Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 3 з дисципліни «Алгоритми та структури даних-1. Основи алгоритмізації»

«Дослідження лінійних алгоритмів»

Варіант 24

Виконав студент ІП-12 Орищенко Ярослав Олександрович

(шифр, прізвище, ім'я, по батькові)

Перевірив \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Лабораторна робота 6

Дослідження складних циклічних алгоритмів

Мета: дослідити особливості роботи рекурсивних алгоритмів та набути практичних навичок їх використання під час складання програмних специфікацій підпрограм.

**Розв’язок:**

1. ***Постановка задачі***. Потрібно знайти перші 10 п’ятизначних паліднромів, які є простими числами. Для цього ми будемо використовувати дві булеві функції, перша з яких буде перевевіряти число на паліндром шляхом знаходженння оберненого числа до даного (123 – 321), друга буде перевіряти число на простоту, та третю функцію, яка, за умови що обидві попередні функції повертають значення True, буде виводити задану кількість простих п’ятизначних паліндромів.
2. ***Математична модель*.** Складемо таблицю імен змінних.

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| Потрібна кількість паліндромів | Цілий | n | Проміжне дане |

1. ***Псевдокод***

*Крок 1 Крок 2*

**початок** в введення n d primePalyndroms (n) **кінець**

**початок**  в n = 10 d primePalyndroms(n) **кінець**

***Псевдокод підпрограм***

**palyndrome**(n) n n1 = n

reversed = 0

**поки** n1 != 0

reversed \*= 10

reversed += n1 % 10

n1 /= 10

**все поки**

**якщо** reversed == n, то

return 1

**інакше**

return 0

**все якщо**

**кінець** **palyndrome**

**prime**(n) n d divider = 2

**поки** n % divider != 0

divider++

**все поки**

**якщо** divider == n, то

return 1

**інакше**

return 0

**все якщо**

**кінець** **prime**

**primePalyndromes**(n) s i = 0;

num = 10000;

**повторити** для num від 10000 до 100000

**якщо** і < n

**якщо** palyndrome == true && prime == true

виведення num

і++

**все якщо**

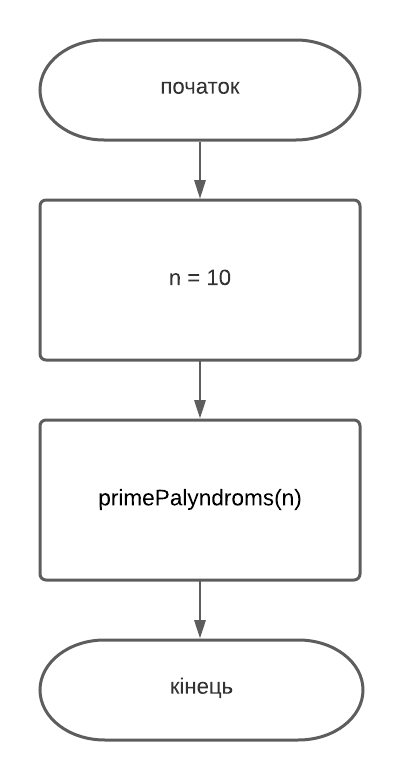
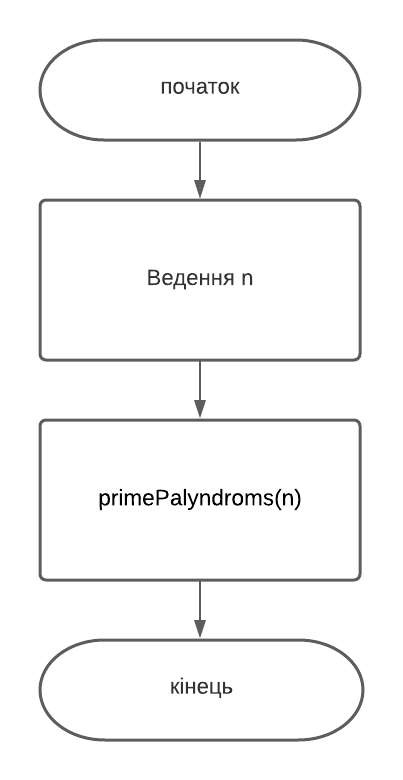
**все якщо**

