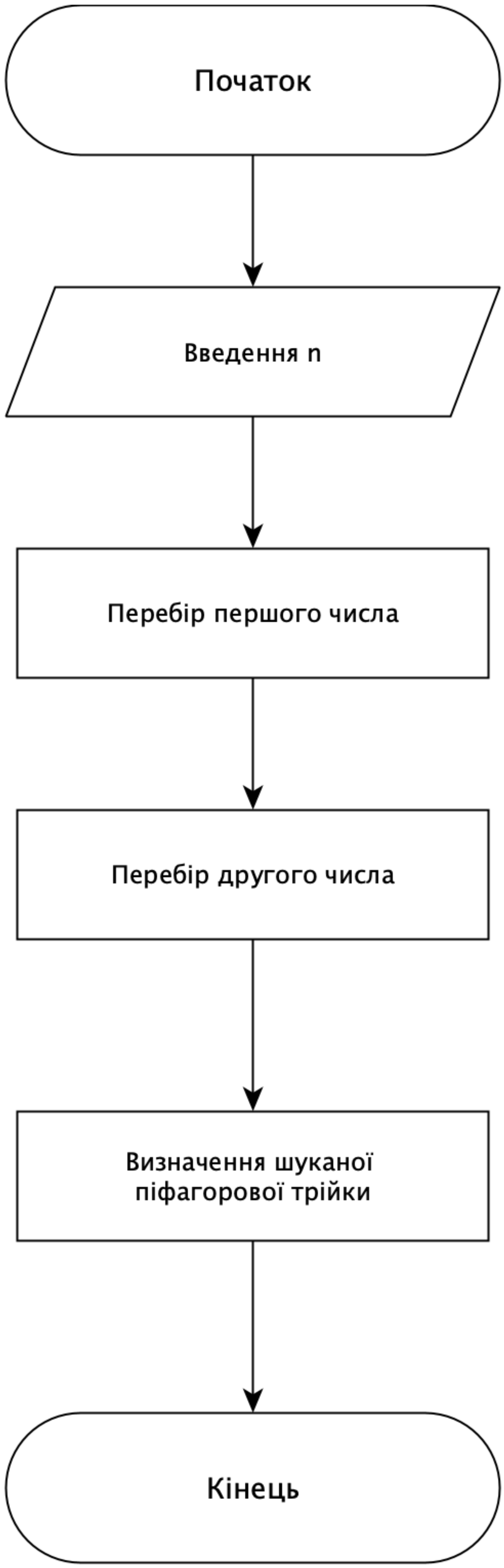
все повторити

все повторити

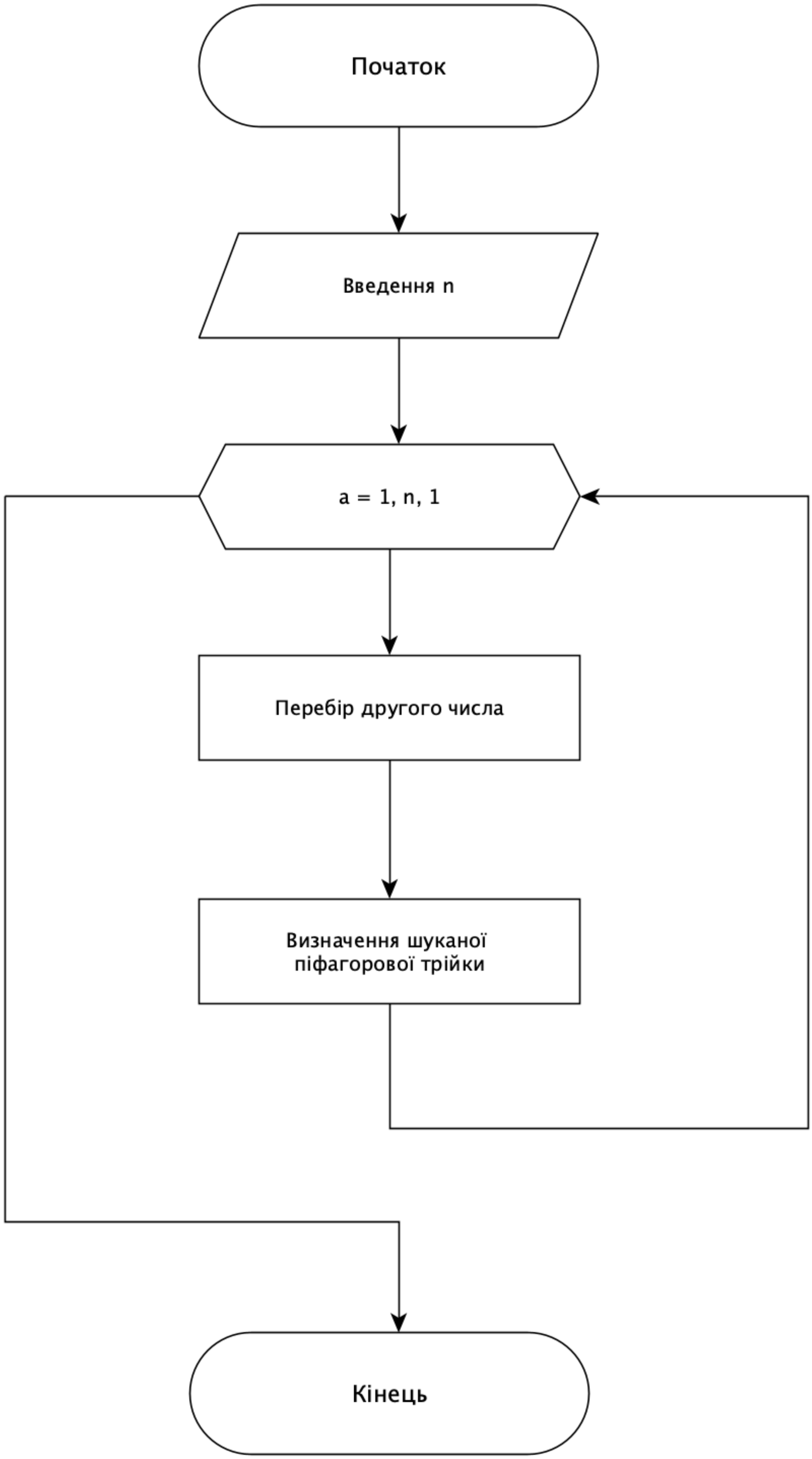
Кінець

Блок-схема

Крок 1

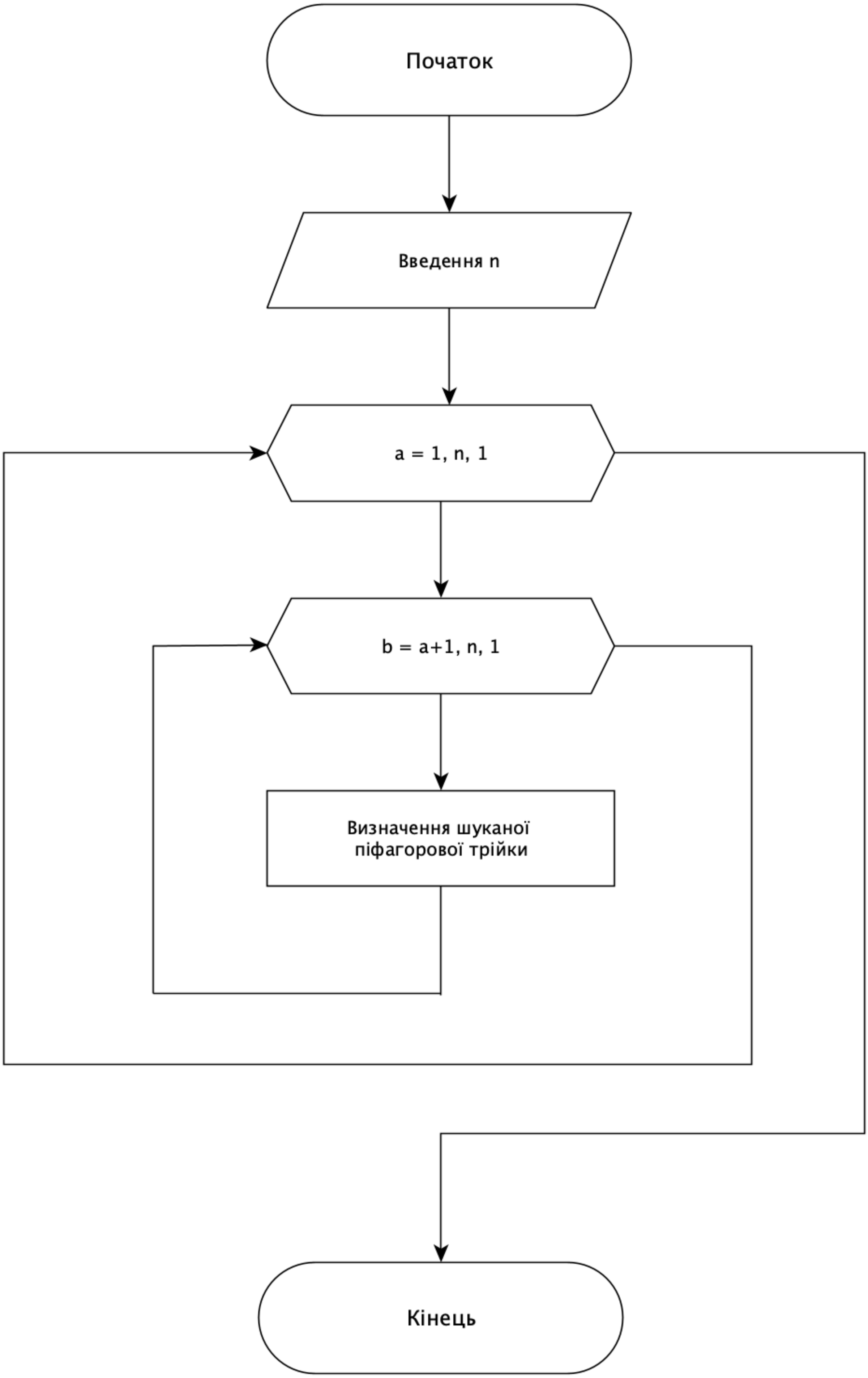


Крок 2

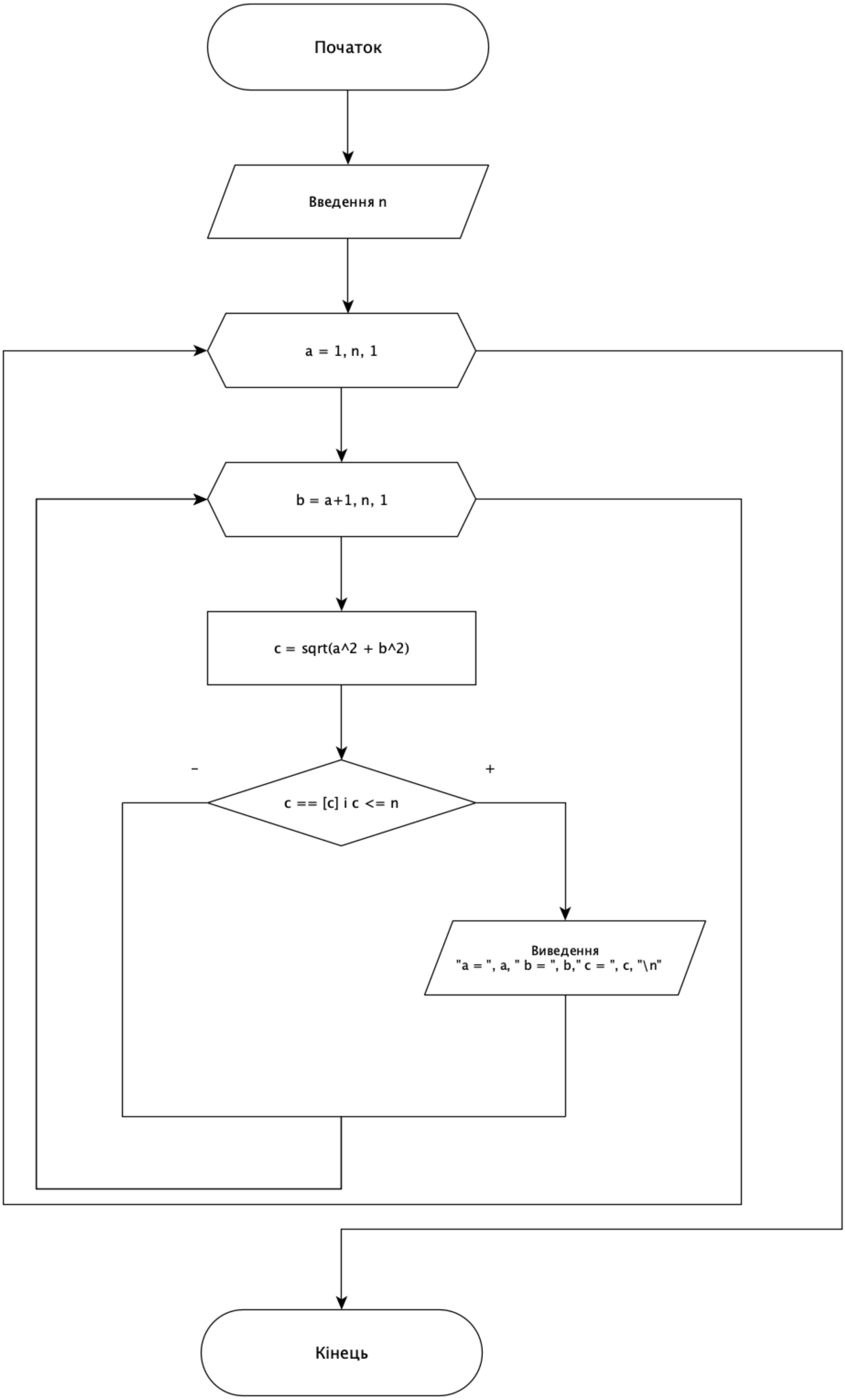


Початок
Введення n
a = 1, n, 1
Перебір другого числа
Визначення шуканої піфагорової трійки
Кінець

Крок 3


Початок
Введення n
a = 1, n, 1
b = a+1, n, 1
