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 4

Виконав студент ІП-11 Воробйова Анастасія Денисівна

(шифр, прізвище, ім’я, по-батькові)

Перевірив

(прізвище, ім’я, по-батькові)

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.

**Завдання (варіант 4):** Визначити усі "щасливі" шестизначні числа. Число вважається "щасливим", якщо сума перших його цифр дорівнює сумі останніх трьох цифр.

**Постановка задачі**: Математичне формулювання задачі зводиться до перевірки кожного шестизначного числа на його відповідність умові, що виконується за допомогою зовнішнього циклу. Водночас у зовнішньому циклі будуть виконуватися два внутрішні цикли, перший з яких знаходить суму останніх трьох цифр числа, а другий – перших. Для знаходження суми будемо визначати кожну цифру, використовуючи виведену для цього формулу, що є комбінацією цілочисельного ділення та знаходження остачі від ділення. Останнім кроком перевіряємо, чи суми перших і останніх трьох членів рівні між собою, якщо так – виводимо число на екран.

**Математична модель**:

Складемо таблицю змінних:

|  |  |  |
| --- | --- | --- |
| Змінна | Тип | Призначення |
| i | Цілочисельний | Проміжні дані |
| lastsum | Цілочисельний | Проміжні дані |
| firstsum | Цілочисельний | Проміжні дані |
| k | Цілочисельний | Проміжні дані |
| a | Цілочисельний | Проміжні дані |
| n | Цілочисельний | Проміжні дані |
| b | Цілочисельний | Проміжні дані |

**Псевдокод:**

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо дію перебору усіх шестизначних чисел.

Крок 3. Деталізуємо дію ініціалізації змінних.

Крок 4. Деталізуємо дію знаходження суми останніх трьох цифр числа.

Крок 5. Деталізуємо дію знаходження суми перших трьох цифр числа.

Крок 6. Деталізуємо дію перевірки рівності сум.

