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 31

Виконав студент ІП-11 Трикош Іван Володимирович

Перевірила Мартинова О. П.

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.

Варіант 31. Дано перший член і різницю арифметичної прогресії, знайти n-й член прогресії.

**Постановка задачі –** обчислити n-й член арифметичної прогресії можна за допомогою рекурсивної функції: додаємо різницю і знову викликаємо функцію, це повторюється n-1 разів, далі додаємо перший член і виводимо результат.

**Побудова математичної моделі**

Складемо таблицю імен змінних:

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| Перший член | Дійсний | a | Початкове дане |
| Різниця прогресії | Дійсний | d | Початкове дане |
| Кількість членів | Цілий додатний | n | Початкове дане |
| Параметр функції | Цілий додатний | k | Проміжна змінна |
| n-й член | Дійсний | an | Результат |

Таким чином, математичне формулювання задачі зводиться до виклику рекурсивної функції n-1 разів і при цьому додавати різницю арифметичної прогресії; після цього додати перший член прогресії і вивести результат.

**Псевдокод алгоритму**

*Крок 1. Визначимо основні дії.*

*Крок 2. Деталізуємо дію введення першого члена, різниці та кількості членів арифметичної прогресії.*

*Крок 3. Деталізуємо дію обчислення n-члена арифметичної прогресії.*

*Крок 4. Визначаємо підпрограму(функцію).*

*Крок 5. Деталізуємо умову для рекурсії.*

*Крок 6. Деталізуємо певернення результату.*

*Крок 5. Виводимо n-член прогресії*

**Основна програма**

**Початок**

Ввід a, d, n

an = dif\_y(n) + a

Вивід an

**Кінець**

**Підрограма**

dif\_y (k)

**якщо** (k > 1) **то**

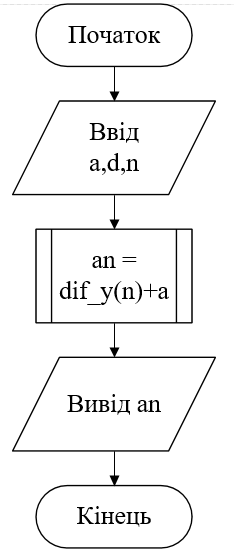
**повернути** d + y(k-1)

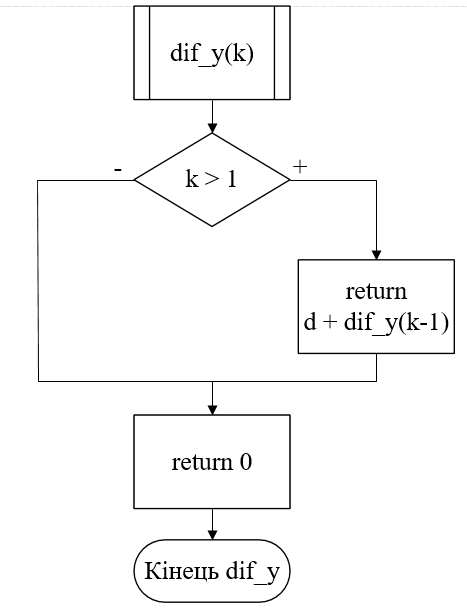
**все якщо**

**повернути** 0

**Кінець**

**Блок-схема алгоритму**



****

**Код програми на C++:**

#include <iostream>

using namespace std;

double dif\_y(int);

double a,d;

int n;

int main()

{

double an;

cout << "Enter the first term of the arithmetic progression: ";

cin >> a;

cout << "Enter the difference of the arithmetic progression: ";

cin >> d;

cout << "Enter n > 0: ";

cin >> n;

an = a + dif\_y(n);

cout << n << "-term of the arithmetic progression = " << an;

}

double dif\_y(int k)

{

if (k > 1)

{

return d + dif\_y(k-1);

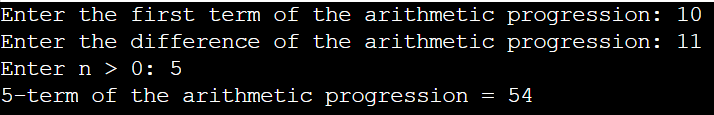
}

return 0;

}

**Випробування алгоритму**

|  |  |
| --- | --- |
| Блок | Дія |
|  | Початок |
| 1 | Ввід a = 10; d = 11; n = 5 |
|  | Перший виклик рекурсивної функції: |
| 2 | k = 5 |
| 3 | k > 5 |
| 4 | an = 11 |
|  | Останній виклик рекурсивної функції: |
| 5 | k = 2 |
| 6 | k > 1 |
| 7 | an = 44 |
| 8 | an = 54 |
| 9 | Вивід 54 |
|  | Результат: 54 |
|  | Кінець |

****

**Висновок –** дослідив особливості роботи рекурсивних алгоритмів та набув практичних навичок їх використання під час складання програмних специфікацій підпрограм, удосконалив свої навички розв’язання прикладних задач з програмування та покращив свої знання в програмуванні на мові C++. У результаті виконання програми з введеними значеннями першого члена, різницю та кількості членів арифметичної прогресії я одержав значення n-го члена арифметичної прогресії.