

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 35

Виконав студент ІП-15, Шабанов Метін Шаміль огли
(шифр, прізвище, ім'я, по батькові)

Перевірів Вечерковська Анастасія Сергіївна
(прізвище, ім'я, по батькові)

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 35

Умова задачі

Цифровий корінь натурального числа – це одноцифрове значення, яке отримується із цифр числа шляхом ітераційного процесу знаходження спочатку суми цифр даного числа, а потім, якщо потрібно, суми цифр значень, отриманих на попередній ітерації знаходження відповідних сум (якщо значення суми не є цифрою). Цей процес триває до тих пір, поки не буде отримано однорозрядне число. Наприклад, цифровим коренем числа 65536 є 7, так як $6+5+5+3+6=25$ і $2+5=7$. Знайти всі числа з інтервалу (100, 200), які кратні своєму цифровому кореню.

Постановка задачі

Обчислити всі цифрові корені чисел із заданого інтервалу, а потім перевірити кратність цих чисел своїм цифровим кореням. Результатом виконання алгоритму є набір числових значень.

Побудова математичної моделі

Змінна	Тип	Ім'я	Призначення
Лічильник зовнішнього циклу	Ціле додатне	i	Проміжні дані
Лічильник першого внутрішнього циклу	Ціле додатне	j	Проміжні дані
Лічильник другого внутрішнього циклу	Ціле додатне	f	Проміжні дані
Лічильник третього внутрішнього циклу	Ціле додатне	k	Проміжні дані
Обмежувач зовнішнього циклу	Ціле додатне	lim1	Проміжні дані
Універсальний обмежувач для внутрішніх циклів	Ціле додатне	lim2	Проміжні дані

Заперечення рівності	Оператор	!=	Позначити логічне “не дорівнює”
Цілочисельне ділення	Оператор	/	Розділити число без залишку
Отримання залишку від ділення	Оператор	%	Отримати залишок від ділення
Значення цифрового кореню для першого випадку	Ціле	result	Вихідні дані
Значення цифрового кореню для другого випадку	Ціле	circle2	Вихідні дані
Значення цифрового кореню для третього випадку	Ціле	circle3	Вихідні дані

Для обчислення значення цифрового кореню використаємо зовнішній цикл, який буде перебирати числа від 100 до 200 включно; перший внутрішній цикл з умовою $j \neq 0$ та кроком $j / 10$, який буде обчислювати значення цифрового кореня за формулою $result = result + j \% 10$; два допоміжних цикли з аналогічними умовами, кроками і формулами, на випадок, якщо $result \geq 10$.

Розв’язання

Програмні специфікації запишемо у псевдокодi та графічній формi у вигляді блок-схеми.

Крок 1. Визначимо основні дії;

Крок 2. Деталізуємо присвоєння змінній result початкового значення.

Крок 3. Деталізуємо схему роботи зовнішнього циклу.

Крок 4. Деталізуємо схему роботи першого внутрішнього циклу.

Крок 5. Деталізуємо схему роботи допоміжних умовних операторів.

Крок 6. Деталізуємо присвоєння змінній circle2 початкового значення.

Крок 7. Деталізуємо схему роботи другого внутрішнього циклу.

Крок 8. Деталізуємо присвоєння змінній circle3 початкового значення.

Крок 9. Деталізуємо схему роботи третього внутрішнього циклу.

Крок 10. . Деталізуємо схему роботи умовного оператора перевірки кратності.