**Крок 1:**

**Початок**

Перевірка всіх шестизначних чисел

**Кінець**

**Крок 2:**

**Початок**

**повторити**

**для** і **від** 100000 **до** 999999

ініціалізація lastsum, firstsum

знаходження lastsum

знаходження firstsum

перевірка рівності сум

**все повторити**

**Кінець**

**Крок 3:**

**Початок**

**повторити**

**для** і **від** 100000 **до** 999999

lastsum := 0

firstsum := 0

знаходження lastsum

знаходження firstsum

перевірка рівності сум

**все повторити**

**Кінець**

**Крок 4:**

**Початок**

**повторити**

**для** і **від** 100000 **до** 999999

lastsum := 0

firstsum := 0

**повторити**

**для** k **від** 0 **до** 2

a = (i/pow(10,k))%10

lastsum += a

**все повторити**

знаходження firstsum

перевірка рівності сум

**все повторити**

**Кінець**

**Крок 5:**

**Початок**

**повторити**

**для** і **від** 100000 **до** 999999

lastsum := 0

firstsum := 0

**повторити**

**для** k **від** 0 **до** 2

a = (i/pow(10,k))%10

lastsum += a

**все повторити**

**повторити**

**для** n **від** 3 **до** 5

b = (i/pow(10,n))%10

firstsum += b

**все повторити**

перевірка рівності сум

