**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.

**Постановка задачі:** Для рекурентного обчислення добудку елементів арифметичної прогресії, ми використаємо підпрограмму, що буде повертати an\* an+1, де

**Математична модель:**

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Назва | Тип | Призначення |
| 1 | result | Integer | Змінна для збереження добутку елементів |
| 2 | arifNum | Integer | Функція, для n-ного члена прогресії |
| 3 | start | Integer | Початкове значення |
| 4 | step | Integer | Крок прогресії |
| 5 | end | Integer | Кінцеве значення |

**Псевдокод**:

**Основна програма:**

**Крок1**

**Початок**

—Ініціалізуємо змінні

—Обчислити добуток

—Вивести результат

**Кінець**

**Крок2**

**Початок**

—start = 35, step = -10,

end = 5

—Обчислити добуток

—Вивести результат

**Кінець**

**Крок3**

**Початок**

— start = 35, step = -10,

end = 5

—result = arifNum(start,step, end)

—Вивести результат

**Кінець**

**Підпрограми:**

**arifNum(s,st,e)**

**якщо** s == e

**то**

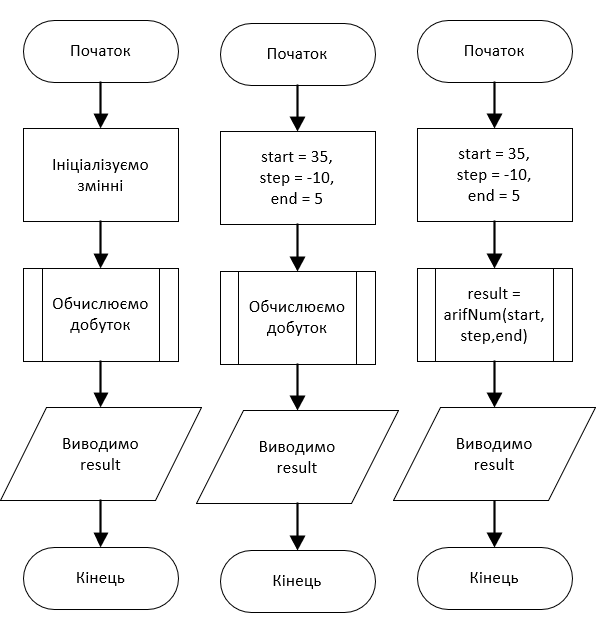
**return** s

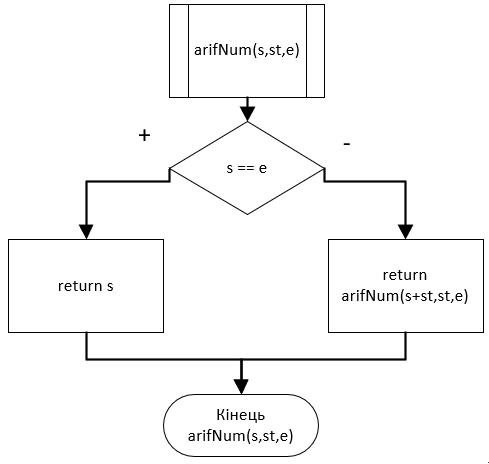
**інакше**

**return s \* arifNum(s+st, st, e)**

**Кінець arifNum**

**Блок-схема:**

****

****

Перевірка алгоритму:

|  |  |
| --- | --- |
| Блок | Дія |
|  | Початок |
| 1 | start = 35, step = -10, end = 5 |
| 2 | arifNum(35,-10,5) => 35 \* arifNum(25,-10,5) |
| 3 | arifNum(25,-10,5) => 25 \* arifNum(15,-10,5) |
| 4 | arifNum(15,-10,5) => 15 \* arifNum(5,-10,5) |
| 5 | arifNum(5,-10,5) => 5 |
| 6 | result = 35\*25\*15\*5 = 65625 |
|  | Кінець |

**Висновок:**

Дослідив особливості роботи рекурсивних алгоритмів та набув практичних навичок їх використання під час складання програмних специфікацій підпрограм.

**Код на Python:**

def rec\_mult(start, step, end):

    if start == end:

        return start

    return start \* rec\_mult(start + step, step, end)

print(rec\_mult(35,-10,5))