Псевдокод

Крок 1

Початок

Присвоєння змінній result початкового значення

Робота зовнішнього циклу

Робота першого внутрішнього циклу

Робота допоміжних умовних операторів

Присвоєння змінній circle2 початкового значення

Робота другого внутрішнього циклу

Присвоєння змінній circle3 початкового значення

Робота третього внутрішнього циклу

Робота умовного оператора перевірки кратності

Кінець

Крок 2

Початок

result = 0

Робота зовнішнього циклу

Робота першого внутрішнього циклу

Робота допоміжних умовних операторів

Присвоєння змінній circle2 початкового значення

Робота другого внутрішнього циклу

Присвоєння змінній circle3 початкового значення

Робота третього внутрішнього циклу

Робота умовного оператора перевірки кратності

result = 0

Кінець

Крок 3

Початок

result = 0

повторити

для i від 100 включно до 200 включно

Робота першого внутрішнього циклу

Робота допоміжних умовних операторів

Присвоєння змінній `circle2` початкового значення

Робота другого внутрішнього циклу

Присвоєння змінній `circle3` початкового значення

Робота третього внутрішнього циклу

Робота умовного оператора перевірки кратності

`result = 0`

все повторити

Кінець

Крок 4

Початок

`result = 0`

повторити

для i від 100 включно до 200 включно

$j = i$

повторити

`result = result + j % 10`

$j = j / 10$

поки $j = 0$

все повторити

Робота допоміжних умовних операторів

Присвоєння змінній `circle2` початкового значення

Робота другого внутрішнього циклу

Присвоєння змінній `circle3` початкового значення

Робота третього внутрішнього циклу

Робота умовного оператора перевірки кратності

`result = 0`

все повторити

Кінець

Крок 5

Початок

result = 0

повторити

для i від 100 включно до 200 включно

j = i

повторити

result = result + j % 10

j = j / 10

поки j = 0

все повторити

якщо result >= 10

Присвоєння змінній circle2 початкового значення

Робота другого внутрішнього циклу

якщо circle2 >= 10

Присвоєння змінній circle3 початкового значення

Робота третього внутрішнього циклу

Вивід circle3, i

все якщо

інакше

Вивід circle2, i

все якщо

інакше

Вивід result, i

все якщо

Робота умовного оператора перевірки кратності

result = 0

все повторити

Кінець

Крок 6

Початок

result = 0

повторити

для i від 100 включно до 200 включно

j = i

повторити

result = result + j % 10

j = j / 10

поки j = 0

все повторити

якщо result >= 10

circle2 = 0

Робота другого внутрішнього циклу

якщо circle2 >= 10

Присвоєння змінній circle3 початкового значення

Робота третього внутрішнього циклу

Вивід circle3. i

все якщо

