

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 32

Виконав студент ІП-14, Шляхтун Денис Михайлович  
(шифр, прізвище, ім'я, по батькові)

Перевірів доц. кафедри ІІІ Мартинова Оксана Петрівна  
(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.

**Задача.** Отримати всі чотиризначні числа, в записі яких немає двох однакових цифр.

**Постановка задачі.** Результатом розв'язку є виведення усіх чотиризначних чисел, у яких немає однакових цифр. Для порівняння потрібно ввести змінну для кожної цифри, для наочного вигляду в умовному операторі буде використано окрему булеву змінну. Для перебору окремо тисяч з сотнями і десятками з одиницями буде використано два арифметичні цикли, для яких потрібні два лічильники. Ввідних даних не вимагається.

**Побудова математичної моделі.** Складемо таблицю імен змінних

<i>Змінна</i>	<i>Тип</i>	<i>Ім'я</i>	<i>Призначення</i>
Кількість тисяч	Цілий	d1	Проміжне дане
Кількість сотень	Цілий	d2	Проміжне дане
Кількість десятків	Цілий	d3	Проміжне дане
Кількість одиниць	Цілий	d4	Проміжне дане
Лічильник тисяч і сотень	Цілий	i	Проміжне дане
Лічильник десятків і одиниць	Цілий	k	Проміжне дане
Перевірка	Булевий	check	Проміжне дане
Число	Цілий	num	Результат

В алгоритмі використовуються наступні позначення:

$i / 10$  – цілочисельне ділення

$i \% 10$  – остача від ділення

$d1 \neq d2$  – число не дорівнює іншому

$\&\&$  – одночасність виконання умов

$i++$  – збільшення числа на 1

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

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо перебір тисяч та сотень.

Крок 3. Деталізуємо перебір сотень та одиниць.

Крок 4. Деталізуємо дію перевірки цифр на рівність.

Крок 5. Деталізуємо дію обрахунку та виведення результату.

### **Псевдокод.**

Крок 1.

**початок**

перебір тисяч та сотень

перебір десятків та одиниць

перевірка цифр на рівність

обрахунок результату

**кінець**

Крок 2.

**початок**

**повторити**

для  $i = 10; i < 100; i++$

$d1 = i / 10$

$d2 = i \% 10$

перебір десятків та одиниць

перевірка цифр на рівність

обрахунок результату

**все повторити**

**кінець**

Крок 3.

**початок**

**повторити**

**для  $i = 10; i < 100; i++$**

$d1 = i / 10$

$d2 = i \% 10$

**повторити**

**для  $k = 0; k < 100; k++$**

$d3 = k / 10$

$d4 = k \% 10$

перевірка цифр на рівність

обрахунок результату

**все повторити**

**все повторити**

**кінець**

Крок 4.

**початок**

**повторити**

**для  $i = 10; i < 100; i++$**

$d1 = i / 10$

$d2 = i \% 10$

**повторити**

**для  $k = 0; k < 100; k++$**

$d3 = k / 10$

$d4 = k \% 10$

$check = d1 != d2 \&\& d1 != d3 \&\& d1 != d4$

$check = check \&\& d2 != d3 \&\& d2 != d4 \&\& d3 != d4$

**якщо  $check = true$**

**то**

обрахунок результату

**все якщо**  
**все повторити**  
**все повторити**  
**кінець**

Крок 5.

