

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 23

Виконав студент ПІ-13 Недельчев Євген Олександрович
(шифр, прізвище, ім'я, по батькові)

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

Київ 20211

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 23

23. Для чисел, що належать діапазону $[a, b]$ визначити дільники, що є членами послідовності Фібоначі.

Постановка задачі

Заданий діапазон $[a, b]$. Розробити універсальний алгоритм, результатом роботи якого є пошук та виведення дільників чисел у діапазоні $[a, b]$, які є членами послідовності Фібоначі.

Побудова математичної моделі

<i>Змінна</i>	<i>Тип</i>	<i>Ім'я</i>	<i>Призначення</i>
Нижня границя діапазону	Цілий	a	Вхідні дані
Верхня границя діапазону	Цілий	b	Вхідні дані
n-те число ряду Фібоначі	Цілий	fib1	Результат
(n+1) число ряду Фібоначі	Цілий	fib2	Результат
Лічильник у вкладеному циклі	Цілий	i	Лічильник
Тимчасова змінна для коректної зміни значень fib1 та fib2	Цілий	temp	Тимчасова змінна
Тимчасова змінна, за допомогою якої уникається повторення дільників	Цілий	last	Тимчасова змінна

Розв'язання

Крок 1. Визначимо основні дії.

Крок 2. Визначимо ряд Фібоначчі до числа b

Крок 3. Перевіримо, чи є взятий елемент ряду Фібоначчі дільником кожного елемента послідовності $[a, b]$

Крок 4. Перевірка останнього $fib2$ для запобігання повторення дільників

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

Псевдокод

Крок 1.

початок

введення змінних a та b

визначення ряду Фібоначчі до числа b

перевірка, чи є взятий елемент ряду Фібоначчі дільником кожного елемента послідовності $[a, b]$

перевірка останнього $fib2$ для запобігання повторення дільників

кінець

Крок 2.

початок

введення змінних a та b

$fib1 := 1$

$fib2 := 1$

$last := 0$

виведення 1 // 1 є дільником будь-якого числа та належить ряду Фібоначчі

повторити

$temp := fib1$

$fib1 := fib2$

$fib2 += temp$

перевірка, чи є взятий елемент ряду Фібоначчі дільником кожного елемента послідовності $[a, b]$

перевірка останнього $fib2$ для запобігання повторення дільників

поки $fib2 < b$

все повторити

кінець

Крок 3.

початок

введення змінних a та b

$fib1 := 1$

$fib2 := 1$

$last := 0$

виведення 1 // 1 є дільником будь-якого числа та належить ряду Фібоначчі

повторити

```
temp := fib1
fib1 := fib2
fib2 += temp
повторити для i від a до b+1
    якщо i % fib2 == 0
        перевірка останнього fib2
    все якщо
поки fib2 < b
все повторити
кінець
```

Крок 4.

