

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 __16__

Виконав студент __ІП-15, Куманецька І. В.____
Перевірив _Вечерковська А. С._____

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.

Індивідуальне завдання

Варіант 16

16. З чисел Фібоначі в інтервалі від 1 до 100 визначити тільки прості числа, а також їх порядкові номери в ряду Фібоначі.

Постановка задачі

Знайти числа Фібоначі в інтервалі від 1 до 100 та визначити прости серед них. Результатом виведення має бути саме число та його порядковий номер в ряду Фібоначі.

Побудова математичної моделі

Змінна	Тип	Ім'я	Призначення
Можливі дільники числа	Ціле натуральне	i	Проміжні дані
Сума усіх дільників числа	Ціле натуральне	sum_i	Проміжні дані
Максимальне число Фібоначі	Ціле натуральне	max_fib	Вхідні дані
Передостаннє знайдене число Фібоначі	Ціле натуральне	sec_last	Проміжні дані
Останнє знайдене число Фібоначі	Ціле натуральне	last	Проміжні та кінцеві дані

Нове підраховане число Фібоначі	Ціле натуральне	new	Проміжні дані
Номер останнього числа Фібоначі в загальному ряду	Ціле натуральне	num_new	Проміжні та кінцеві дані

Вводяться початкові значення $num_sec=0$ та $num_last=sec_last=last=1$, після чого кожне наступне нове число з ряду Фібоначі розраховується як сума двох попередніх. У зовнішньому циклі замінюються значення num_new , sec_last , $last$ та new , у внутрішньому циклі перевіряється умова, чи є знайдене число простим. Через $\text{sqrt}()$ позначається знаходження квадратного кореня від числа. Через $a+=b$ позначається операція $a = b + a$. Через $\%$ позначається операція знаходження остачі від ділення.

Розв'язання

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

Крок 1. Визначимо основні дії.

Крок 2. Створення і присвоєння початкових значень $num_new=sec_last=last=new=1$.

Крок 3. Деталізуємо дію розрахування нового new та заміни значень $last$, num_new , sec_last , присвоєння початкового значення i та sum_i .

Крок 4. Деталізуємо дію перевірки умови, чи є new простим числом.

Псевдокод

Крок 1

початок

створення i присвоєння початкових значень

num_new=sec_last=last=new=1

розрахування нового new та заміни значень last, num_new, sec_last,
присвоєння початкового значення i та sum_i

перевірка умови, чи є new простим числом

виведення new та num_new

кінець

Крок 2

початок

new := 1

num_new := 1

sec_last := 1

last := 1

розрахування нового new та заміни значень last, num_new, sec_last,
присвоєння початкового значення i та sum_i