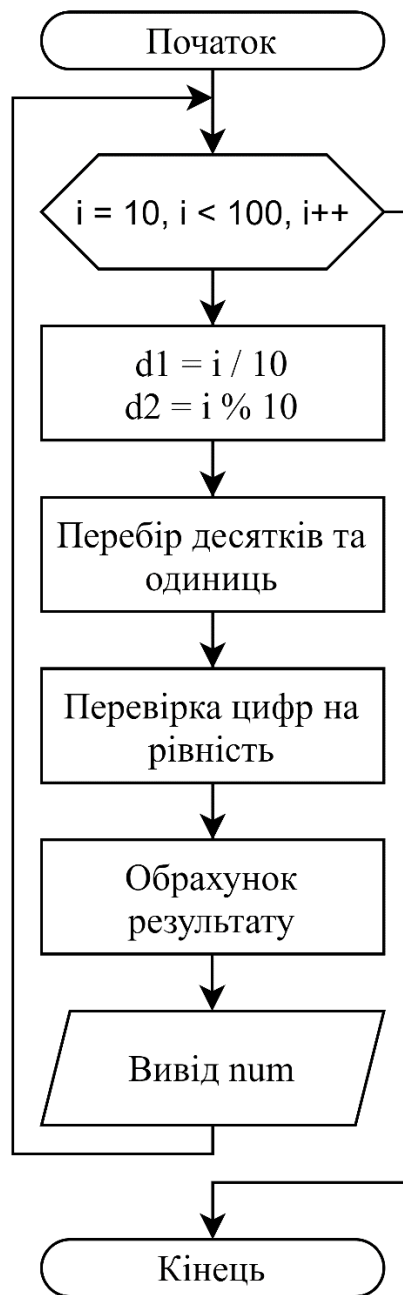
**початок**  
**повторити**  
  **для**  $i = 10; i < 100; i++$   
     $d1 = i / 10$   
     $d2 = i \% 10$   
    **повторити**  
      **для**  $k = 0; k < 100; k++$   
         $d3 = k / 10$   
         $d4 = k \% 10$   
         $check = d1 != d2 \&\& d1 != d3 \&\& d1 != d4$   
         $check = check \&\& d2 != d3 \&\& d2 != d4 \&\& d3 != d4$   
        **якщо**  $check = true$   
          **то**  
             $num = i * 100 + k$   
      **все якщо**  
    **все повторити**  
  **все повторити**  
**кінець**

