

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 № 4 з дисципліни
«Алгоритми та структури даних-1. Основи
алгоритмізації»

«Дослідження лінійних алгоритмів»

Варіант 11

Виконав студент ІП-13, Дем'янчук Олександр Петрович

(шифр, прізвище, ім'я, по батькові)

Перевірів

(прізвище, ім'я, по батькові)

Лабораторна робота 4

Дослідження арифметичних циклічних алгоритмів

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

Індивідуальне завдання

Варіант 11

Завдання

Визначити n -не число Каталана. Кожне число Каталана, починаючи з третього, обчислюється за рекурентною формулою:

$$k_0 = k_1 = 1, \quad k_n = \frac{k_{n-1}(4n-6)}{n}.$$

1. Постановка задачі

Використовуючи арифметичний цикл, знайти значення n -ного числа Каталана — lastK, де n -ний член вираховується за формулою $\text{lastK} = (\text{prevK} * (4n-6)) / n$. Цикл завершується тоді, коли n досягає заданого значення.

2. Математична модель

Побудуємо таблицю імен змінних:

<i>Змінна</i>	<i>Тип</i>	<i>Ім'я</i>	<i>Призначення</i>
Порядок числа Каталана	Цілий	n	Вхідні дані
Арифметичний параметр	Цілий	i	Проміжні дані
Перше число Каталана	Дійсний	zeroK	Проміжні дані
Друге число Каталана	Дійсний	firstK	Проміжні дані
n -не число Каталана	Дійсний	lastK	Вихідні дані

n - кінцеве значення арифметичного параметра;

i - власне арифметичний параметр: **i1=3**, **i2=n**, **i3=1**;

zeroK є сталим значенням і дорівнює 1;

firstK є сталим значенням і дорівнює 1;

lastK обчислюємо за формулою **lastK = (prevK*(4i-6))/i**;