початок

введення змінних a та b

fib1 := 1

fib2 := 1

last := 0

виведення 1 // 1 є дільником будь-якого числа та належить ряду Фібоначчі

повторити

temp := fib1

fib1 := fib2

fib2 += temp

повторити для i від a до b+1

якщо i % fib2 == 0

якщо last != fib2

виведення fib2

last := fib2

все якщо

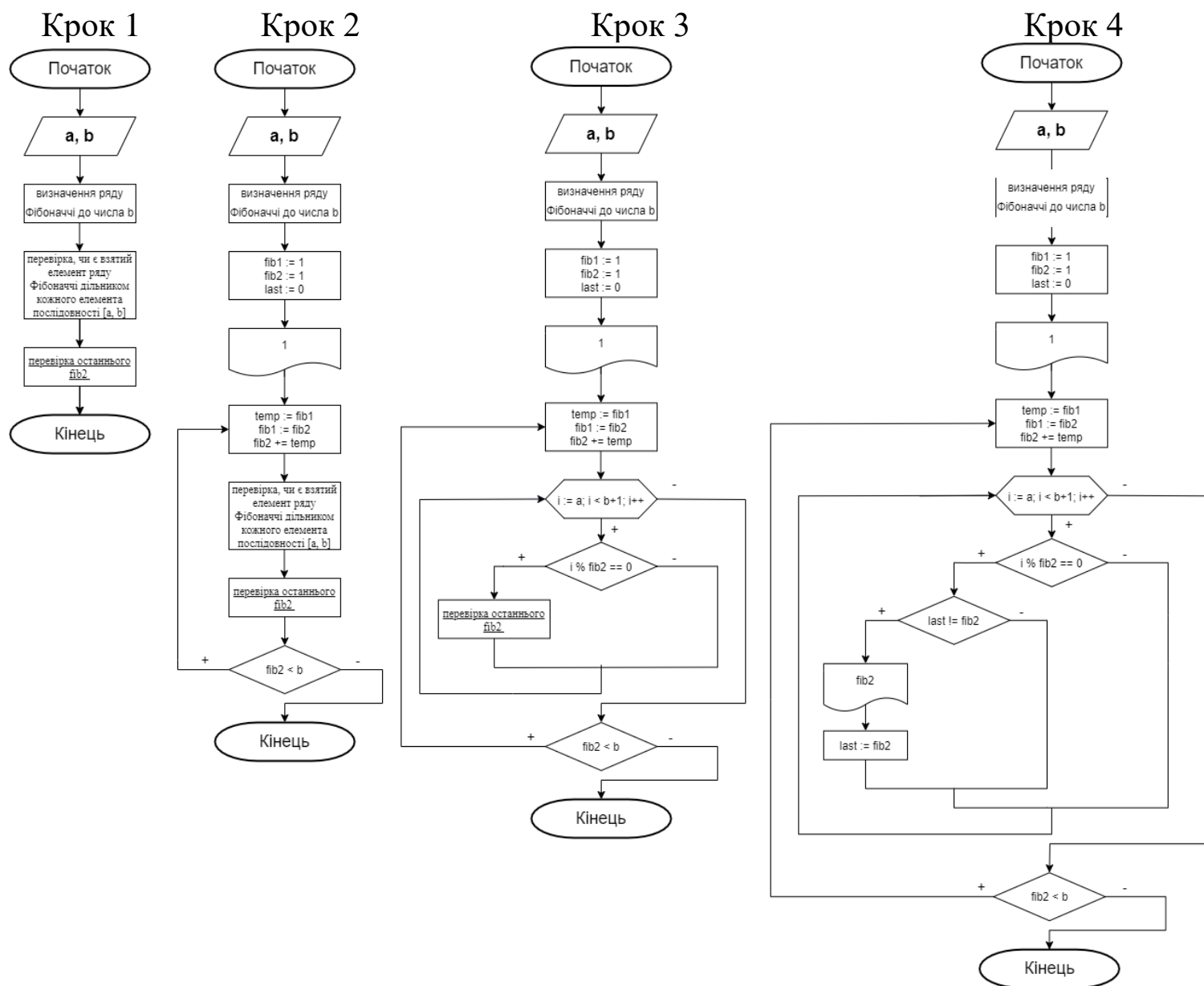
все якщо

поки fib2 < b

все повторити

кінець

Блок-схема



Тестування алгоритму

Блок	Дія
	Початок
1	Введення: $a = 10$ $b = 35$
	Виведення: 1
2	$temp = 1$
	$fib1 = 1$
	$fib2 = 2$
	$10 \% 2 == 0 = true$
	Виведення: 2
	$fib2 < b = true$
....
9	$temp = 21$
	$fib1 = 34$
	$fib2 = 55$
	$fib2 < b = false$
10	Кінець

Висновки

Під час роботи я дослідив особливості роботи складних циклів та набув практичних навичок їх використання під час складання програмних специфікацій.