інакше

Вивід circle2, i

все якщо

інакше

вивід result, i

все якщо

Робота умовного оператора перевірки кратності

result = 0

все повторити

Кінець

Крок 7

Початок

result = 0

повторити

для i від 100 включно до 200 включно

j = i

повторити

result = result + j % 10

j = j / 10

поки j = 0

все повторити

якщо result >= 10

circle2 = 0

f = result

повторити

circle2 = circle2 + f % 10

f = f / 10

поки f = 0

все повторити

якщо circle2 >= 10

Присвоєння змінній circle3 початкового значення

Робота третього внутрішнього циклу

Вивід circle3, i

все якщо

інакше

Вивід circle2, i

все якщо

інакше

вивід result, i

все якщо

Робота умовного оператора перевірки кратності

result = 0

все повторити

Кінець

Крок 8

Початок

result = 0

повторити

для i від 100 включно до 200 включно

j = i

повторити

result = result + j % 10

j = j / 10

поки j = 0

все повторити

якщо result >= 10

circle2 = 0

f = result

повторити

circle2 = circle2 + f % 10

f = f / 10

поки f = 0

все повторити

якщо circle2 >= 10

circle3 = 0

Робота третього внутрішнього циклу

Вивід circle3, i

все якщо

інакше

Вивід circle2, i

все якщо

інакше

вивід result, i

все якщо

Робота умовного оператора перевірки кратності

result = 0

все повторити

Кінець

Крок 9

Початок

result = 0

повторити

для i від 100 включно до 200 включно

j = i

повторити

result = result + j % 10

j = j / 10

поки j = 0

все повторити

якщо result >= 10

circle2 = 0

f = result

повторити

circle2 = circle2 + f % 10

f = f / 10

поки f = 0

якщо circle2 >= 10

circle3 = 0

k = circle2

повторити

circle3 = circle3 + k % 10

$k = k / 10$

поки $k = 0$

все повторити

Вивід circle3, i

все якщо

інакше

Вивід circle2, i

все якщо

все повторити

інакше

вивід result, i

все якщо

Робота умовного оператора перевірки кратності

result = 0

все повторити

Кінець

Крок 10

Початок

result = 0

повторити

для i від 100 включно до 200 включно

j = i

повторити

result = result + j % 10

j = j / 10

поки j = 0

все повторити

якщо result >= 10

circle2 = 0

f = result

повторити

circle2 = circle2 + f % 10

f = f / 10

поки f = 0

все повторити

якщо circle2 >= 10

circle3 = 0

k = circle2

повторити

circle3 = circle3 + k % 10

k = k / 10

поки k = 0

все повторити

якщо i % circle3 == 0

Вивід circle3, i

все якщо

все якщо

інакше

якщо i % circle2 == 0

Вивід circle2, i

все якщо

все якщо

інакше

якщо i % result == 0

Вивід result, i

все якщо

все якщо

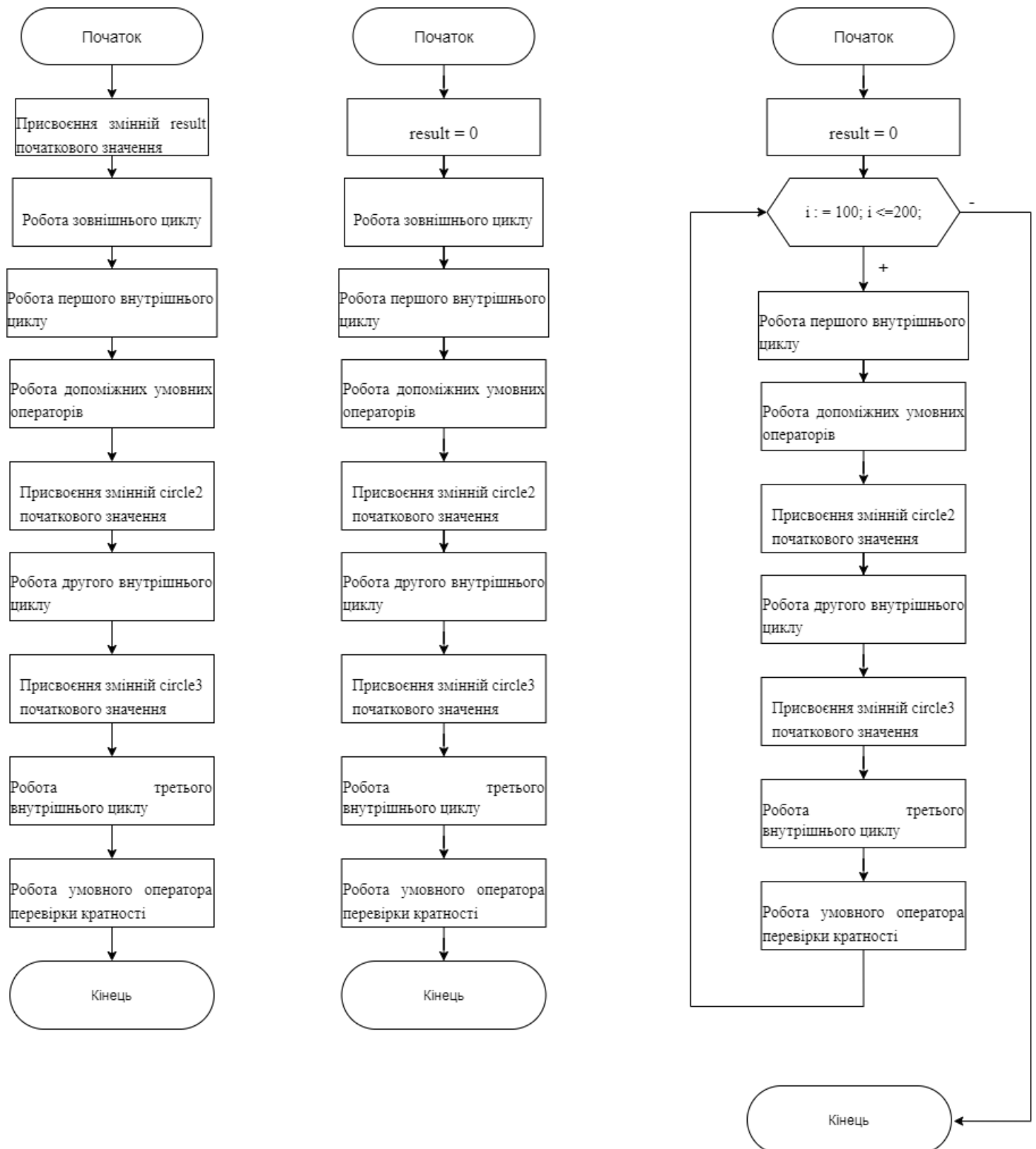
Робота умовного оператора перевірки кратності

