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 рекурсивних алгоритмів» Варіант 2

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

Перевірила Мартинова Оксана Петрівна

( прізвище, ім'я, по батькові)

Київ 2021

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

**Варіант 2**

**Задача**: Обчислення добутку елементів геометричної прогресії, що убуває: початкове значення – 64, кінцеве значення – 1, крок – 4.

**Постановка задачі:**

Вводимо функцію func. Ця функція виконує діюпідрахунку добутку членів геометричної прогресії за допомогою рекурсії. num – саме число, добуток геометричної прогресії якого будемо рахувати, q – це крок зменшення числа. Функція виконується, поки *num* не буде дорівнювати 1. У виводі повертається значення функції після 4 викликів самої себе.

**Побудова математичної моделі:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Змінна** | **Тип** | **Ім’я** | **Призначення** |
| Початкове значення | Натуральне число | *num* | Задане значення |
| Крок | Натуральне число | q | Задане значення |
| Розрахунок суми | Функція | func(num, q) | Розрахунок суми |

**Розв’язання**

Крок 1: Визначити основні дії

Крок 2: Деталізувати функцію знаходження добутку геометричної прогресії

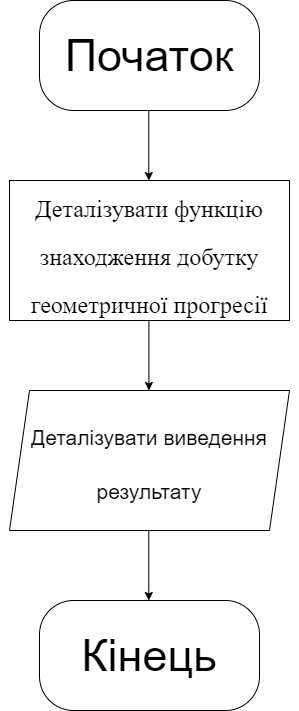
Крок 3: Деталізувати виведення результату

**Псевдокод**

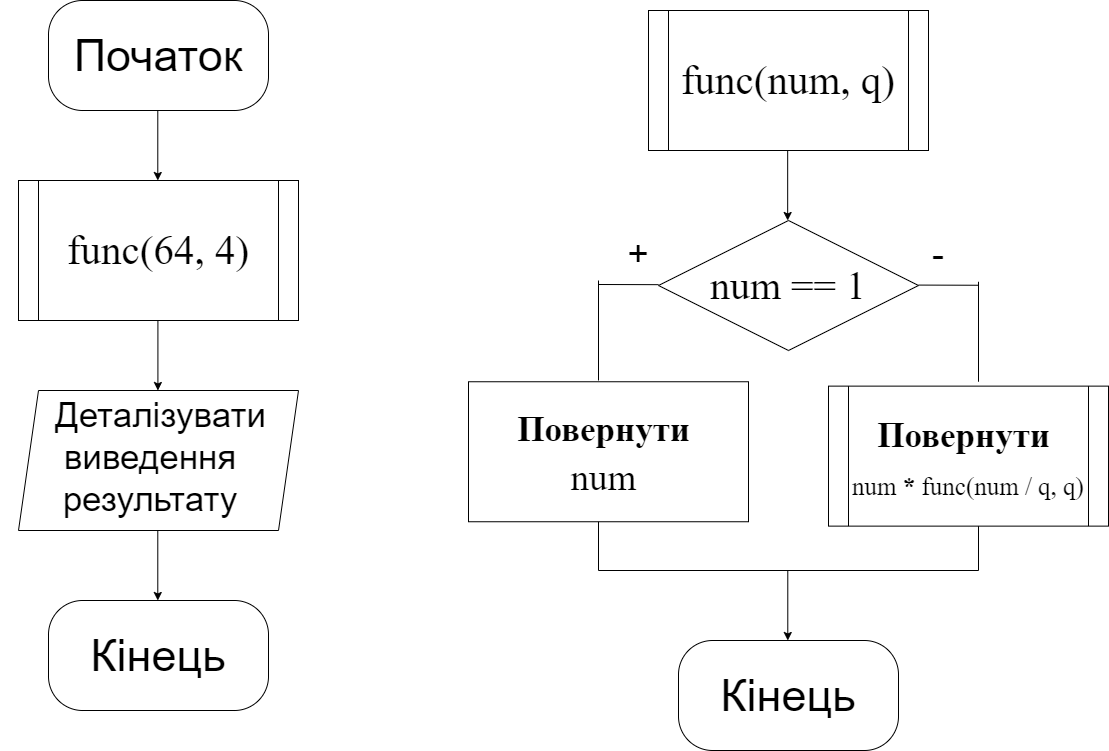
|  |  |
| --- | --- |
| Крок 1 | **Початок**  Деталізувати функцію знаходження добутку геометричної прогресії  Деталізувати виведення результату  **Кінець** |
| Крок 2 | **Початок**  **Функція** :  **якщо** result == 1:  **то**  **повернути**  **інакше:**  **повернути**  **все якщо**  Деталізувати виведення результату  **Кінець** |
| Крок 3 | **Початок**  **Функція** :  **якщо** result == 1:  **то**  **повернути**  **інакше:**  **повернути**  **все якщо**  **Вивести:**  **Кінець** |

