

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 ІІІ-15 Ткач Владислав Анатолійович

Перевірив: _____

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.

Завдання

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

Постановка задачі

Результатом розв'язку є n -й член арифметичної прогресії, який знайдемо за допомогою рекурсії. Початкові дані: перший член, різниця арифметичної прогресії та номер шуканого елемента.

Побудова математичної моделі

Складемо таблицю імен змінних:

Змінна	Тип	Ім'я	Призначення
Перший член	Дійсне	a_1	Початкове дане
Різниця прогресії	Дійсне	d	Початкове дане
Номер члена	Ціле	n	Початкове дане
n -й член прогресії	Функція	Progress	Обрахунок даних
n -й член	дійсне	res	Результат

Для визначення результат використаємо рекурсію – функція **Progress()** яка запускає сама себе, приймає у собі значення першого члена – a_1 арифметичної прогресії, різницю - d та номер члена прогресії – n . У середині функції відбувається перевірка значення n для того, щоб перевірити на якому члені знаходиться функція, що реалізується за допомогою стеку, за рахунок цього ж

стеку додавання **d** починається з 2 члену прогресії що дає аналог покрокового знайодження **n-го** члену прогресії. Якщо умова **n > 1** не виконується, функція повертає значення **res** – це і буде наш результат.

Розв'язання

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо дію знаходження n-го члена арифметичної прогресії

Псевдокод алгоритму

Крок 1.

Початок

Ввід a1, d, n

знаходження n-го члена арифметичної прогресії

за допомогою функції Progress()

Вивід res

Кінець

Крок 2.

Початок

Ввід a1, d, n

res = Progress(a1, d, n)

Вивід res

Кінець

Підпрограма:

Progress(a1, d, n)

res = a1

якщо n > 1

res = Progress(res + a1, d, n - 1)

