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 No 6 з дисципліни
«Алгоритми та структури даних-1.
Основи алгоритмізації»

«Дослідження рекурсивних алгоритмів»

Варіант 7

Виконав студент ІП-1407 Грицина Діана Русланівна (шифр, прізвище, ім'я, по батькові)

Перевірив Мартинова Оксана Петрівна (прізвище, ім'я, по батькові)

Київ 2021

Лабораторна робота 6

Дослідження рекурсивних алгоритмів

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

Задача:

7. Перетворення додатного цілого десяткового значення в значення у вісімковій системі числення

Розв’язання

Постановка задачі

Перетворення числа у вісімкову систему числення виконаємо послідовним діленням числа на 8. Остача від ділення записується в результат в відповідний їй розряд, який визначається за порядковим номером у послідовності.

Побудова математичної моделі

Змінна	Тип	Призначення
i	Цілий	Розряд числа
oct_num	Цілий	Результат виклику функції
number	Цілий	Проміжне значення
Функція octal	Цілий	Перетворення числа у 8 систему числення
user_number	Цілий	Початкове дане
octal_user_number	Цілий	Результат

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

Крок 1. Визначимо основні дії
Крок 2. Встановлення умови для виконання рекурсивної гілки у функції
Крок 3. Перетворення числа у вісімкову систему числення

Псевдокод

Крок 1
початок
Перетворення числа у вісімкову систему числення
Встановлення умови для виконання рекурсивної гілки у функції
кінець

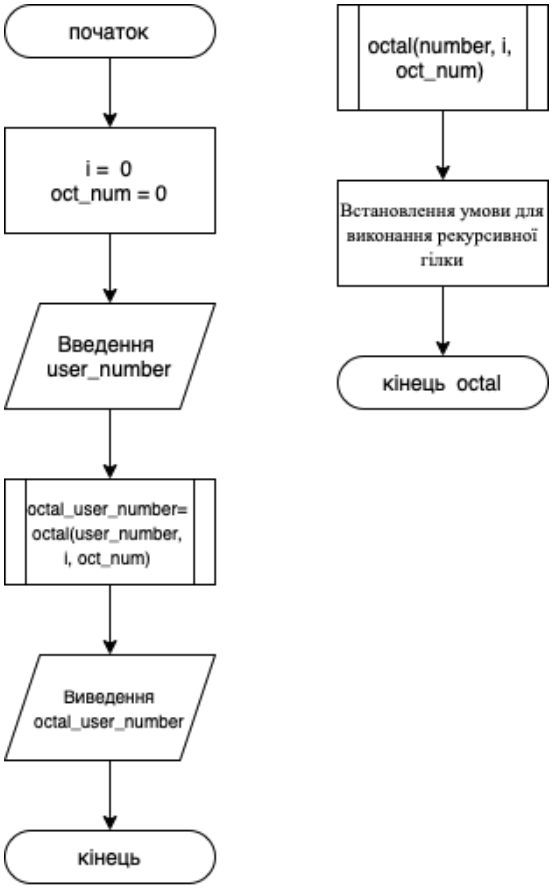
Крок 2
початок
i = 0
oct_num = 0
функція octal(number, i, oct_num)
    Встановлення умови для виконання рекурсивної гілки у функції
кінець octal
введення user_number
octal_user_number = octal(user_number, i, oct_num)
кінець

Крок 3
початок
i = 0
oct_num = 0
функція octal(number, i, oct_num)
    якщо number > 0
        то
          oct_num = oct_num + number%8 * pow(10, i)
           i += 1
          number = number//8
          виконати те ж, що і в останній раз
        інакше
          повернути oct_num
кінець octal
введення user_number
octal_user_number = octal(user_number, i, oct_num)
кінець

