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 32

Виконав студент ІП-14, Шляхтун Денис Михайлович

(шифр, прізвище, ім'я, по батькові)

Перевірив доц. кафедри ІПІ 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**. Дано перший член і різницю арифметичної прогресії, знайти суму n перших членів прогресії.

**Постановка задачі**. Результатом розв’язку є сума членів арифметичної прогресії. Ввідні дані, що вимагаються для виконання алгоритму: значення першого члена прогресії, різниця прогресії, кількість членів прогресії. Сума n перших членів прогресії буде обраховуватися за допомогою рекурсивної функції.

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

|  |  |  |  |
| --- | --- | --- | --- |
| ***Змінна*** | ***Тип*** | ***Ім’я*** | ***Призначення*** |
| Перший член прогресії | Дійсний | num | Проміжне дане |
| Різниця прогресії | Дійсний | diff | Проміжне дане |
| Кількість членів прогресії | Цілий | n | Проміжне дане |
| Результат | Дійсний | result | Результат |

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

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо дію знаходження суми прогресії.

**Псевдокод.**

Крок 1.

**початок**

знаходження суми прогресії

**кінець**

Крок 2.

**початок**

result = sum (num, diff, n)

**кінець**

**функція** sum (num, diff, n)

**якщо** n > 1

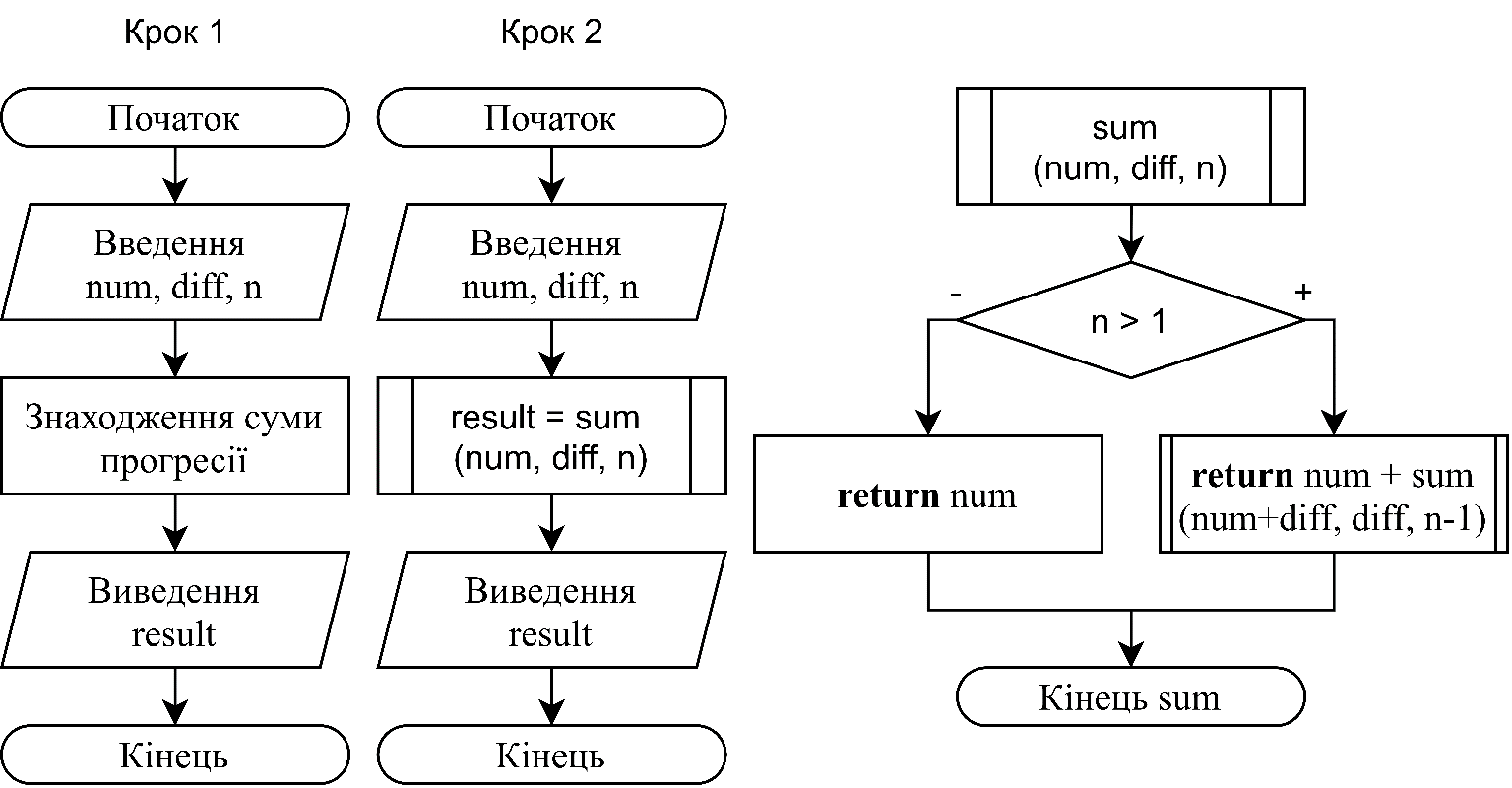
**то повернути** num + sum (num + diff, diff, n - 1)

**інакше повернути** num

**все якщо**

**кінець функції**

**Блок-схема алгоритму**



**Код програми.**

**Зображення, що містить текст, монітор, екран

Автоматично згенерований опис**

**Результат виконання програми.**

**Зображення, що містить текст, екран

Автоматично згенерований опис**

**Випробування алгоритму.**

|  |  |
| --- | --- |
| ***Заглиблення***  ***рекурсії*** | ***Дія*** |
|  | Початок |
|  | Введення: num = 1, diff = 4, n = 4 |
| **1** | sum (1, 4, 4) |
|  | n > 1 – true |
| **2** | sum (5, 4 ,3) |
|  | n > 1 – true |
| **3** | sum (9, 4, 2) |
|  | n > 1 – true |
| **4** | sum (13, 4, 1) |
|  | n > 1 – false |
|  | return 13 |
| **3** | return 13 + 9 = 22 |
| **2** | return 22 + 5 = 27 |
| **1** | return 27 + 1 = 28 |
|  | result = 28 |
|  | Виведення 28 |
|  | Кінець |

Перевіримо правильність відповіді: 1 + 5 + 9 + 13 = 28, отже відповідь правильна.

**Висновок.**

При виконанні лабораторної роботи було використано рекурсивну підпрограму, особливістю якої є виклик у тілі функції самої себе. Таким чином, рекурсивна підпрограма може бути використана замість арифметичних чи ітераційних циклів. Головною умовою для роботи рекурсивної функції є оператор розгалуження (у випадку даної лабораторної роботи – альтернативного типу).