все якщо

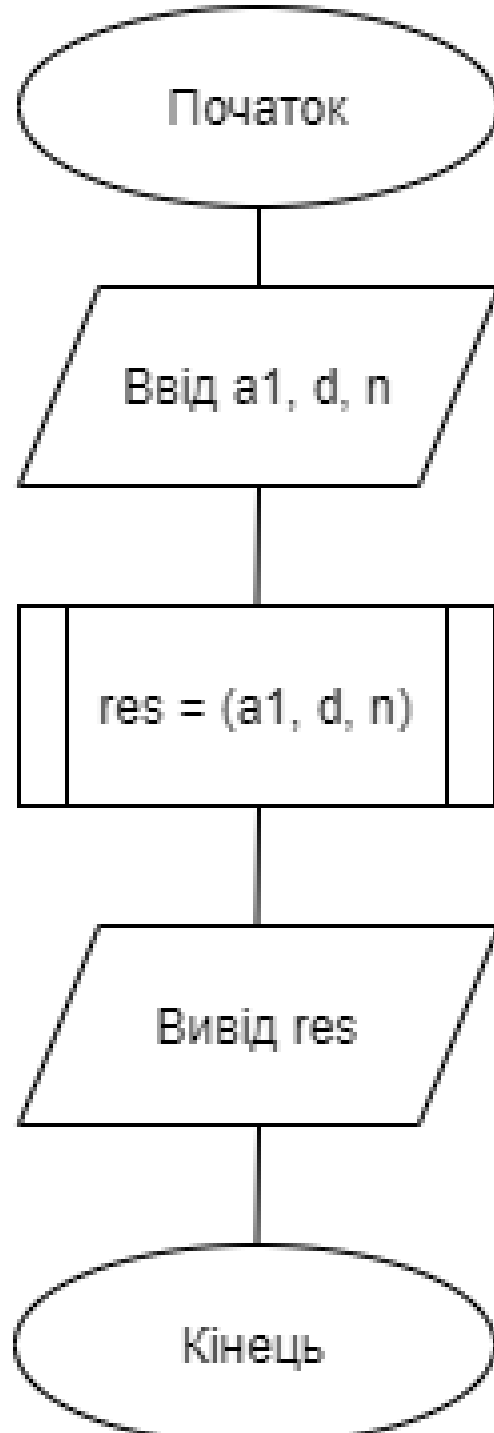
повернути res

Блок схема алгоритму

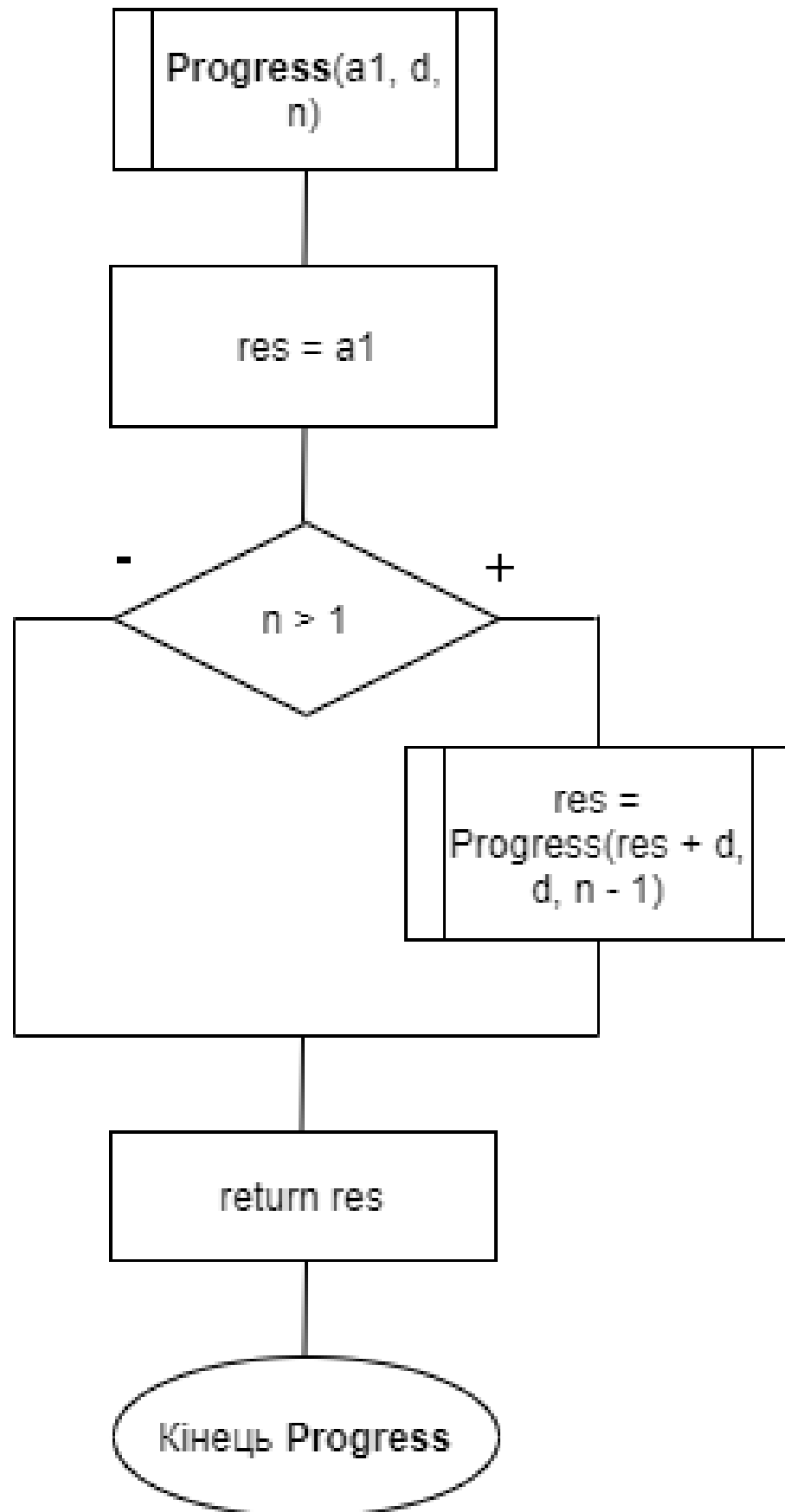
Крок 1



Крок 2



Підпрограма



Код програми

```
1  #include <bits/stdc++.h>           //підключення бібліотек
2  using namespace std;
3
4  float Progress(float a1, int d, int n){    //оголошення функції
5      float res = a1;                    //оголошення даних
6      if(n > 1){                          //умова виходу з рекурсії
7          res = Progress(res + d, d, n - 1); //рекурсія: програма запускає сама себе
8      }
9      return res;                        //повернення результату res
10 }
11
12 int main(){
13     float a1, d;                        //оголошення даних
14     int n;                              //оголошення даних
15     cout << "first element: ";
16     cin >> a1;                          //введення даних
17     cout << "difference: ";
18     cin >> d;                            //введення даних
19     cout << "elements number : ";
20     cin >> n;                            //введення даних
21     double res = Progress(a1, d, n);     //присвоєння значення, яке повертає функція - res
22     cout << n << "th element - " << res; //виведення даних
23 }
```

Випробування програми

```
first element: 2
difference: 4
elements number : 5
5th element - 18
-----
Process exited after 3.198 seconds with return value 0
Press any key to continue . . .
```

```
first element: 5.5
difference: -2.4
elements number : 5
5th element - -2.5
-----
Process exited after 7.787 seconds with return value 0
Press any key to continue . . .
```

```
first element: 159
difference: 9547.5
elements number : 1
1th element - 159
-----
Process exited after 11.42 seconds with return value 0
Press any key to continue . . .
```

Висновки

Ми дослідили особливості роботи рекурсивних алгоритмів та набули практичних навичок їх використання під час складання програмних специфікацій підпрограм.

У результаті виконання лабораторної роботи ми отримали алгоритм для знаходження n -го члену арифметичної прогресії за допомогою рекурсії: визначили основні дії, деталізували дію знаходження n -го члену прогресії за допомогою функції, яка викликає сама себе.