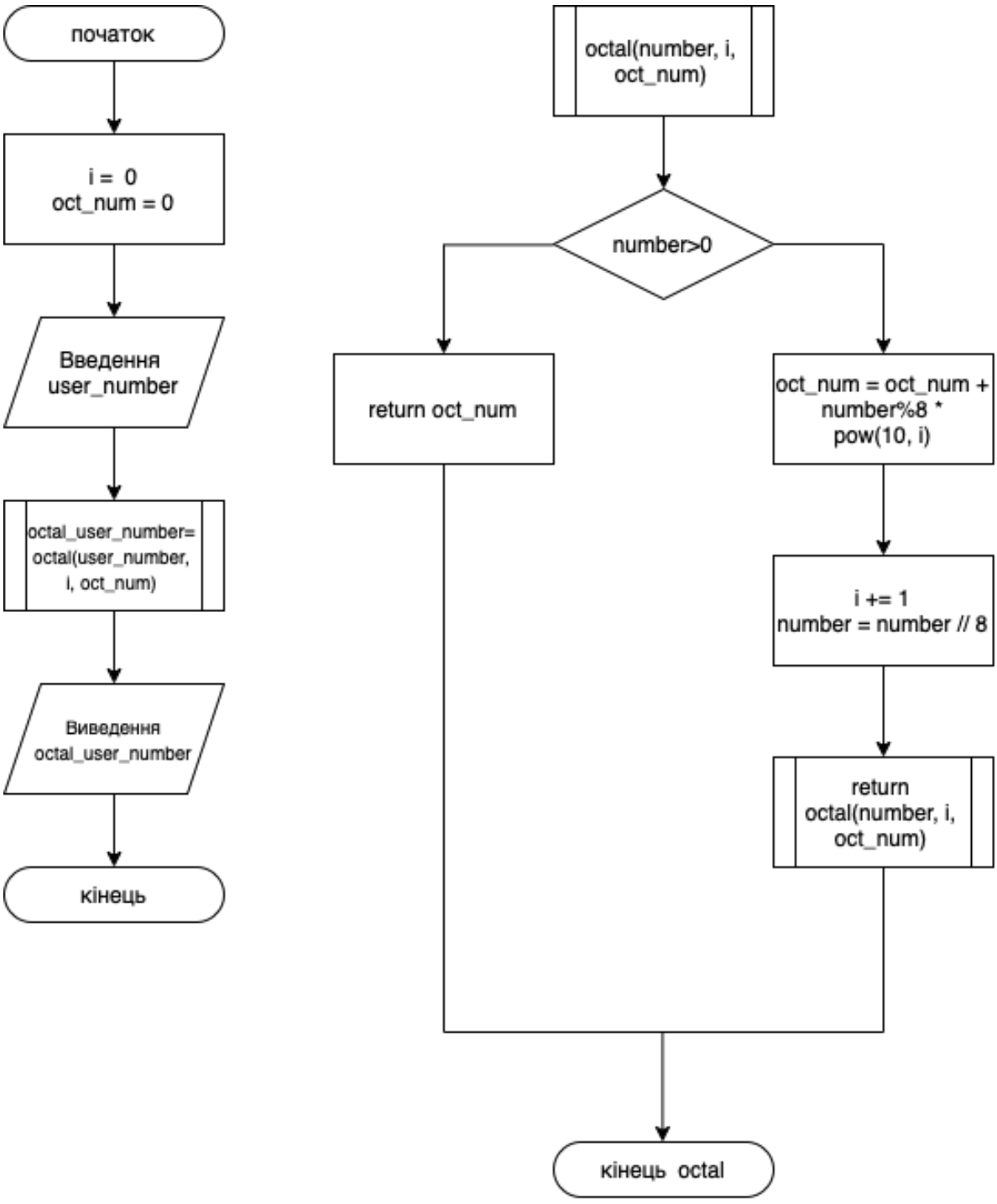
Блок-схема

Крок 1

Крок 2



Крок 3


початок
i = 0
oct_num = 0
Введення
user_number
octal_user_number=
octal(user_number,
i, oct_num)
Виведення
octal_user_number
кінець
octal(number, i,
oct_num)
number>0
return oct_num
oct_num = oct_num +
number%8 *
pow(10, i)
i += 1
number = number // 8
return
octal(number, i,
oct_num)
кінець octal

Код програми на Python

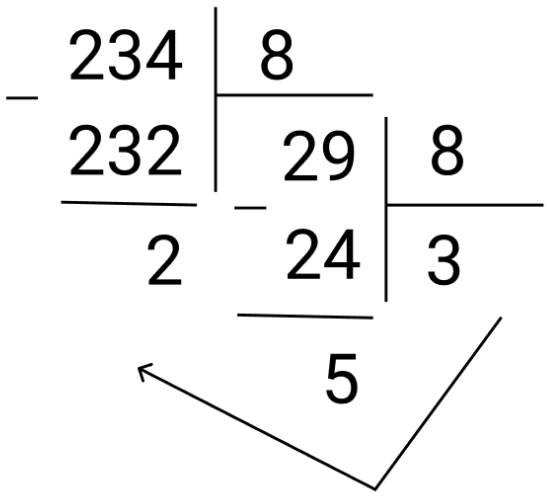
```python
import math
i = 0
oct_num = 0
def octal(number, i, oct_num):
    if number > 0:
        oct_num = oct_num + number%8 * pow(10, i)
        i += 1
        number = number//8
        return octal(number, i, oct_num)
    else:
        return oct_num

user_number = int(input("Enter number"))
octal_user_number = octal(user_number, i, oct_num)
print(octal_user_number)
```

Результат

```
Enter number234
352
```

Перевірка



$$234_{(10)} = 352_{(8)}$$

Висновок: У роботі досліджено особливості роботи рекурсивних алгоритмів, за допомогою яких побудовано функцію, логіка якої базується на результатах багаторазового виклику підпрограми в її тілі. Це дозволяє виконати послідовне ділення числа на 8 і повторювати виклики функції стільки ж разів, скільки розрядів має введене число.