**Блок-схема алгоритму**

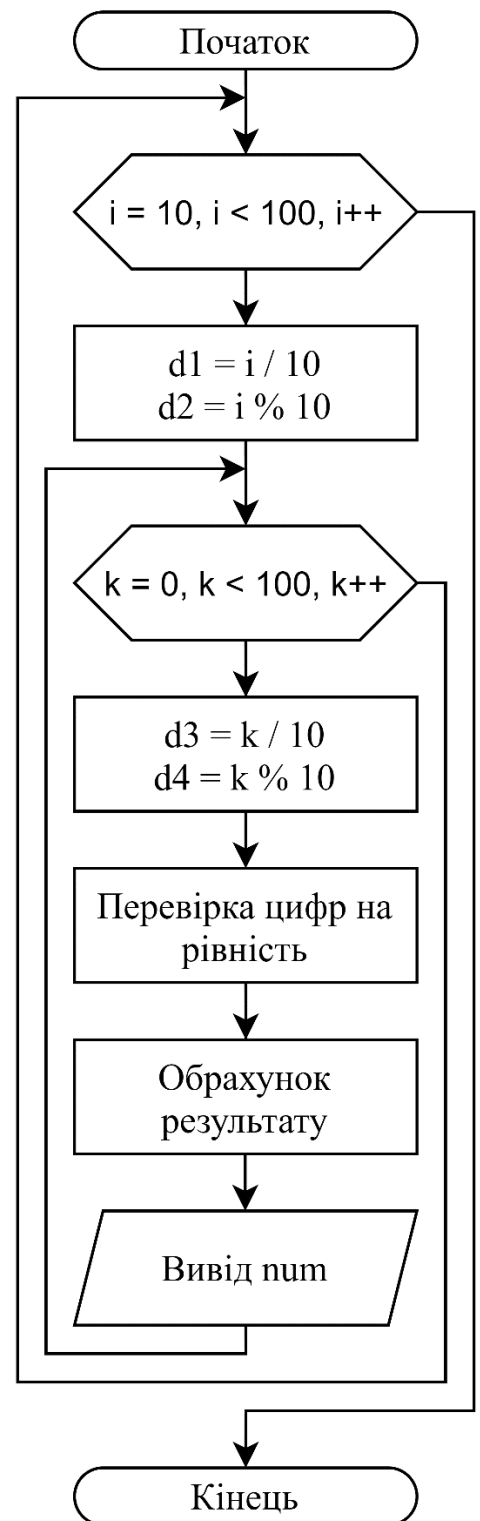
Крок 1



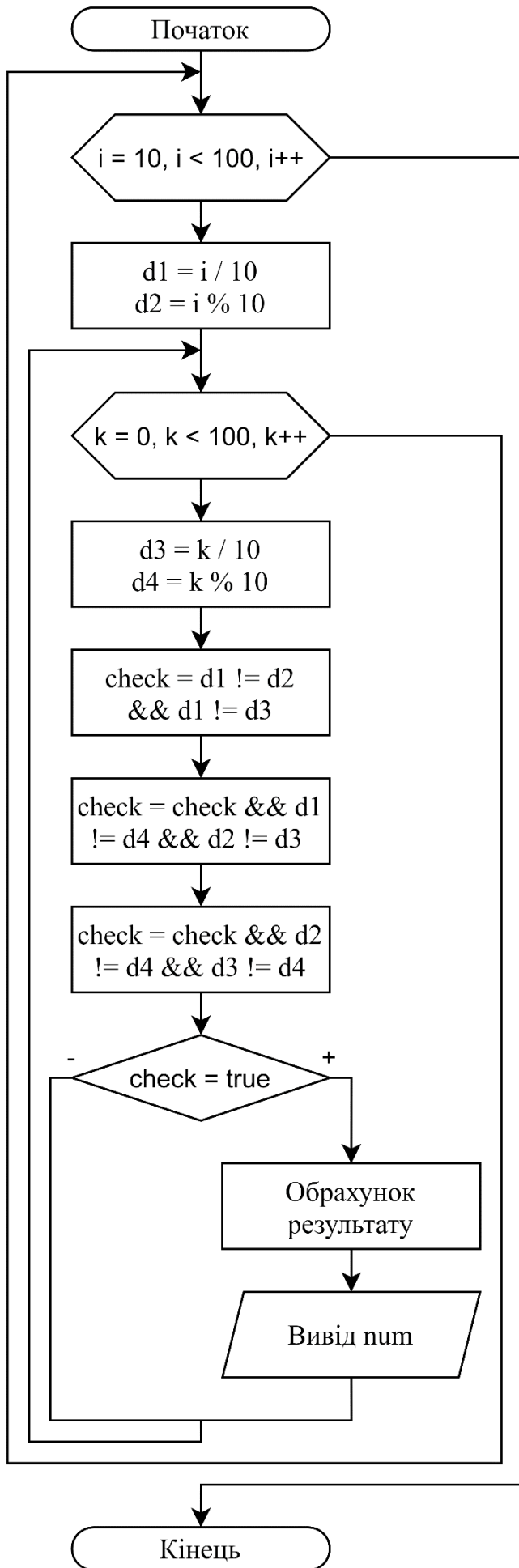
Крок 2



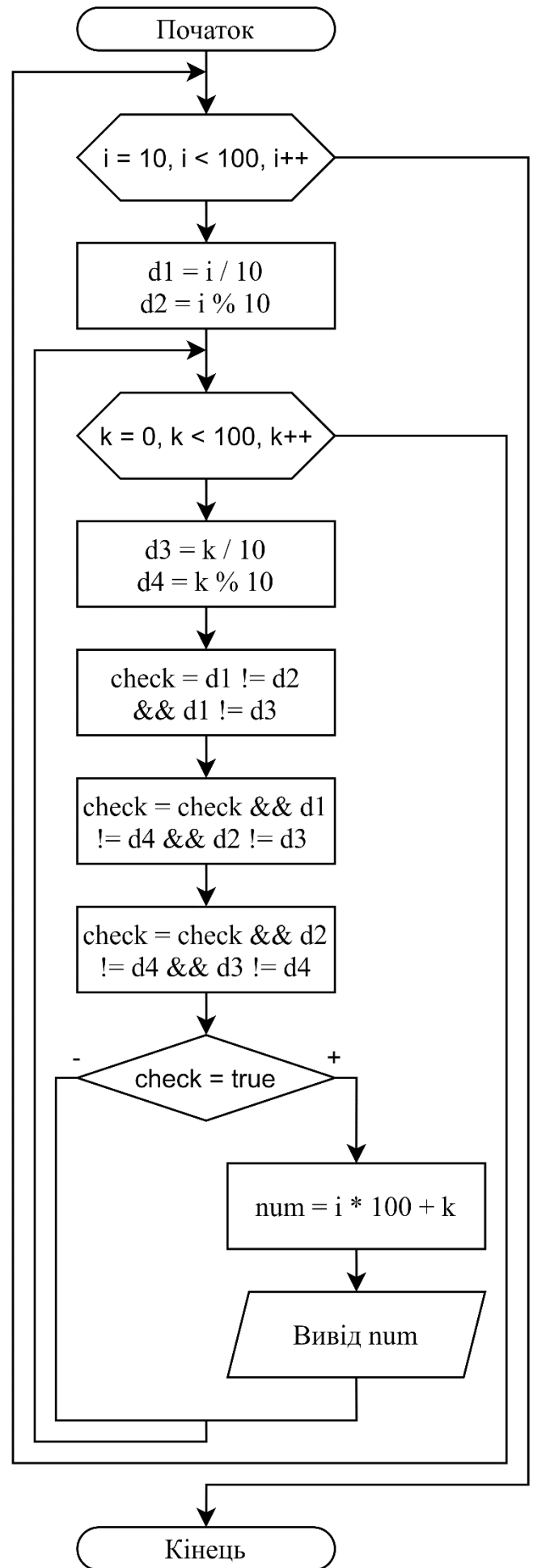
Крок 3



Крок 4



Крок 5



**Випробування алгоритму.**

<i>Ітерація</i>	<i>Дія</i>
	Початок
<b>1</b>	$i = 10$
	$d1 = 10/10=1; d2 = 10\%10=0$
1.1	$k = 0$
	$d3 = 0/10 = 0; d4 = 0\%10=0$
	$check = false$
...	...
1.24	$k = 23$
	$d3 = 23/10 = 2; d4 = 23\%10=3$
	$check = true$
	$num = 10 * 100 + 23 = 1023$
	Вивід: 1023
...	...
<b>89</b>	$i = 98$
	$d1 = 98/10=9; d2 = 98\%10=8$
...	...
89.77	$k = 76$
	$d3 = 76/10 = 7; d4 = 76\%10=6$
	$check = true$
	$num = 98 * 100 + 76 = 9876$
	Вивід: 9876
...	...
<b>90</b>	$i = 99$
	$d1 = 99/10=9; d2 = 99\%10=9$
...	...
90.100	$k = 99$
	$d3 = 99/10 = 9; d4 = 99\%10=9$



	check = false
	Кінець

### **Висновок.**

При виконанні лабораторної роботи було використано структуру з вкладеним циклом. У середині одного арифметичного циклу знаходиться інший арифметичний цикл. Особливістю такої конструкції є проходження всіх ітерацій внутрішнього циклу під час однієї ітерації зовнішнього. Ця структура була використана для перебору всіх десятків та одиниць для кожних можливих сотень та тисяч. Дана задача може бути виконана за допомогою чотирьох вкладених циклів, але алгоритм буде занадто важким для читання та розуміння, тому було прийнято рішення перебору окремо тисяч з сотнями та десятків з одиницями. Також в алгоритмі було використано оператор розгалуження умовного типу, а для зручності перевірка умови виконується за допомогою булевої змінної.