Визначення шуканої піфагорової трійки
Кінець

Крок 4



Початок
Введення n
a = 1, n, 1
b = a+1, n, 1
c = sqrt(a^2 + b^2)
-
+
c == [c] і c <= n
Виведення
"a = ", a, " b = ", b," c = ", c, "\n"
Кінець

Випробування алгоритму

Блок	Дія
	Початок
1	Введення n = 20
2	Після 3 ітерації для a та 1 ітерації для b: a = 3, b = 4, c = 5 c = [c] - так Виведення "a = 3, b = 4, c = 5"
4	Після 5 ітерації для a та 7 ітерації для b: a = 5, b = 12, c = 13 c = [c] - так Виведення "a = 5, b = 12, c = 13"
5	Після 6 ітерації для a та 2 ітерації для b: a = 6, b = 8, c = 10 c = [c] - так Виведення "a = 6, b = 8, c = 10"
6	Після 8 ітерації для a та 7 ітерації для b: a = 8, b = 15, c = 17 c = [c] - так Виведення "a = 8, b = 15, c = 17"
7	Після 9 ітерації для a та 3 ітерації для b: a = 9, b = 12, c = 15 c = [c] - так Виведення "a = 9, b = 12, c = 15"
8	Після 12 ітерації для a та 4 ітерації для b: a = 12, b = 16, c = 20 c = [c] - так Виведення "a = 12, b = 16, c = 20"
	Кінець

Висновок

Отже, я дослідив особливості складних циклів, використавши два арифметичні цикли, один вкладений в інший, для знаходження усіх можливих піфагорових трійок чисел із заданим обмеженням та отримав коректний результат.