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**



**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 | і | 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 sqrt() позначається знаходження квадратного кореня від числа. Через a+=b позначається операція a = b + а. 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 і та sum\_i.

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

**Псевдокод**

*Крок 1*

**початок**

створення і присвоєння початкових значень num\_new=sec\_last=last=new=1

розрахування нового new та заміни значень last, num\_new, sec\_last, присвоєння початкового значення і 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 і 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** і **від** 1 **до** sqrt(new)

**якщо** new % і = 0

sum\_i += і

**все якщо**

**все повторити**

**якщо** sum\_i = 1

виведення new та num\_new

**все якщо**

**все повторити**

**кінець**

**Блок-схема**

*Крок 1*

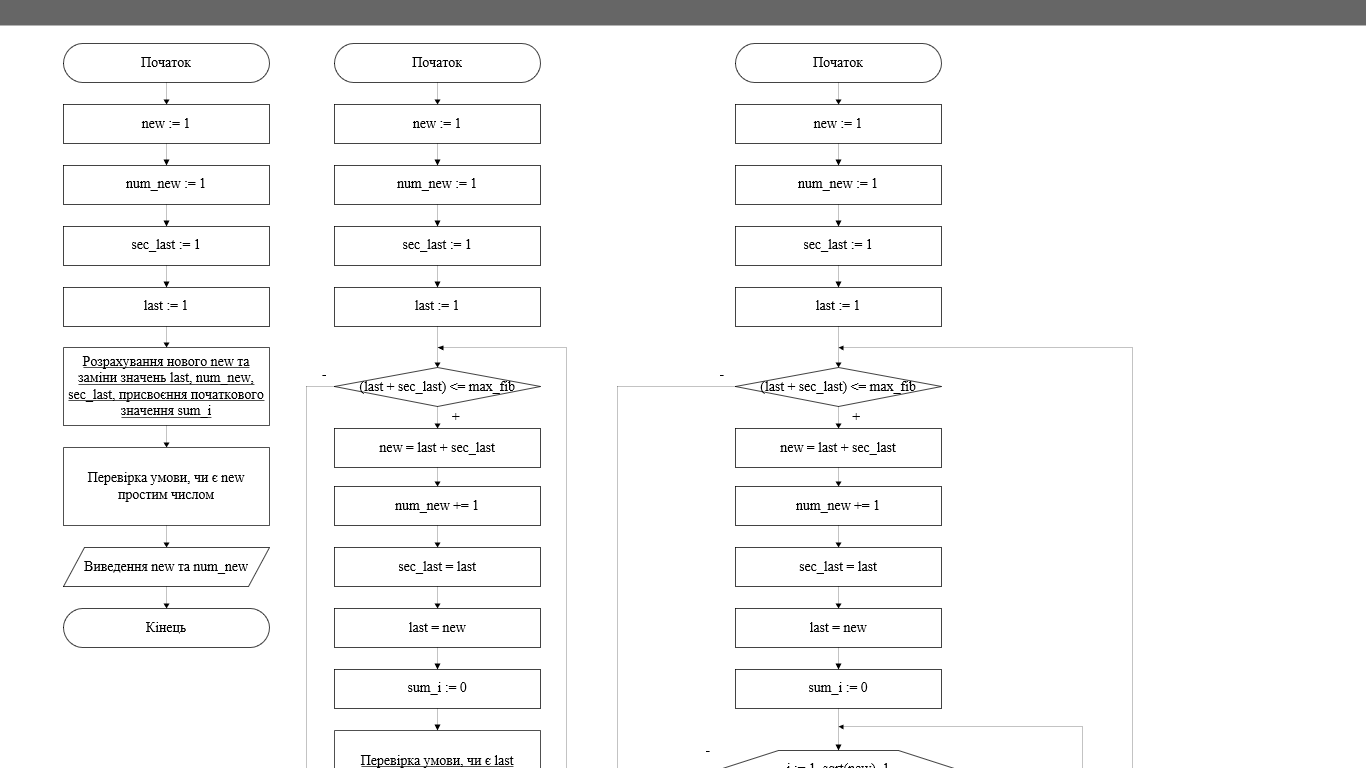


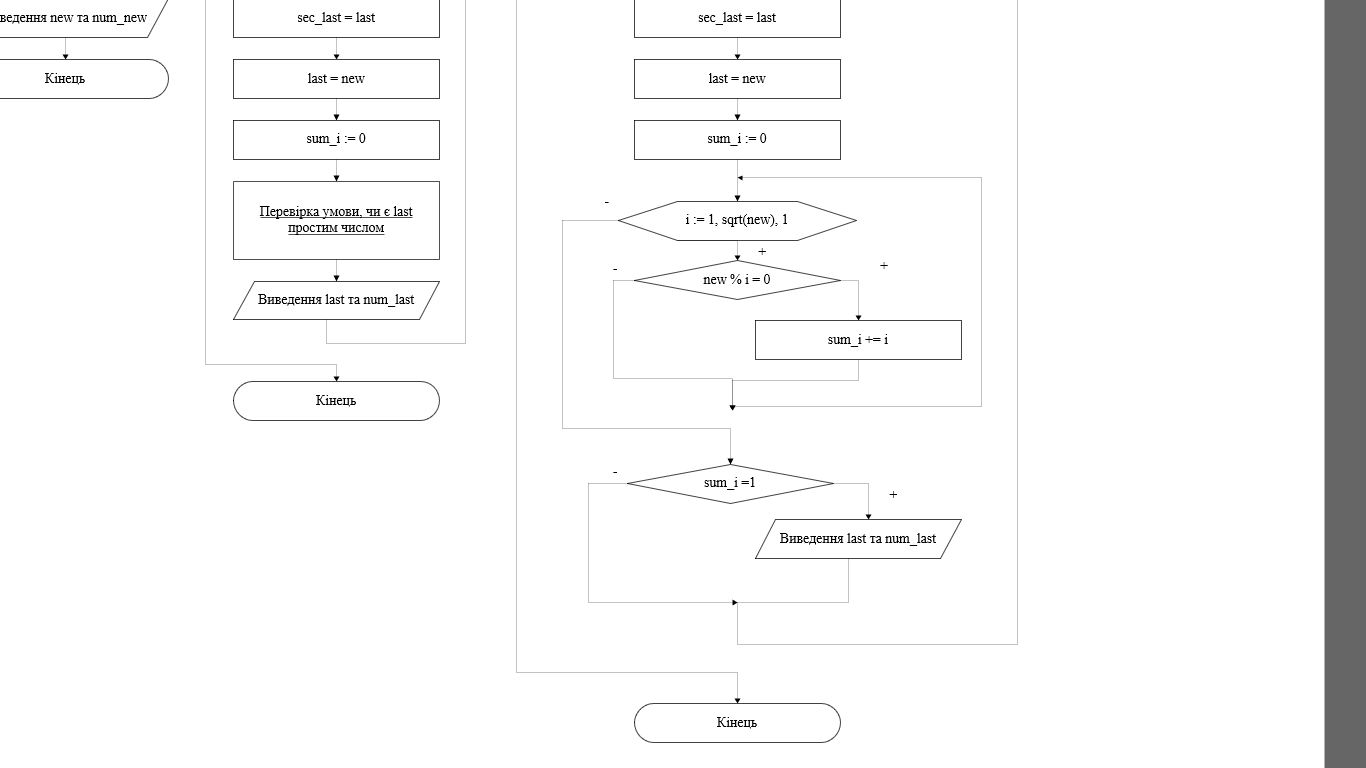
*Крок 2*



*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) | і = 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) | і = 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) | і = 1  5 % 1 = 0 істина  sum\_i = 1 |
| (внутрішній цикл 2) | і = 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) | і = 1  5 % 1 = 0 істина  sum\_i = 1 |
| (внутрішній цикл 2) | і = 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) | і = 1  13 % 1 = 0 істина  sum\_i = 1 |
| (внутрішній цикл 2) | і = 2  13 % 2 = 0 хиба |
| (внутрішній цикл 3) | і = 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) | і = 1  89 % 1 = 0 істина  sum\_i = 1 |
| (внутрішній цикл 2) | і = 2  89 % 2 = 0 хиба |
| … |  |
| (внутрішній цикл 9) | і = 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 і 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.