**Блок-схема алгоритму**

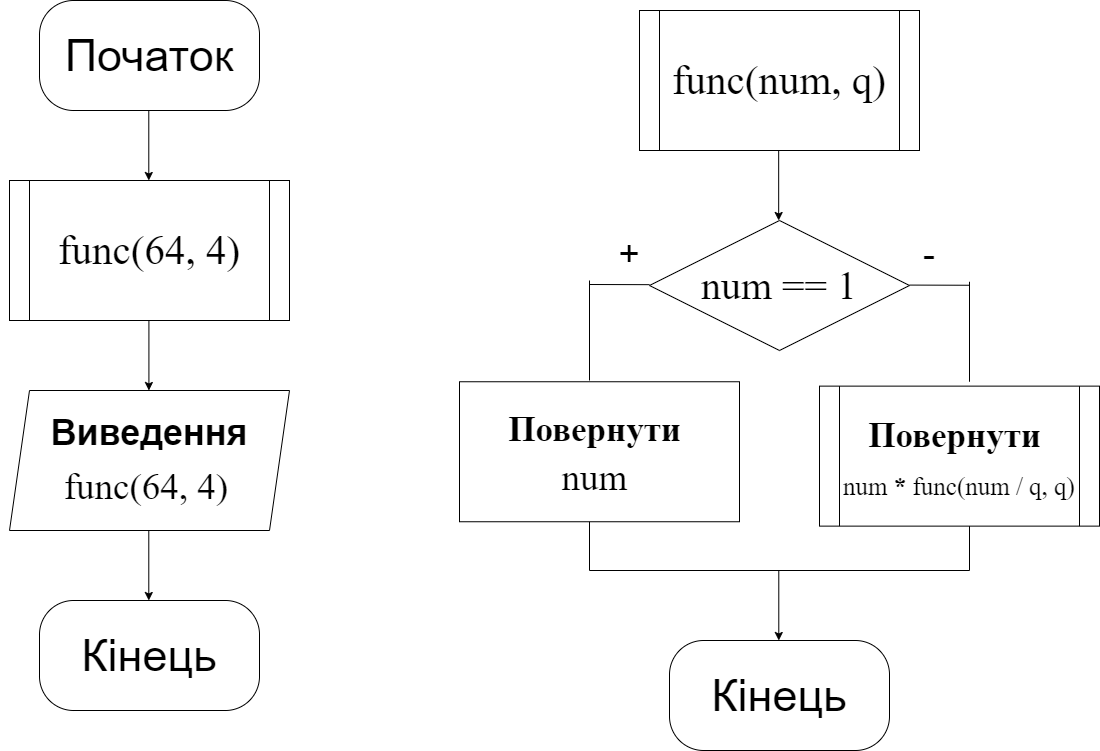
**Крок 1**

****

**Крок 2**

****

**Крок 3**

****

**Випробування алгоритму:**

**Код алгоритму**



**Випробування**

|  |  |
| --- | --- |
| Блок | Дія |
|  | **початок** |
| 1 | 64 == 1 //неправда  **Повернути** 64 \* func(64 / 4, 4)  .  16 == 1 //неправда  **Повернути** 16 \* func(16 / 4, 4)  .  4 == 1 //неправда  **Повернути** 4 \* func(4 / 4, 4)  .  1 == 1 //правда  **Повернути** 1  **Повернути** 1 \* 4 \* 16 \* 64 = 4096 |
| 2 | Вивід: 4096 |
|  | **кінець** |



**Висновок:**

Протягом шостої лабораторної роботи я дослідив особливості роботи

рекурсивних алгоритмів та набув практичних навичок їх використання під час складання програмних специфікацій підпрограм. В результаті виконання

роботи я отримав алгоритм, який рекурсивно знаходить добуток елементів геометричної прогресії, що спадає. Під час виконання роботи були використані оператори альтернативного вибору, функції, та оператор ‘return’ для повернення результату функції.