

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 лінійних алгоритмів»

Варіант __30__

Виконав студент _____ ІП-15 Розін Олексій Іванович _____
(шифр, прізвище, ім'я, по батькові)

Перевірив _____ Вечерковська Анастасія Сергіївна _____
(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.

Індивідуальне завдання

Варіант 30

Постановка задачі

Дано перший член і знаменник геометричної прогресії. Обчислити суму n перших членів прогресії та знайти n -й член прогресії.

Математична модель

Змінна	Тип	Ім'я	Призначення
Перший член прогресії	Дійсний	firstEl	Вхідні дані
Знаменник прогресії	Дійсний	denominator	Вхідні дані
Порядковий номер члена прогресії	Цілий	n	Вхідні дані
Сума n членів прогресії	Дійсний	sum	Результат
Значення n члена прогресії	Дійсний	nVal	Результат
Пошук суми n членів прогресії	Процедура	progressionSum	Проміжні дані
Пошук значення n члена прогресії	Процедура	findProgressionElement	Проміжні дані

Користувач задає значення firstEl, denominator та n. Перша рекурсивна функція progressionSum приймає 3 параметра: first, denom, n (first та denom дійсного типу, n - цілого). В тілі функції перевіряємо якщо $n \leq 0$, то повертаємо 0 (return 0), інакше повертаємо $\text{first} + \text{progressionSum}(\text{first} * \text{denom}, \text{denom}, n - 1)$. Друга рекурсивна функція findProgressionElement приймає такі ж параметри (first, denom, n). В тілі цієї функції перевіряємо, якщо $n \leq 1$, то повертаємо перший аргумент first (return first), інакше повертаємо $\text{findProgressionElement}(\text{first}, \text{denom}, n - 1) * \text{denom}$. Ініціалізуємо змінну sum значенням, яке поверне визов функції progressionSum з параметрами (firstEl, denom, n). Ініціалізуємо змінну nVal значенням, яке поверне визов функції findProgressionElement з параметрами (firstEl, denom, n). Вивід sum та nVal.

Розв'язання

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

Крок 1. Визначимо основні дії.

Крок 2. Введення firstEl, denominator, n.

Крок 3. Ініціалізація sum.

Крок 4 Ініціалізація nVal.

Крок 5. Вивід sum, nVal.

Псевдокод – основна програма

Крок 1

початок

введення firstEl, denominator, n

ініціалізація sum

ініціалізація nVal

вивід sum, nVal

кінець

Крок 2

початок

введення firstEl, denominator, n

sum = progressionSum(firstEl, denominator, n)

ініціалізація nVal

вивід sum, nVal

кінець

Крок 3

початок

введення firstEl, denominator, n

sum = progressionSum(firstEl, denominator, n)

`nVal = findProgressionElement(firstEl, denominator, n)`

вивід sum, nVal

кінець

Крок 4

початок

введення firstEl, denominator, n

`sum = progressionSum(firstEl, denominator, n)`

`nVal = findProgressionElement(firstEl, denominator, n)`

вивід sum, nVal

кінець

Псевдокод – підпрограми

progressionSum(first, denom, n)

якщо `n <= 0`

то повернути 0

все якщо

повернути `first + progressionSum(first * denom, denom, n - 1)`

кінець

findProgressionElement(firstEl, denominator, n)