**все повторити**

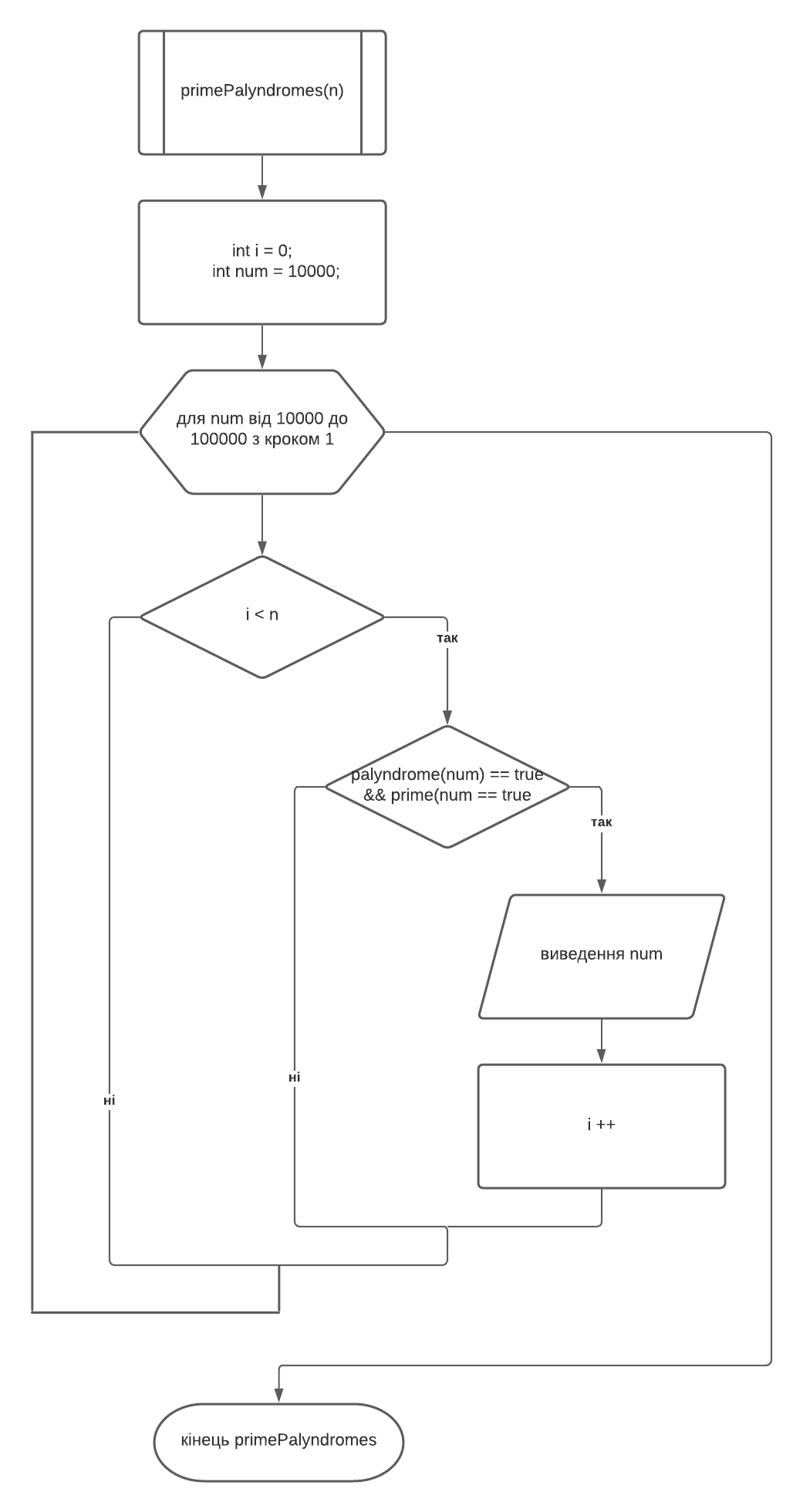