перевірка умови, чи є new простим числом

виведення new та num_new

кінець

Крок 3

початок

new := 1

num_new := 1

sec_last := 1

last := 1

поки (last + sec_last) <= max_fib

повторити

new = last + sec_last

num_new += 1

sec_last = last

last = new

sum_i := 0

перевірка умови, чи є new простим числом

все повторити

виведення new та num_new

кінець

Крок 4

початок

new := 1

num_new := 1

sec_last := 1

last := 1

поки (last + sec_last) <= max_fib

повторити

$new = last + sec_last$

$num_new += 1$

$sec_last = last$

$last = new$

$sum_i := 0$

повторити

для i від 1 до \sqrt{new}

якщо $new \% i = 0$

$sum_i += i$

все якщо

все повторити

якщо $sum_i = 1$

виведення new та num_new

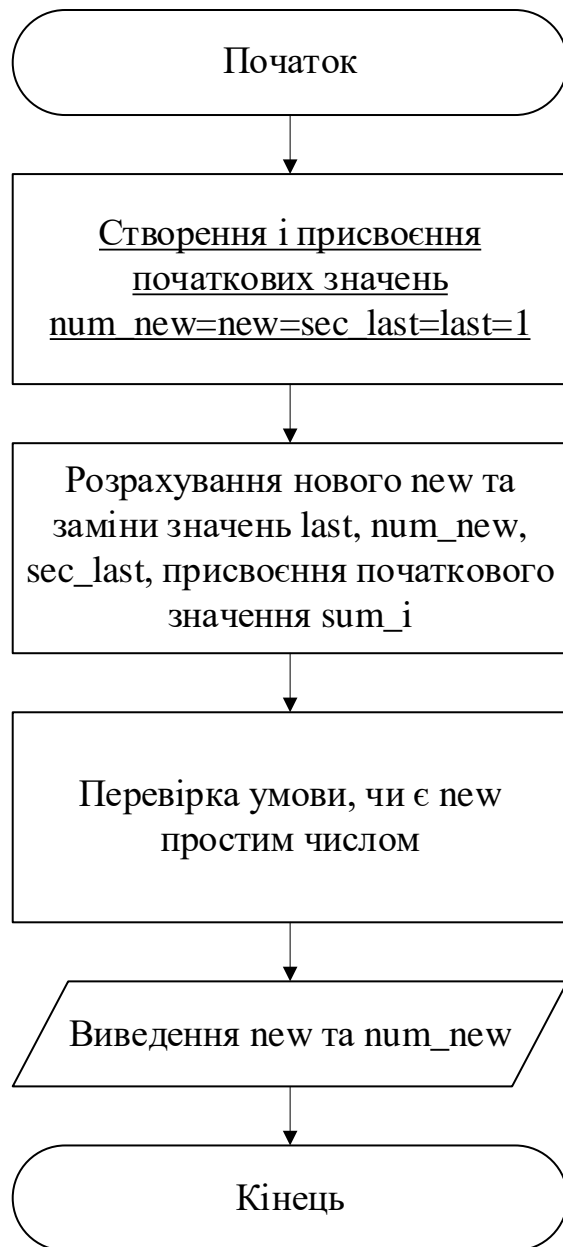
все якщо

все повторити

кінець

Блок-схема

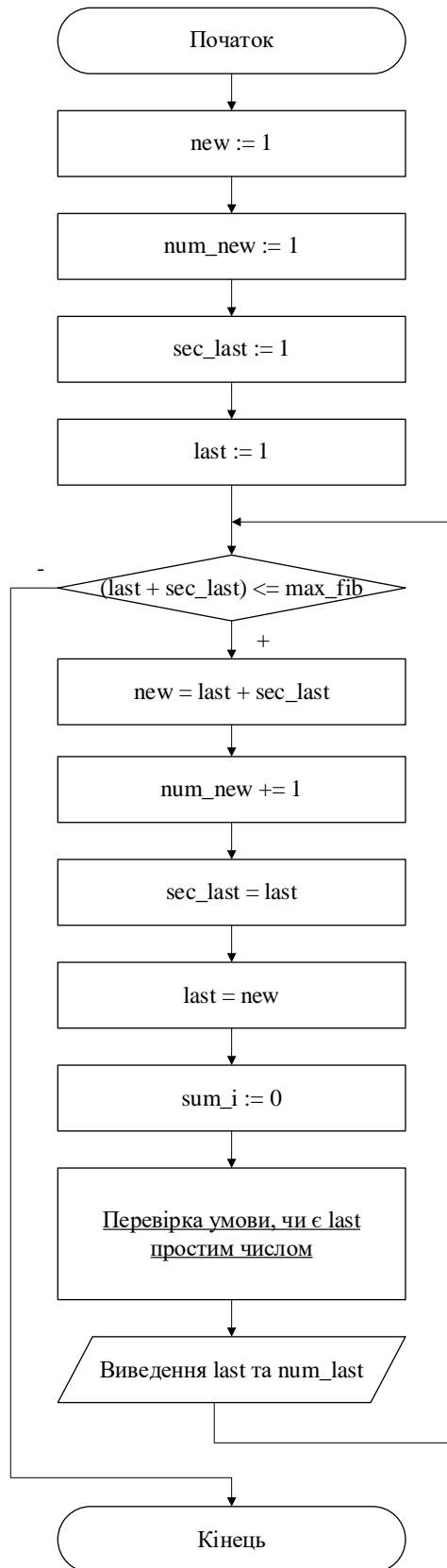
Крок 1



Крок 2

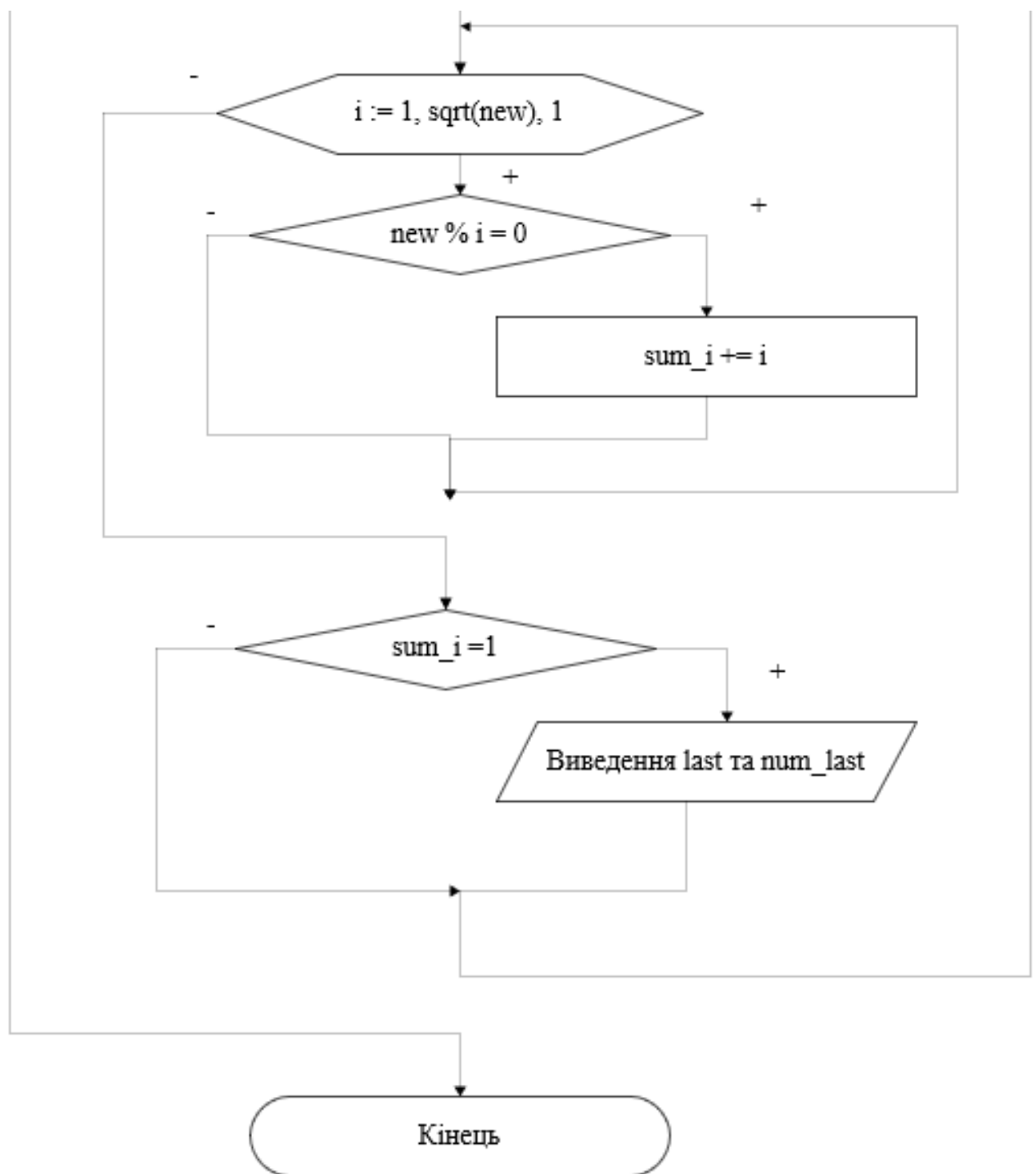


Крок 3



Крок 4





Випробування

Блок	Дія
	Початок
1	new := 1
2	num_new := 1
3	sec_last := 1
4	last := 1
5 (зовнішній цикл 1)	2 <= 100 істина new = 2 num_new = 2 sec_last = 1 last = 2 sum_i := 0
(внутрішній цикл 1)	i = 1 2 % 1 = 0 істина sum_i = 1
(зовнішній цикл 1)	sum_i = 1 істина виведення 2 та 2
6 (зовнішній цикл 2)	3 <= 100 істина new = 3 num_new = 3 sec_last = 2 last = 3 sum_i := 0
(внутрішній цикл 1)	i = 1 3 % 1 = 0 істина sum_i = 1

(зовнішній цикл 2)	sum_i = 1 істина виведення 3 та 3
7 (зовнішній цикл 3)	5 <= 100 істина new = 5 num_new = 4 sec_last = 3 last = 5 sum_i := 0
(внутрішній цикл 1)	i = 1 5 % 1 = 0 істина sum_i = 1
(внутрішній цикл 2)	i = 2 5 % 2 = 0 хиба