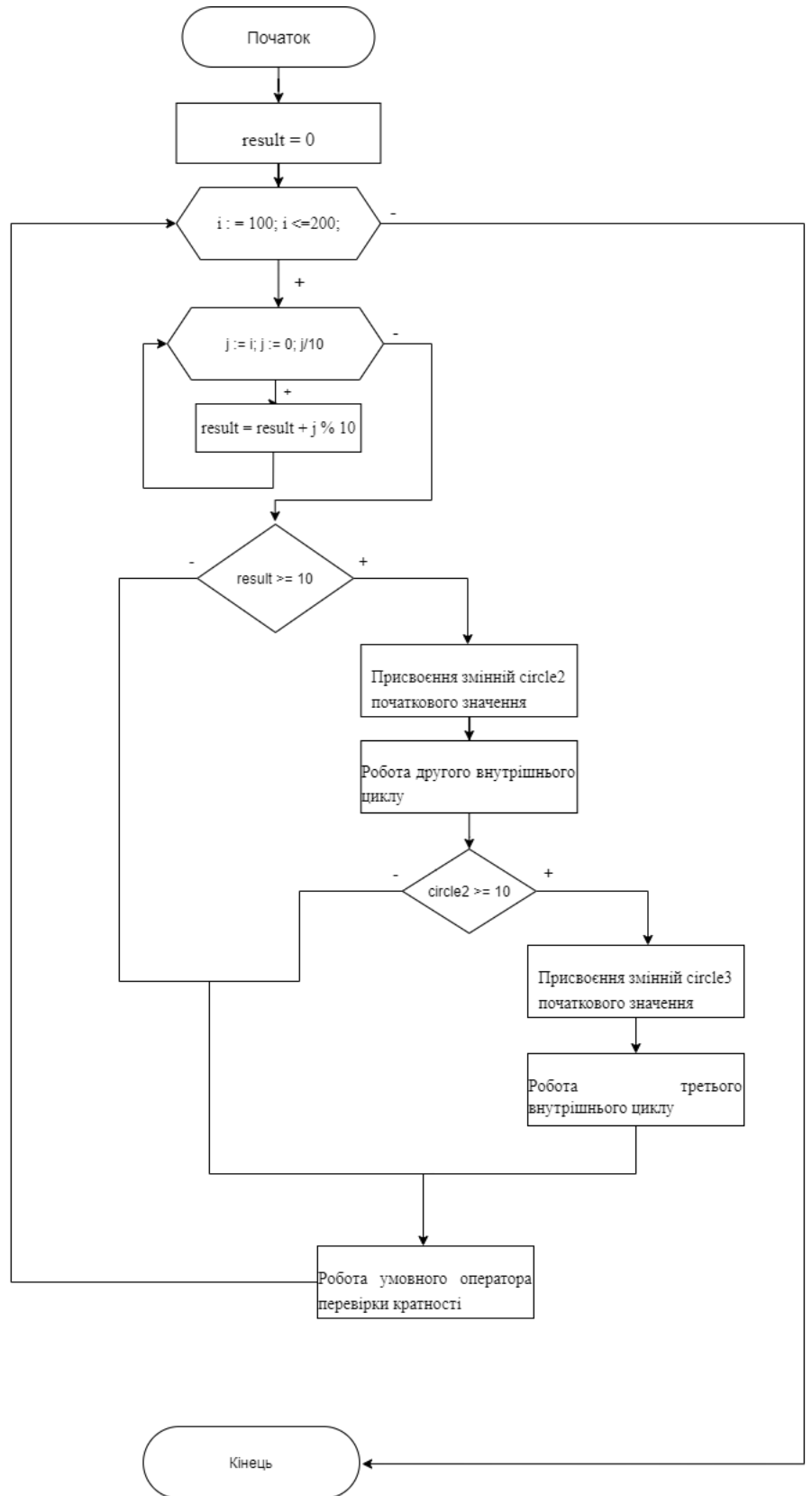
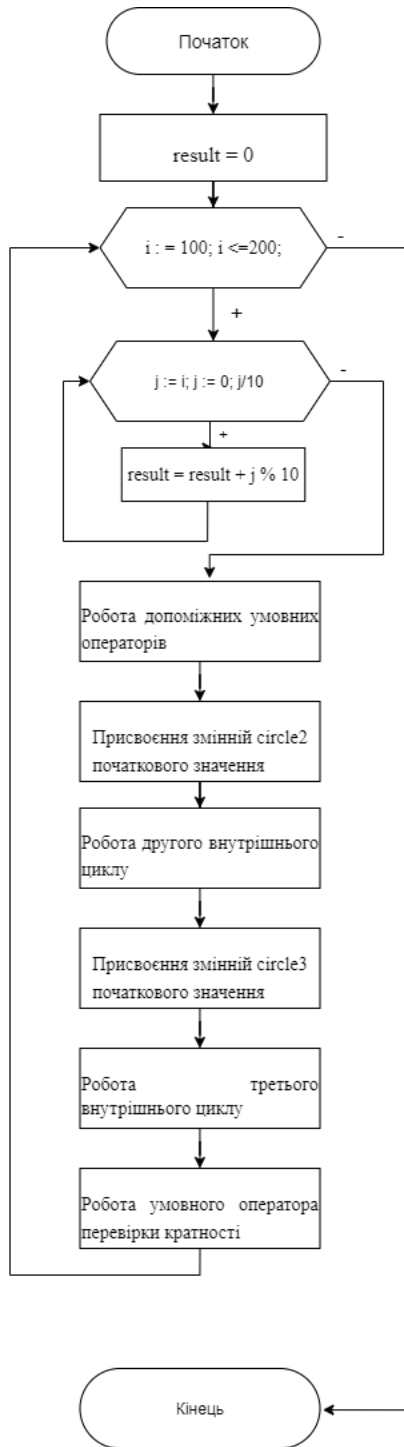
result = 0

