# 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 №6 з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«Дослідження лінійних алгоритмів»

Варіант 7

Виконав студент ІП-12 Васильєв Єгор Костянтинович

(шифр, прізвище, ім'я, по батькові)

Перевірив

( прізвище, ім'я, по батькові)

Київ 2021

**Лабораторна робота №6**

**Дослідження лінійних алгоритмів**

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

**Індивідуальне завдання:**

**Постановка задачі**

Результатом розв’язку є переведене у вісімкову систему числення задане користувачем додатне ціле десяткове число.

**Побудова математичної моделі**

Щоб переводити число у вісімкову систему числення будемо використовувати рекурсивний виклик допоміжного алгоритму, який щоразу цілочисельним діленням на 8 зменшує задане число та виводить його остачу від ділення на 8, доки число більше 8. Також використаємо оператор альтернативної форми вибору щоб переконатися, що введене користувачем число додатне й оператор умовної форми вибору всередині функції.

*Складемо таблицю змінних*

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| Значення числа | Цілий | n | Переводиться в іншу СЧ |
| Допоміжний алгоритм | Порожній | oct | Переводить число в іншу СЧ |

**Розв’язання**

Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді блок-схеми.

*Крок 1.* Деталізуємо основні дії

*Крок 2.* Деталізуємо допоміжний алгоритм переведення числа у вісімкому СЧ

*Крок 3.* Деталізуємо перевірку введеного числа

*Крок 4.* Деталізуємо виклик допоміжного алгоритму

**Псевдокод**

Крок 1

**-**деталізація допоміжного алгоритму

**початок**

**-**перевірка введеного числа

**-**виклик допоміжного алгоритму

**кінець**

Крок 2

**-oct(n)**

**якщо** n>=8

**то**

**oct(n/8)**

**все якщо**

вивести n%8

**кінець oct(n)**

**початок**

**-**перевірка введеного числа

**-**виклик допоміжного алгоритму

**кінець**

Крок 3

**-oct(n)**

**якщо** n>=8

**то**

**oct(n/8)**

**все якщо**

вивести n%8

**кінець oct(n)**

**початок**

**якщо** n>=0

**то**

виклик допоміжного алгоритму

**інакше**

вивести «Incorrect number»

**все якщо**

**кінець**

Крок 4

**-oct(n)**

**якщо** n>=8

**то**

**oct(n/8)**

**все якщо**

вивести n%8

**кінець oct(n)**

**початок**

**якщо** n>=0

**то**

вивести **oct(n)**

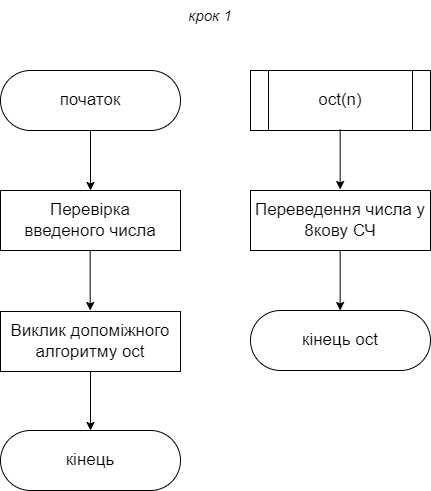
**інакше**

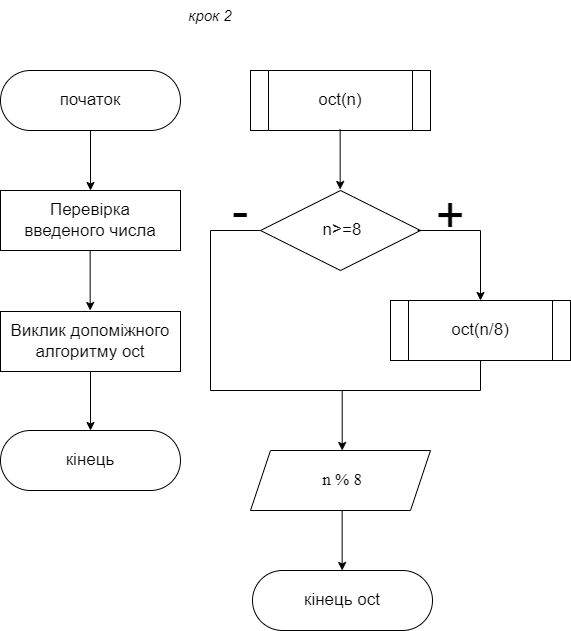
вивести «Incorrect number»