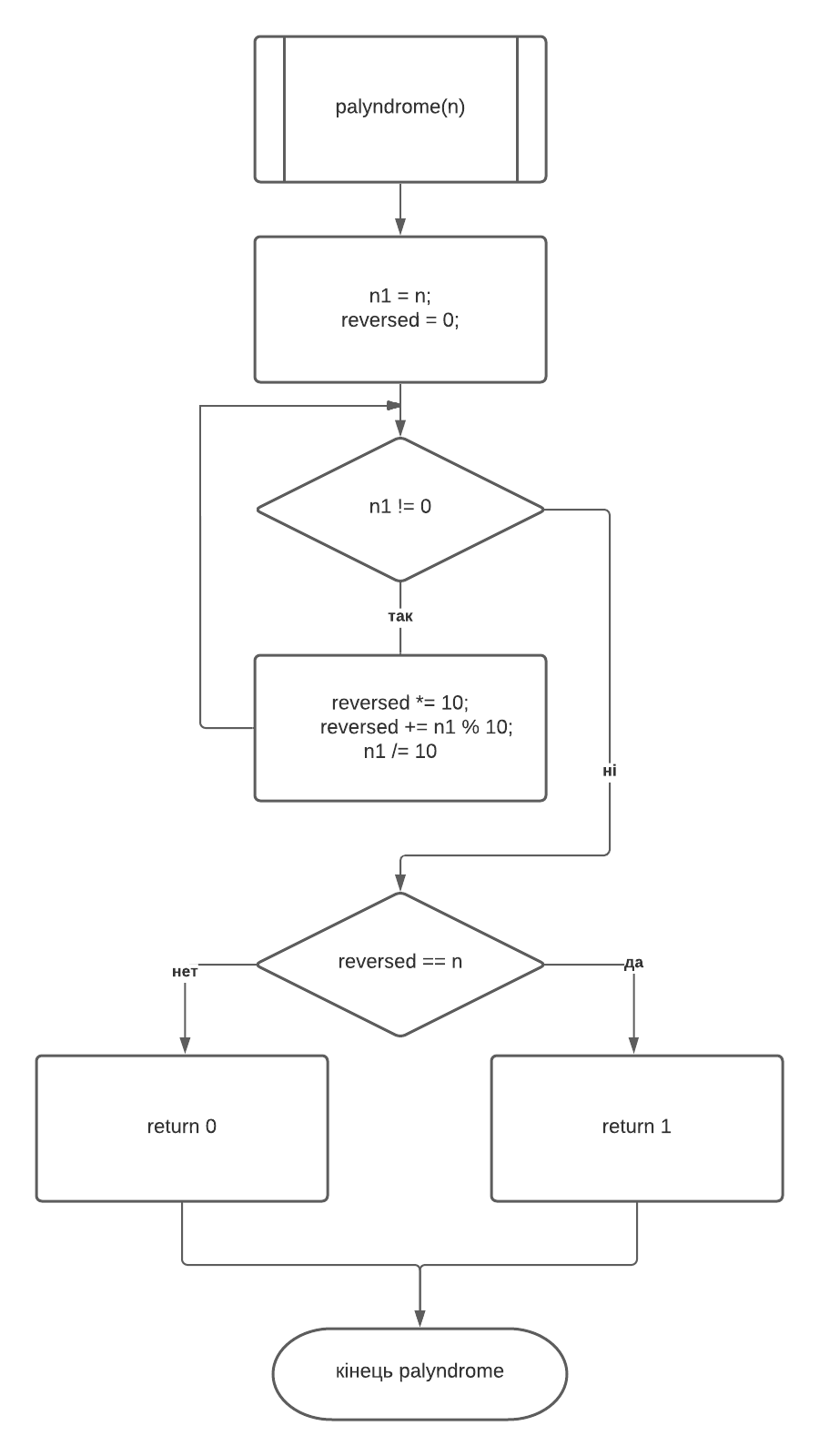