(зовнішній цикл 3)	sum_i = 1 істина виведення 5 та 4
8 (зовнішній цикл 4)	8 <= 100 істина new = 8 num_new = 5 sec_last = 5 last = 13 sum_i := 0
(внутрішній цикл 1)	i = 1 5 % 1 = 0 істина sum_i = 1
(внутрішній цикл 2)	i = 2 5 % 2 = 0 істина sum_i = 3

(зовнішній цикл 4)	sum_i = 1 хиба
9 (зовнішній цикл 5)	13 <= 100 істина new = 13 num_new = 6 sec_last = 8 last = 13 sum_i := 0
(внутрішній цикл 1)	i = 1 13 % 1 = 0 істина sum_i = 1
(внутрішній цикл 2)	i = 2 13 % 2 = 0 хиба
(внутрішній цикл 3)	i = 3 13 % 3 = 0 хиба
(зовнішній цикл 5)	sum_i = 1 істина виведення 13 та 6
...	
13 (зовнішній цикл 9)	89 <= 100 істина new = 89 num_new = 10 sec_last = 55 last = 89 sum_i := 0
(внутрішній цикл 1)	i = 1 89 % 1 = 0 істина sum_i = 1
(внутрішній цикл 2)	i = 2

	89 % 2 = 0 хиба
...	
(внутрішній цикл 9)	i = 9 89 % 9 = 0 хиба
(зовнішній цикл 9)	sum_i = 1 істина виведення 89 та 10
14	144 <= 100 хиба
	Кінець

Висновок

Було досліджено особливості роботи складних циклів та набуто практичних навичок їх використання під час складання програмних специфікацій. В результаті виконання лабораторної роботи було знайдено прості числа Фібоначі від 1 до 100 та їх номери в ряду, розділивши задачу на 4 кроки: визначення основних дій, створення і присвоєння початкових значень num_new=sec_last=last=new=1, деталізація дії розрахування нового new та заміни значень last, num_new, sec_last, присвоєння початкового значення i та sum_i, деталізація дії перевірки умови, чи є new простим числом. В процесі випробування було розраховано результат: 2 2, 3 3, 5 4, 13 6, 89 10.