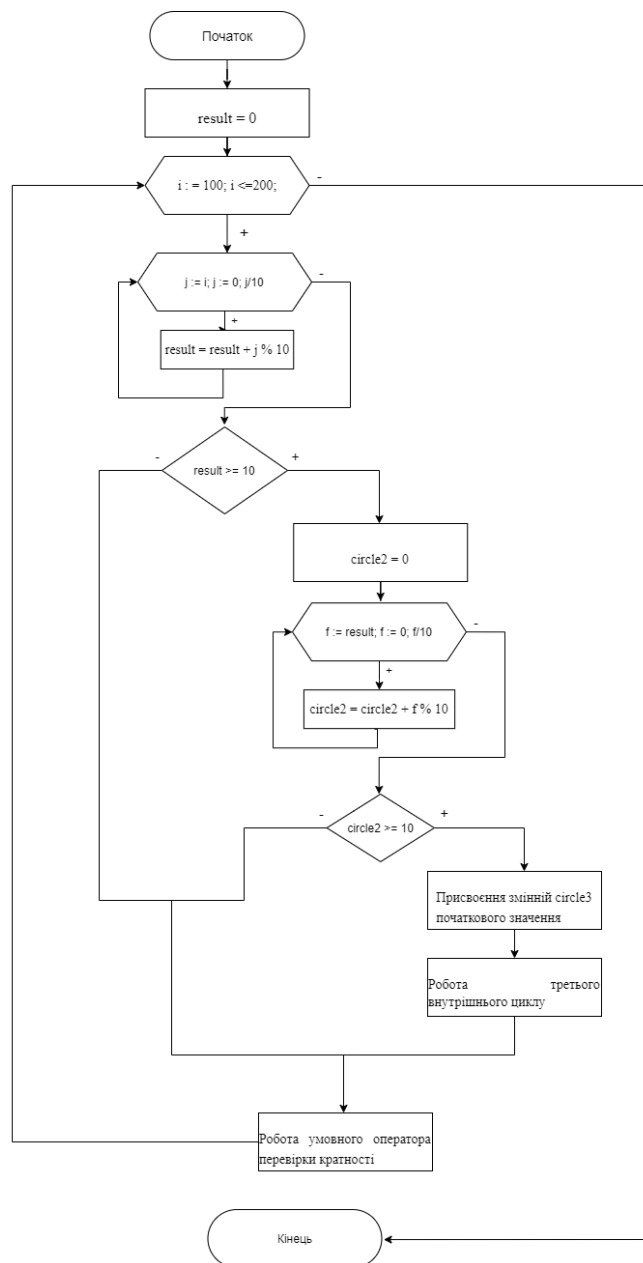
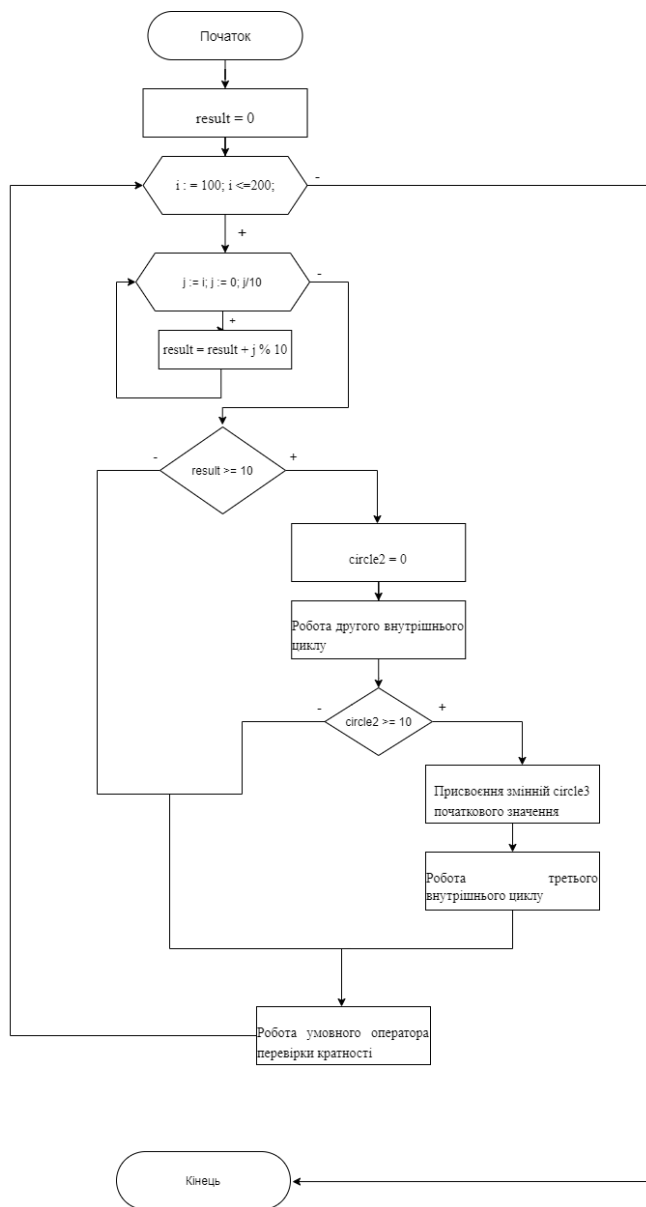
все повторити