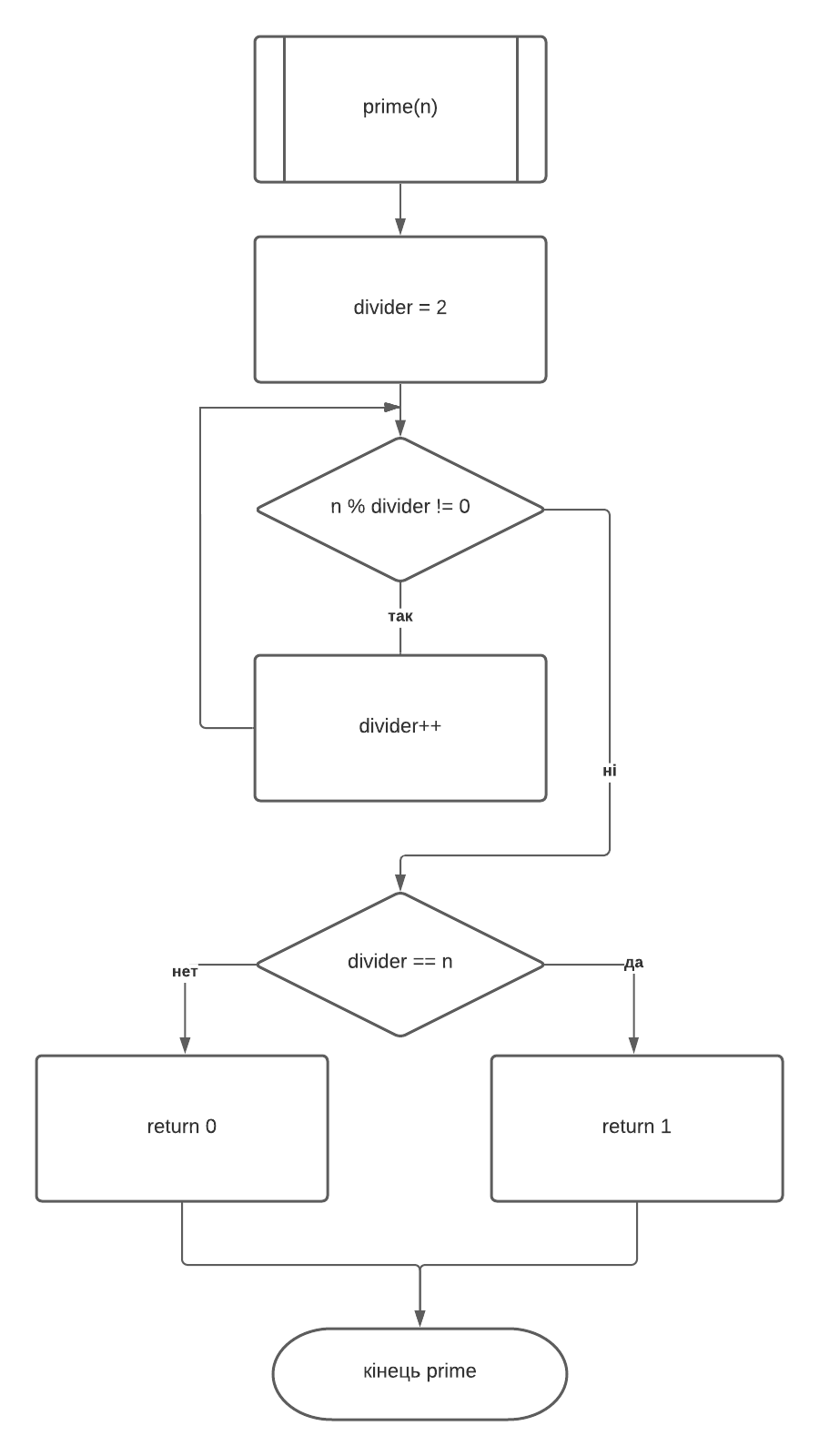
**кінець** **primePalyndromes**

