

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 5

Виконав студент

ІП-13 Вальчишен Ярослав Олександрович

(шифр, прізвище, ім'я, по батькові)

Перевірів

\_\_\_\_\_  
( прізвище, ім'я, по батькові)

## 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 6.

Обчислити суму 6 елементів геометричної прогресії, що зростає: початкове значення – 2, крок – 2

#### 1. Постановка задачі

Результатом розв’язку є сума перших 6 елементів геометричної прогресії, що зростає

#### 2. Побудова математичної моделі

Таблиця змінних

Змінна	Тип	Ім'я	Призначення
Перший член прогресії	Дійсний	firstMember	Початкове дане
Знаменник прогресії	Натуральний	q	Початкове дане
Кількість членів	Натуральний	n	Початкове дане
Сума прогресії	Дійсний	sum	Вихідні дані

Таблиця функцій

Назва	Синтаксис	Призначення
Піднесення до степеню	pow(a, b)	Піднесення a в степінь b

Таким чином математичне формулювання задачі зводиться до реалізації рекурсивного алгоритму, описаного в постановці задачі.

Рекурсивна функція повинна буде приймати 1 параметр: Progression(i = 1), де currentSum – це лічильник рекурсії.

#### 3. Розв’язання

Програмні специфікації запишемо у псевдокодi та графічній формi у вигляді блок-схеми.

Крок 1. Визначимо основні дії

Крок 2. Деталізуємо дію оголошення змінних

Крок 3. Деталізуємо дію виконання рекурсії

#### Псевдокод

Крок 1

**Функція Progression(i = 1)**

Реалізація рекурсії

**Все функція**

**початок**

Оголошення змінних

Виклик функції Progression()

Виведення **sum**

**кінець**

Крок 2

**Функція Progression(i = 1)**

Реалізація рекурсії

**Все функція**

**початок**

firstMember := 2

q := 2

n := 6

Виклик функції Progression()

Виведення **sum**

**кінець**

Крок 3

**Функція Progression(currentSum)**

**якщо** n >= i

**то**

sum := sum + firstMember \* pow(q, i - 1)

Progression(i + 1)

**інакше**

**все якщо**

**Все функція**

**початок**

firstMember := 2

q := 2

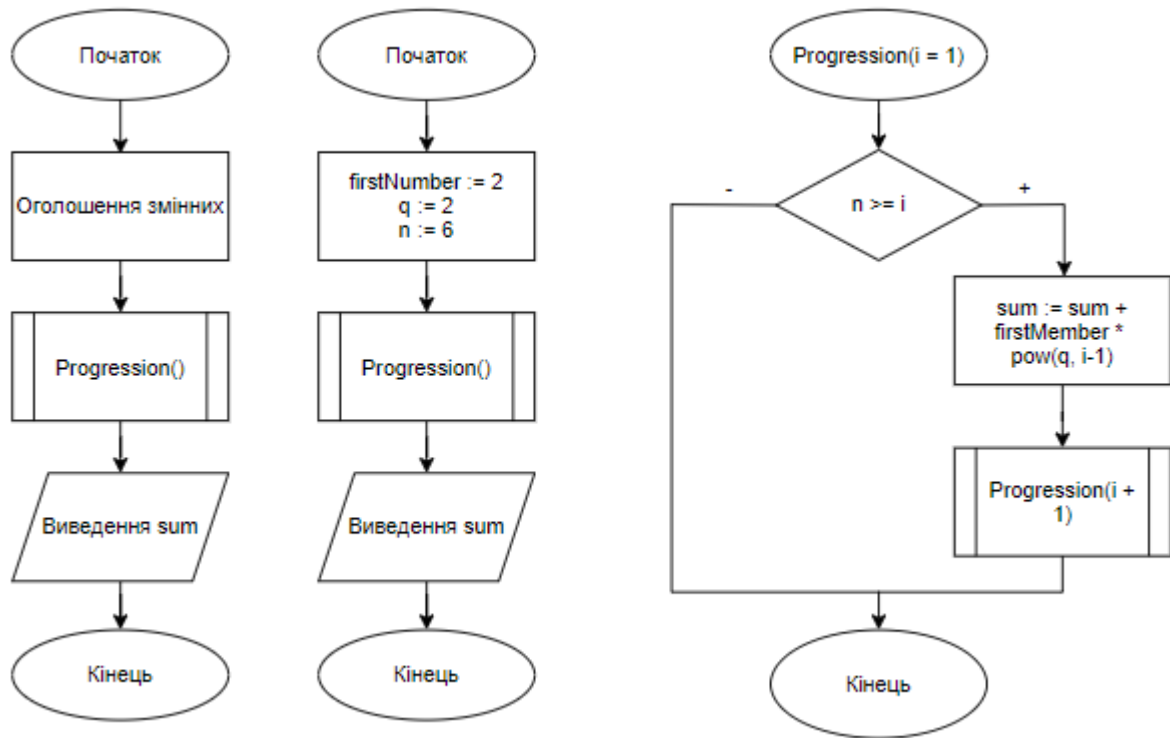
n := 6

Виклик функції Progression(sum)

Виведення **sum**

**кінець**

**Блок-схема**



## Код програми

```

1  using System;
2
3  namespace AsdLab6
4  {
5      Ссылка: 0
6      class Program
7      {
8          Ссылка: 0
9          static void Main(string[] args)
10         {
11             double sum = 0;
12             double firstMember = 2;
13
14             int q = 2;
15             int n = 6;
16
17             void Progression(int i = 1)
18             {
19                 if (n >= i)
20                 {
21                     sum += firstMember * Math.Pow(q, i - 1);
22                     Console.WriteLine("Sum" + i + ": " + sum);
23                     Progression(i + 1);
24                 }
25             }
26
27             Progression();
28             Console.WriteLine(sum);
29         }
30     }
31 }
32

```

### Випробування алгоритму

Блок	Дія
	Початок
1	$i := 1;$ $sum := sum + 2 * pow(2, i - 1);$ $sum := 2$
2	$i := 2;$ $sum := sum + 2 * pow(2, i - 1);$ $sum := 6$
3	$i := 3;$ $sum := sum + 2 * pow(2, i - 1);$ $sum := 14$
4	$i := 4;$ $sum := sum + 2 * pow(2, i - 1);$ $sum := 30$
5	$i := 5;$ $sum := sum + 2 * pow(2, i - 1);$ $sum := 62$
6	$i := 6;$ $sum := sum + 2 * pow(2, i - 1);$ $sum := 126$
7	Виведення <b>sum</b>
	Кінець

### Висновок

Виконуючи лабораторну роботу, я дослідив особливості роботи рекурсивних алгоритмів та набув практичних навичок їх використання під час складання програмних специфікацій.