**все повторити**

**Кінець**

**Крок 6:**

**Початок**

**повторити**

**для** і **від** 100000 **до** 999999

lastsum := 0

firstsum := 0

**повторити**

**для** k **від** 0 **до** 2

a = (i/pow(10,k))%10

lastsum += a

**все повторити**

**повторити**

**для** n **від** 3 **до** 5

b = (i/pow(10,n))%10

firstsum += b

**все повторити**

**якщо** lastsum=firstsum

**то**

виведення i

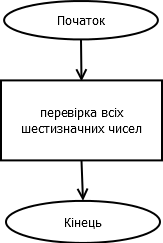
**все якщо**

**все повторити**

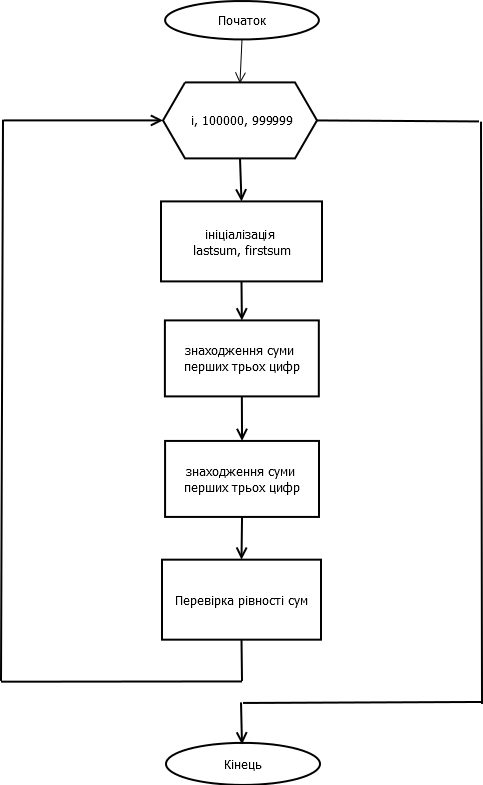
**Кінець**

**Блок-схема:**

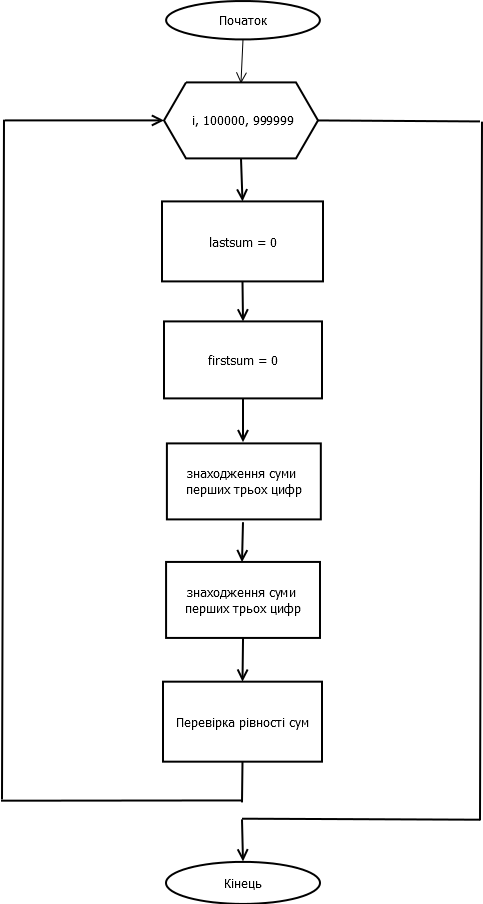
**Крок 1**



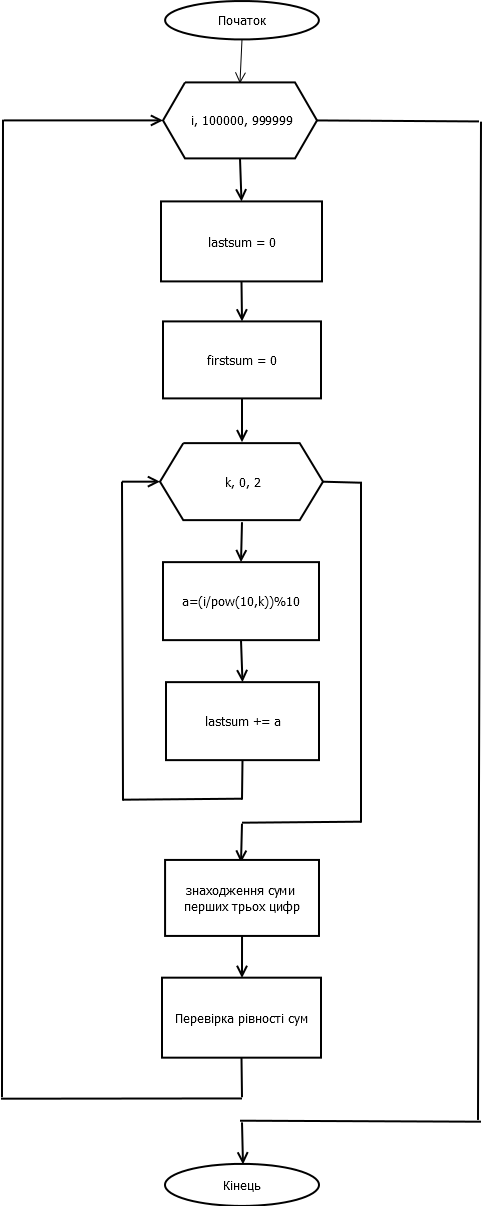
**Крок 2**



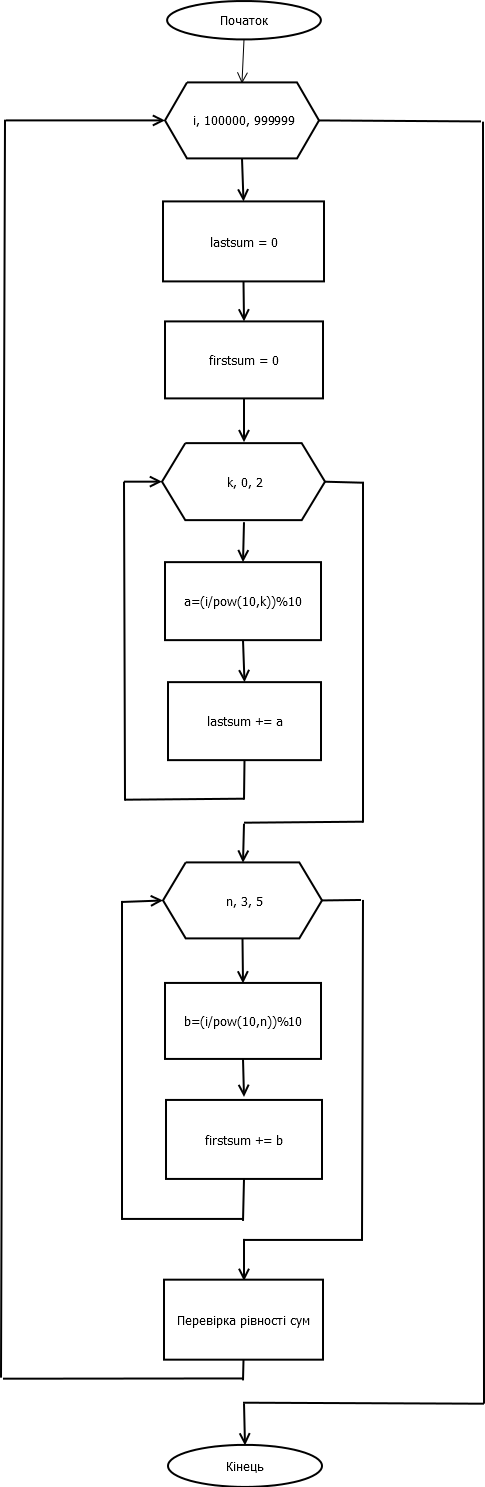
**Крок 3**



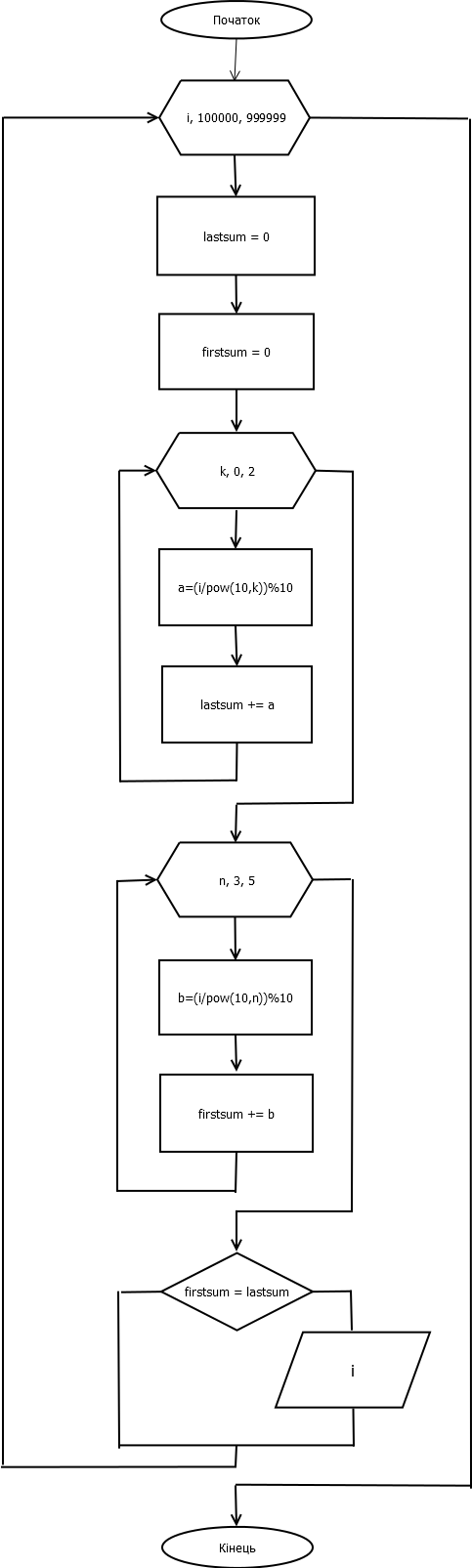
**Крок 4**



**Крок 5**



**Крок 6**



+

-

**Випробування алгоритму:**

|  |  |
| --- | --- |
| Блок | Дія |
|  | Початок |
| 1 | i = 100000 |
| 2 | lastsum = 0 |
| 3 | firstsum = 0 |
|  | Арифметичний цикл з параметром k |
| 4 | Перша ітерація: k=0, a = 0, lastsum = 0  Друга ітерація: k=1, a = 0, lastsum = 0  Третя ітерація: k=2, a = 0, lastsum = 0 |
|  | Арифметичний цикл з параметром n |