Кінець

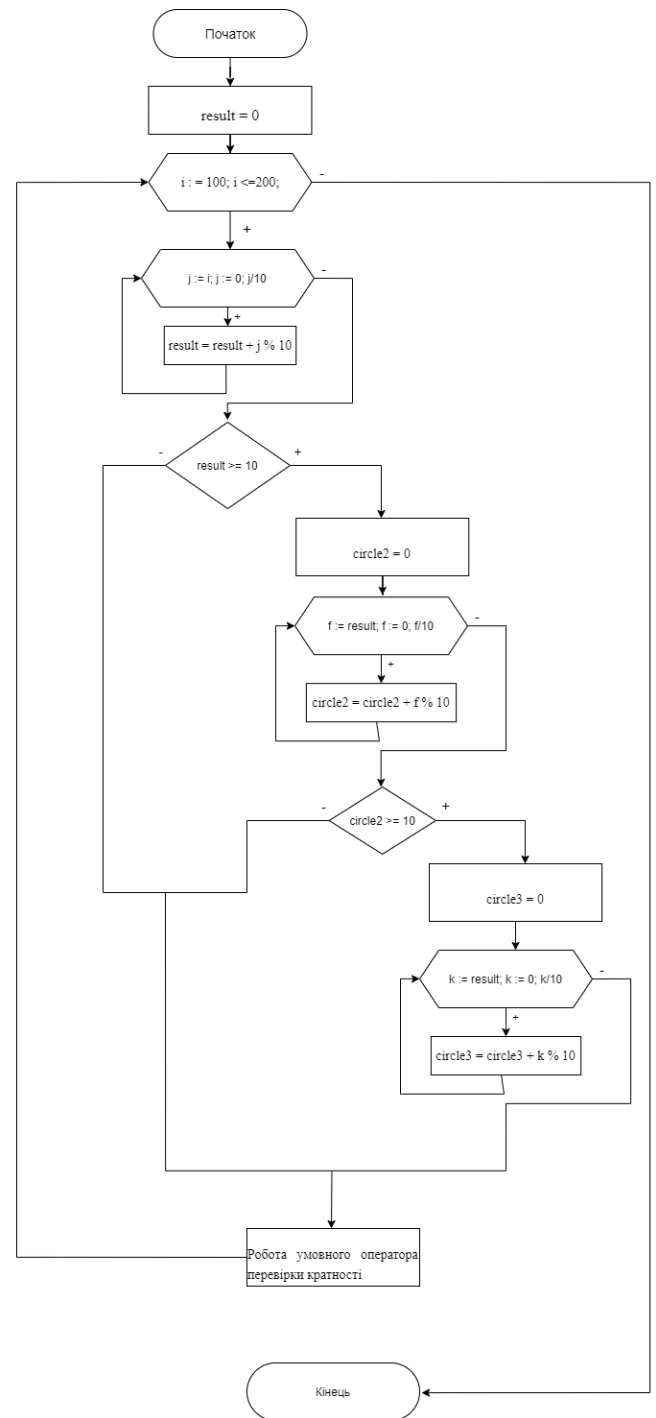
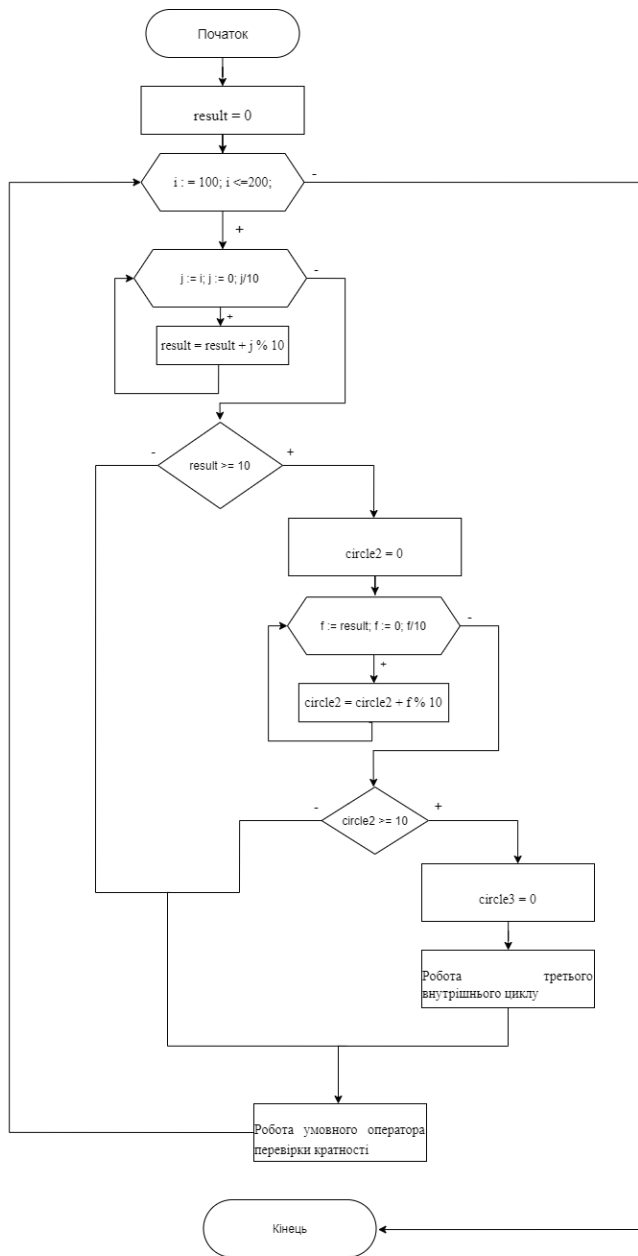
Блок схема