1. ***Блок-схема***

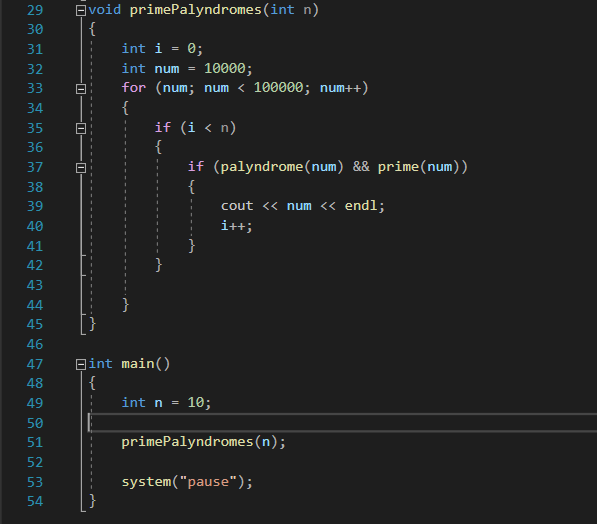
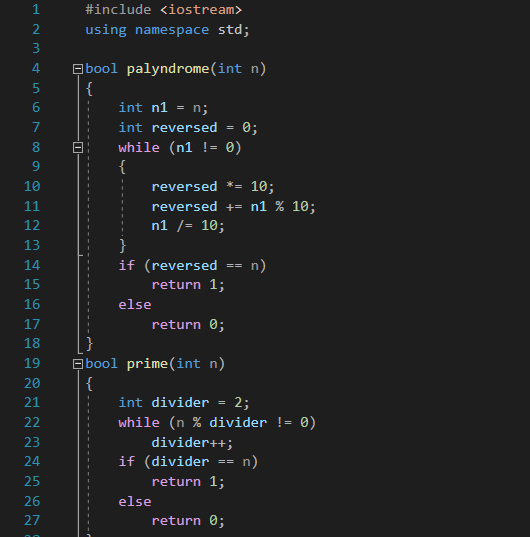
*Крок 1 Крок 2*



*Блок схеми функцій*



1. ***Код програми***



1. ***Перевірка алгоритму***

|  |  |
| --- | --- |
| Блок | Дія |
|  | Початок |
| 1 | Введення і = 1 |
| 2 | i =1; num = 10000; palyndrome == 0; prime == 0 |
| 3 | i =1; num = 10001; palyndrome == 1; prime == 0 |
| 4 | i =1; num = 10002; palyndrome == 0; prime == 0 |
| 5 | ... |
| 6 | i = 1; num = 10201; palyndrome == 1; prime == 0 |
| 7 | ... |
| 8 | i =1; num = 10301; palyndrome == 1; prime == 1; Виведення 10301 |
| 9 | i = 2; num = 10302; palyndrome == 0; prime == 0 |
| 10 | ... |
| 11 | i =2; num = 10501; palyndrome == 1; prime == 1; Виведення 10501 |
| 12 | ... |
| 13 | i =3; num = 10601; palyndrome == 1; prime == 1; Виведення 10601 |
| 14 | ... |
| 15 | i =4; num = 11311; palyndrome == 1; prime == 1; Виведення 11311 |
| 16 | ... |
| 17 | i =5; num = 11411; palyndrome == 1; prime == 1; Виведення 11411 |
| 18 | ... |
| 19 | i =6; num = 12421; palyndrome == 1; prime == 1; Виведення 12421 |
| 20 | ... |
| 21 | i =7; num = 12721; palyndrome == 1; prime == 1; Виведення 12721 |
| 22 | ... |
| 23 | i =8; num = 12821; palyndrome == 1; prime == 1; Виведення 12821 |
| 24 | ... |
| 25 | i =9; num = 13331; palyndrome == 1; prime == 1; Виведення 13331 |
| 26 | ... |
| 27 | i =10; num = 13831; palyndrome == 1; prime == 1; Виведення 13831 |
| 28 | i = 11 |
|  | Кінець |

***6.Висновки.*** Було досліджено особливості роботи рекурсивних алгоритмів та набуто практичних навичок їх використання під час складання програмних специфікацій підпрограм на прикладі задачі зі знаходженням простих алгоритмів, для вирішення якої було створено різноманітні