якщо `n <= 1`

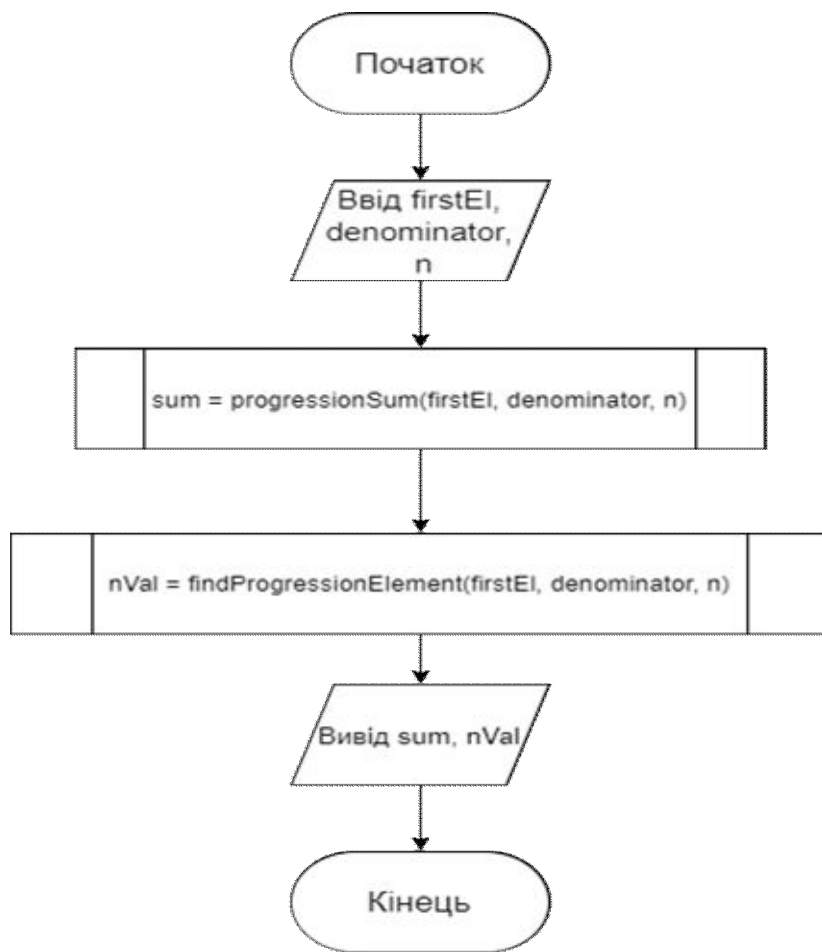
то повернути first

все якщо

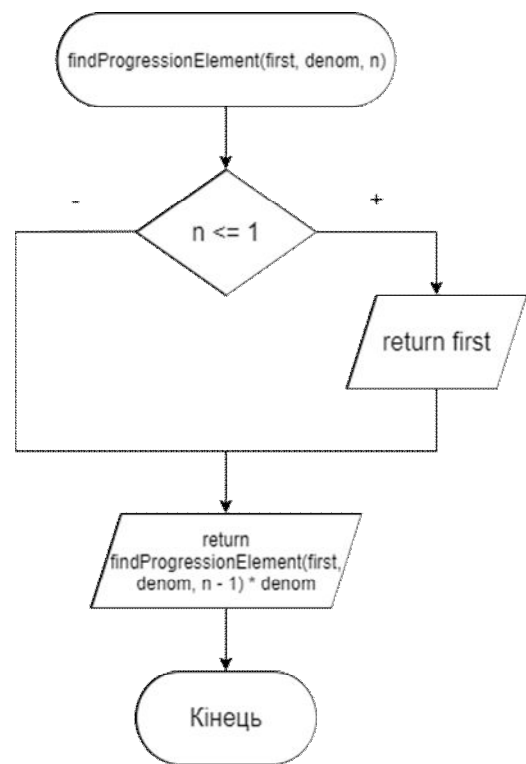
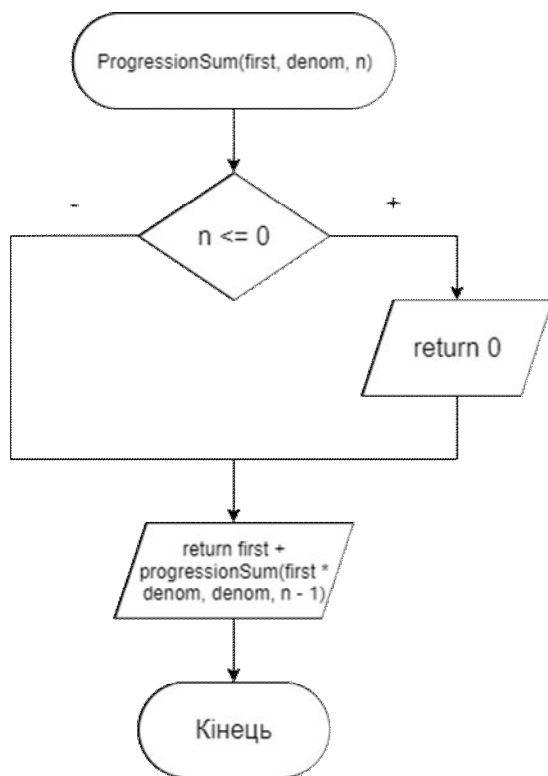
повернути `findProgressionElement(first, denom, n - 1) * denom`