| 5 | Перша ітерація: n=3, b = 0, firstsum = 0  Друга ітерація: n=4, b = 0, firstsum = 0  Третя ітерація: n=5, b = 1, firstsum = 1 |
| 6 | firstsum ≠ lastsum (не виконується виведення i) |
|  | … |
| … | i = 463805 |
| … | lastsum = 0 |
| … | firstsum = 0 |
| … | Арифметичний цикл з параметром k |
| … | Перша ітерація: k=0, a = 5, lastsum = 5  Друга ітерація: k=1, a = 0, lastsum = 5  Третя ітерація: k=2, a = 8, lastsum = 13 |
| … | Арифметичний цикл з параметром n |
| … | Перша ітерація: n=3, b = 3, firstsum =3  Друга ітерація: n=4, b = 6, firstsum = 9  Третя ітерація: n=5, b = 4, firstsum = 13 |
| … | firstsum = lastsum |
| … | Виведенння i |
|  | … |
| … | Кінець |

**Висновок:**

У ході виконання даної лабораторної роботи нами були досліджені особливості роботи складних циклів. Для виконання поставленого завдання було створено математичну модель, написано псевдо-код та побудовано блок-схеми, що використовують зовнішній та вкладені арифметичні цикли та умовну форму оператора вибору. Для перевірки працездатносі алгоритму його було випробовувано і зроблено висновок, що алгоритм працює правильно і знаходить усі «щасливі» шестизначні числа, перевіряючи їх на відповідність умові.