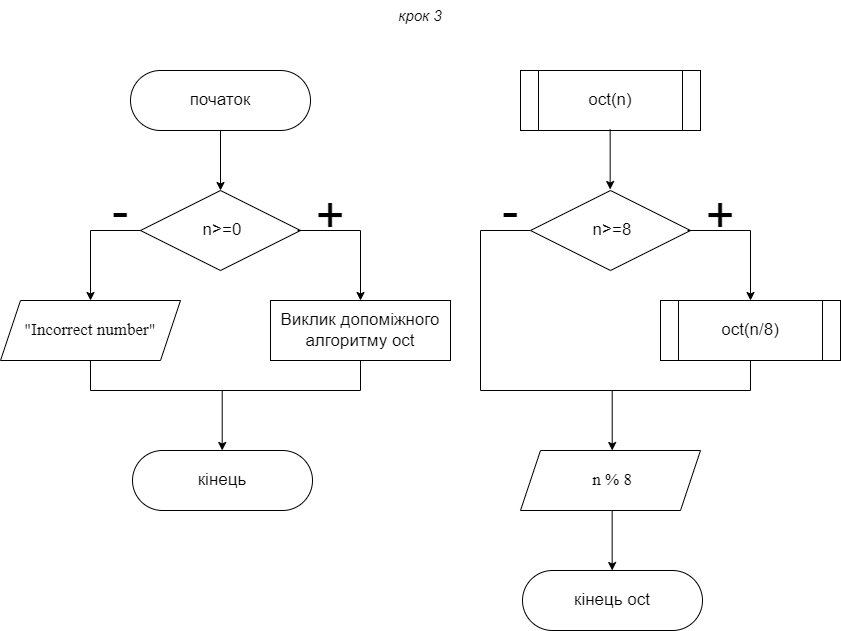
**все якщо**

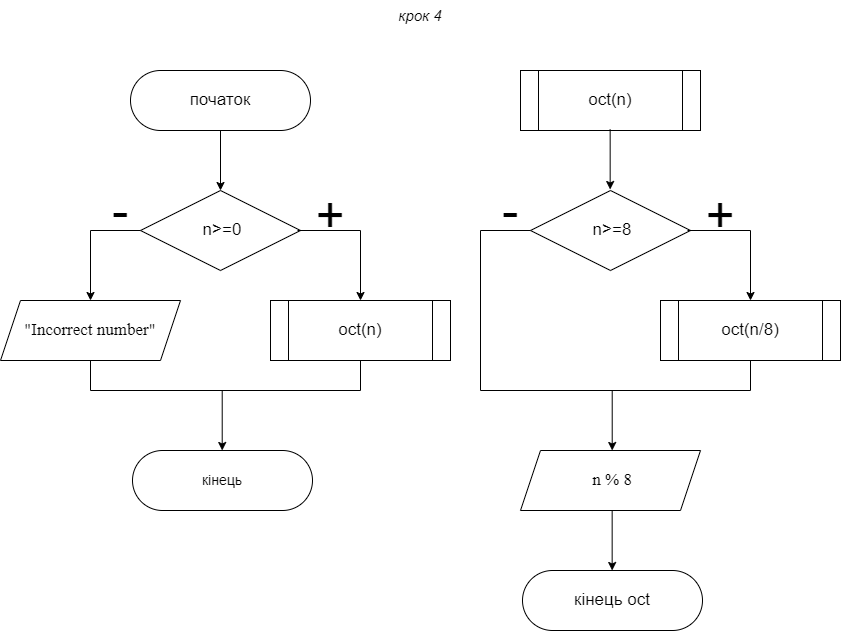
**кінець**

**Блок-схема**





****

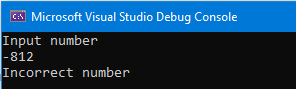
****

**Тестування**

***к****рок 1*

**початок**

-введення n=-812

-перевірка введеного числа

-виклик допоміжного алгоритму

**кінець**

***к****рок 2*

**початок**

-введення n=-812

-«Incorrect number»

**кінець**

***к****рок 1*

**початок**

-введення n=3071

-перевірка введеного числа

-виклик допоміжного алгоритму

**кінець**

***к****рок 2*

**початок**

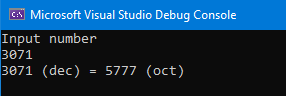
-введення n=3071

-n задовольняє умову

-виклик допоміжного алгоритму

**кінець**

***к****рок 3*

**початок**

-введення n=3071

n>=0

oct(3071)

3071>=8

oct(383)

383>=8

oct(47)

47>=8

oct(5)

5<8

вивести 5 (5%8)

вивести 7 (47%8)

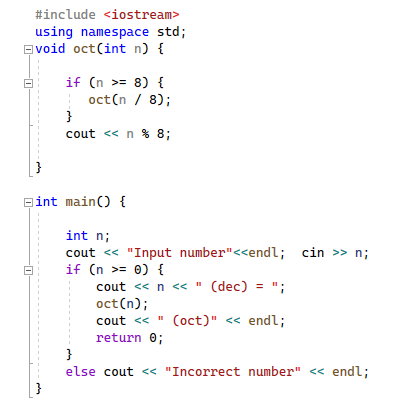
вивести 7 (383%8)

вивести 7 (3071%8)

-3071 (dec) = 5777 (oct)

**кінець**

**Код програми**



**Висновок**

Було досліджено особливості роботи рекурсивних алгоритмів та набуто практичних навичок їх використання під час складання програмних специфікацій підпрограм й розроблено псевдокод і блок-схему до відповідного завдання.