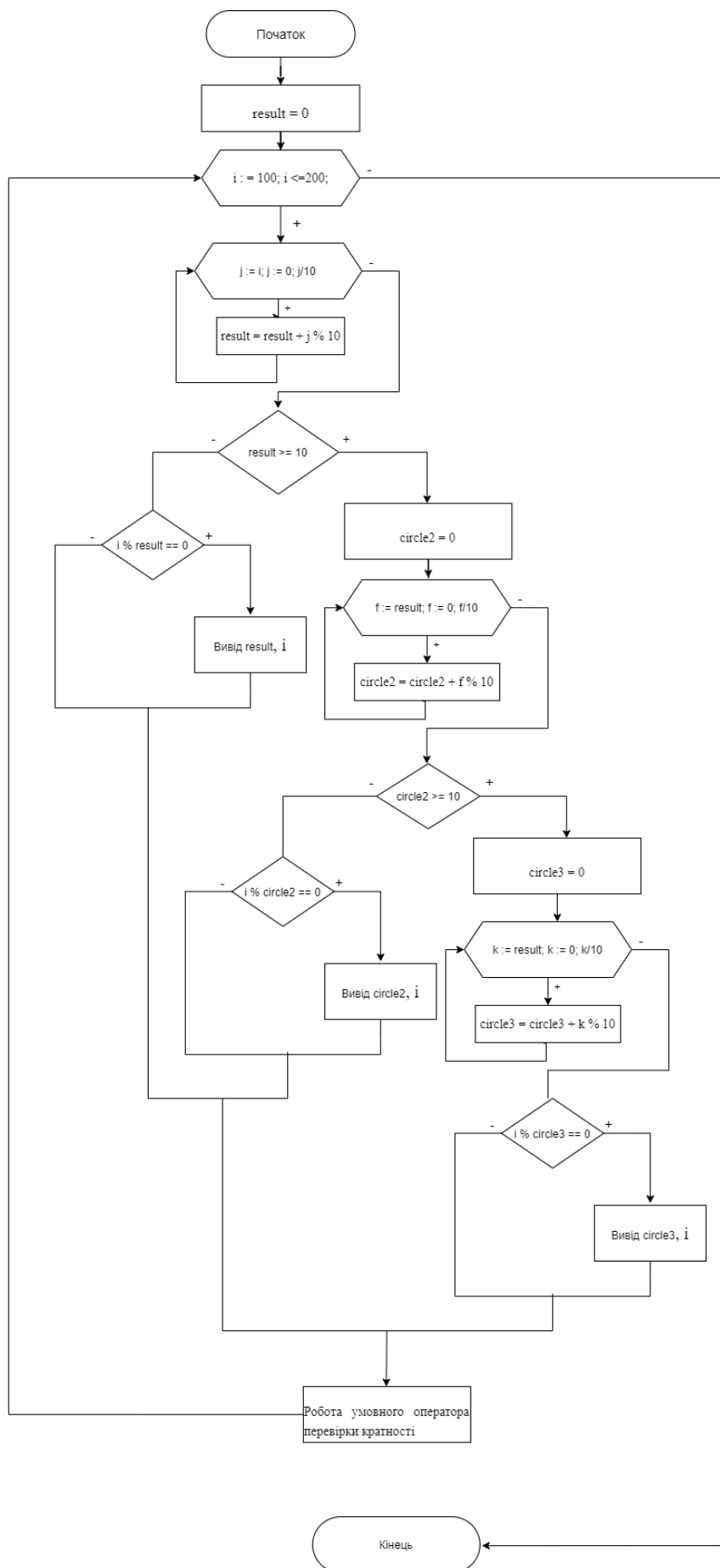






Основи програмування – 1. Алгоритми та структури даних





Випробування алгоритму

I

Блок	Дія
	Початок
1	i = 100; j = 100; result = 0;
2	i = 100; j = 10; result = 0;
3	i = 100; j = 1; result = 0;
4	i = 100; j = 0; result = 1; i % result = 0;
5	i = 101; j = 101; result = 0;
6	i = 101; j = 10; result = 1;
...	...
...	i = 199; j = 19; result = 9;
...	i = 199; j = 1; result = 18;
...	i = 199; j = 0; result = 19;
...	i = 199; f = 19; circle2 = 0;
...	i = 199; f = 1; circle2 = 9;
...	i = 199; f = 0; circle2 = 10;
...	i = 199; k = 10; circle3 = 0;
...	i = 199; k = 1; circle3 = 0;
...	i = 199; k = 0; circle3 = 1; i % circle3 = 0;
...	i = 200; j = 200; result = 0;
...	i = 200; j = 20; result = 0;
...	i = 200; j = 2; result = 0;
...	i = 200; j = 0; result = 2; i % result = 0;
...	Кінець

Висновки

Ми дослідили особливості роботи складних циклів та набули практичних навичок їх використання під час складання програмних специфікацій. Як результат, ми отримали алгоритм знаходження всіх чисел з інтервалу (100, 200), які кратні своєму цифровому кореню, розділивши задачу на десять кроків: визначення основних дій, деталізація присвоєння змінній `result` початкового значення, деталізація схеми роботи зовнішнього циклу, деталізація схеми роботи першого внутрішнього циклу, деталізація схеми роботи допоміжних умовних операторів, деталізація присвоєння змінній `circle2` початкового значення, деталізація схеми роботи другого внутрішнього циклу, деталізація присвоєння змінній `circle3` початкового значення, деталізація схеми роботи третього внутрішнього циклу, деталізація схеми роботи умовного оператора перевірки кратності. В процесі випробовування ми отримали результати 100; 102; 108; 109; 110; 111; 112; 114; 117; 118; 120; 126; 127; 128; 129; 132; 133; 135; 136; 138; 140; 144; 145; 146; 147; 148; 150; 152; 153; 154; 156; 162; 163; 164; 165; 168; 171; 172; 174; 180; 181; 182; 183; 184; 185; 186; 189; 190; 192; 196; 198; 199; 200.