кінець

Блоксхема – основна програма



Блоксхема - підпрограми



Код програми

```
lab6 (Глобальная область)
1  #include <iostream>
2  using namespace std;
3
4  double progressionSum(double first, double denom, int n);
5  double findProgressionElement(double first, double denom, int n);
6
7  int main()
8  {
9      setlocale(LC_ALL, "");
10     double firstEl, denominator;
11     int n;
12     cout << "Введите первый член геометрической прогрессии: ";
13     cin >> firstEl;
14     cout << "Введите знаменатель прогрессии: ";
15     cin >> denominator;
16     cout << "Введите кол-во членов прогрессии: ";
17     cin >> n;
18     double sum = progressionSum(firstEl, denominator, n);
19     double nVal = findProgressionElement(firstEl, denominator, n);
20     cout << "Сумма " << n << " первых членов прогрессии: " << sum << endl;
21     cout << n << "-ый член прогрессии: " << nVal << endl;
22 }
23
24 double progressionSum(double first, double denom, int n) {
25     if (n <= 0) {
26         return 0;
27     }
28     return first + progressionSum(first * denom, denom, n - 1);
29 }
30
31 double findProgressionElement(double first, double denom, int n) {
32     if (n <= 1) {
33         return first;
34     }
35     return findProgressionElement(first, denom, n - 1) * denom;
36 }
```

```
Консоль отладки Microsoft Visual Studio
Введите первый член геометрической прогрессии: 5
Введите знаменатель прогрессии: 3
Введите кол-во членов прогрессии: 3
Сумма 3 первых членов прогрессии: 65
3-ый член прогрессии: 45
```

Випробування

Блок	Дія
	Початок
1	firstEl = 5; denominator = 3; n = 3
2	sum = progressionSum(firstEl, denominator, n)
2.1	Виклик progressionSum(firstEl, denominator, n)
2.2	n == 3 n <= 0 == false

	return first + progressionSum(first * denom, denom, n - 1)
2.3	Виклик progressionSum(firstEl, denominator, n-1)
2.4	n == 2 n <= 0 == false return first + progressionSum(first * denom, denom, n - 1)
2.5	Виклик progressionSum(firstEl, denominator, n-1)
2.6	n == 1 n <= 0 == false return first + progressionSum(first * denom, denom, n - 1)
2.7	Виклик progressionSum(firstEl, denominator, n-1)
2.8	n == 0 n <= 0 == true return 0
3	nVal = findProgressionElement(firstEl, denominator, n)
3.1	Виклик findProgressionElement(firstEl, denominator, n)
3.2	n == 3 n <= 1 == false return findProgressionElement(first, denom, n - 1) * denom
3.3	Виклик findProgressionElement(firstEl, denominator, n-1)
3.4	n == 2 n <= 1 == false return findProgressionElement(first, denom, n - 1) * denom
3.5	Виклик findProgressionElement(firstEl, denominator, n-1)
3.6	n == 1 n <= 1 == true return first
4	sum = 65 nVal = 45
5	Вивід sum, nVal
	Кінець

Висновки

Ми дослідили особливості роботи рекурсивних алгоритмів та набули практичних навичок їх використання під час складання програмних специфікацій підпрограм. У цій лабораторній роботі за допомогою двох рекурсивних функцій ми обчислили суму перших n членів та значення n члена даної нам геометричної прогресії.