Розв'язання

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

Крок 1. Визначимо основні дії

Крок 2. Деталізуємо дію знаходження першого значення **prevK** як **firstK**

Крок 2. Деталізуємо дію обчислення числа Каталана як арифметичний цикл

Псевдокод

Крок 1

початок

введення **n**

визначення **prevK** як другого елемента **firstK**

цикл знаходження **lastK**

виведення **lastK**

кінець

Крок 2

початок

введення **n**

firstK := 1

prevK := **firstK**

цикл знаходження **lastK**

виведення **y**

кінець

Крок 3

початок

введення **n**

zeroK := 1

prevY := **firstK**

повторити **i** раз **P** **кінець_циклу**

lastK := (**prevK** * (**4i** - 6)) / **i**

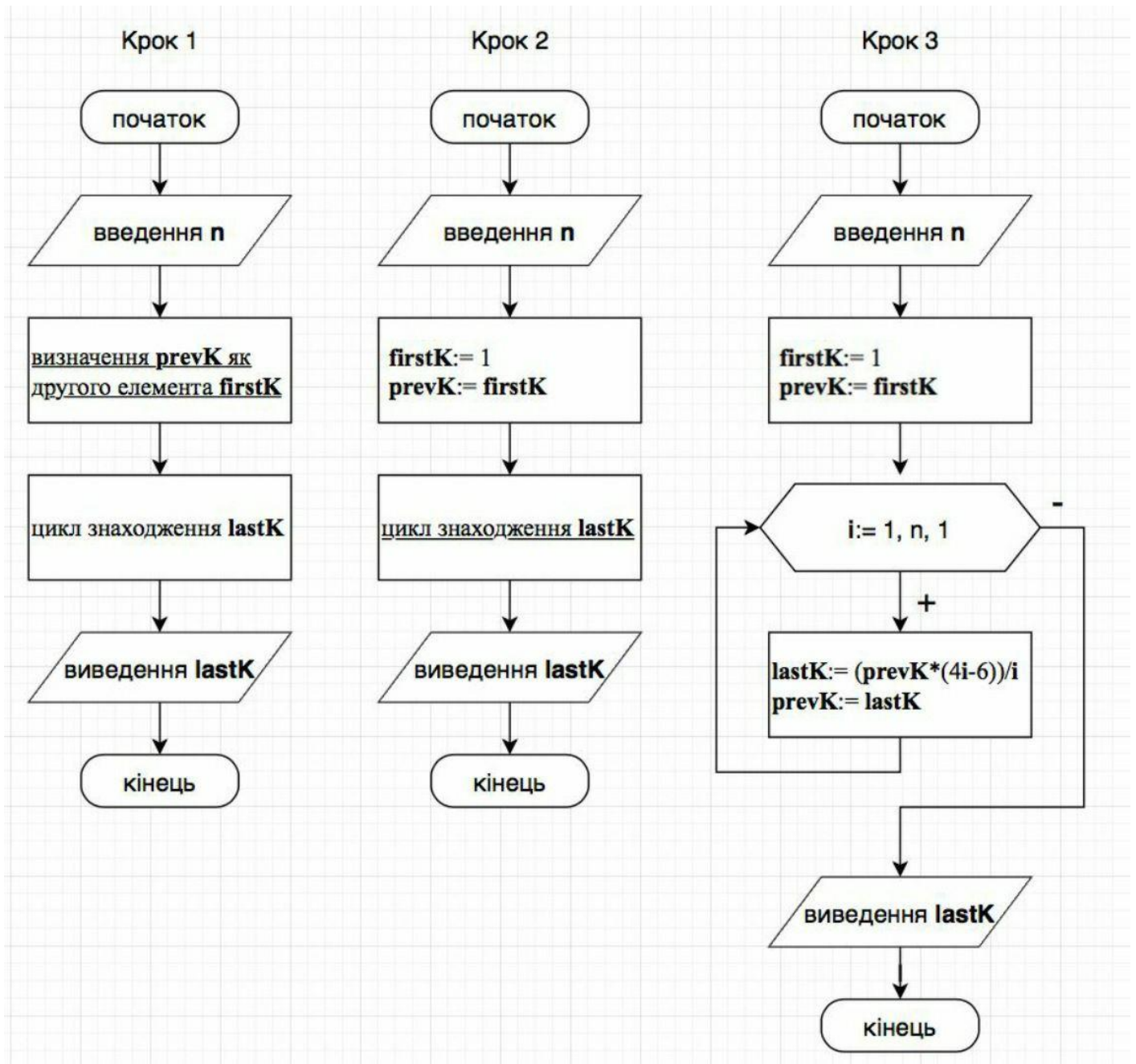
prevK := **lastK**

все повторити

виведення **lastY**

кінець

Блок-схема



Тестування

Блок	Дія
	Початок
1	Введення n = 9
2	firstK = 1; prevK = firstK
3	i = 3, 9, 1
4	lastK = (1*(4*3-6))/3 = 2 prevK = lastK

Основи_програмування – 1. Алгоритми та структури даних

5	i = 4, 9, 1
6	lastK = (2*(4*4-6))/4 = 5 prevK = lastK
7	i = 5, 9, 1
8	lastK = (5*(4*5-6))/5 = 14 prevK = lastK
9	i = 6, 9, 1
10	lastK = (14*(4*6-6))/6 = 42 prevK = lastK
11	i = 7, 9, 1
12	lastK = (42*(4*7-6))/7 = 132 prevK = lastK
13	i = 8, 9, 1
14	lastK = (132*(4*8-6))/8 = 429 prevK = lastK
15	i = 9, 9, 1
16	lastK = (429*(4*9-6))/9 = 1430 prevK = lastK
17	Виведення lastK = 1430
	Кінець

Висновок

На лабораторній роботі дослідив особливості роботи арифметичних циклів, навчився створювати їх та використовувати під час складання програмних специфікацій.