

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 26

Виконав студент: ІП-15 Поліщук Валерій Олександрович

(шифр, прізвище, ім'я, по батькові)

Перевірила: Вечерковська Анастасія Сергіївна

(прізвище, ім'я, по батькові)

Київ 2021

Лабораторна робота №6

Дослідження рекурсивних алгоритмів

Варіант 26

Мета – дослідити особливості роботи рекурсивних алгоритмів та набути практичних навичок їх використання під час складання програмних специфікацій підпрограм.

Постановка задачі

Задано натуральне m та n . Обчислити

$$\sum_{k=m}^n \frac{(-1)^k}{k!} \left(\frac{a_k + 2}{3} \right)^k \quad a_0 = 1, a_k = \sqrt{|4a_{k-1} + 2|}$$

Математична модель

Змінна	Тип	Ім'я	Призначення
Значення m	Натуральне	m	Вхідні дані
Значення n	Натуральне, $\geq m$	n	Вхідні дані
Значення i	Ціле	i	Проміжні дані
Значення sum	Дійсне	sum	Вихідні дані
Функція, що рахує факторіал числа	Функція	Factorial	Рахує факторіал числа
Функція, що рахує значення $a(k)$	Функція	A	Рахує $a(k)$

\wedge - піднесення до степеню

Sqrt() – корінь з числа

Abs() – модуль числа

Ми знаходимо суму в циклі основної програми, а факторіали та значення $a(k)$ рахуються у функціях підпрограми

Розв'язання

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

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо процес обчислення результату sum

Псевдокод

Основна програма

Крок 1

початок

введення n, m

обчислення результату sum за допомогою функцій A та Factorial

виведення sum

кінець

Крок 2

початок

введення n, m

sum = 0

повторити

для i від m до n

sum = sum + $(-1)^i / \text{Factorial}(i) * ((A(i) + 2) / 3)^i$

все повторити

виведення sum

кінець

Підпрограма

Factorial(i)

початок

якщо $i = 0$

то

 повернути 1

все якщо

 повернути $(i * (\text{Factorial}(i - 1)))$

кінець

A(i)

початок

якщо $i = 0$

то

 повернути 1

все якщо

 повернути $\text{Sqrt}(\text{Abs}(4 * A(i-1) + 2))$

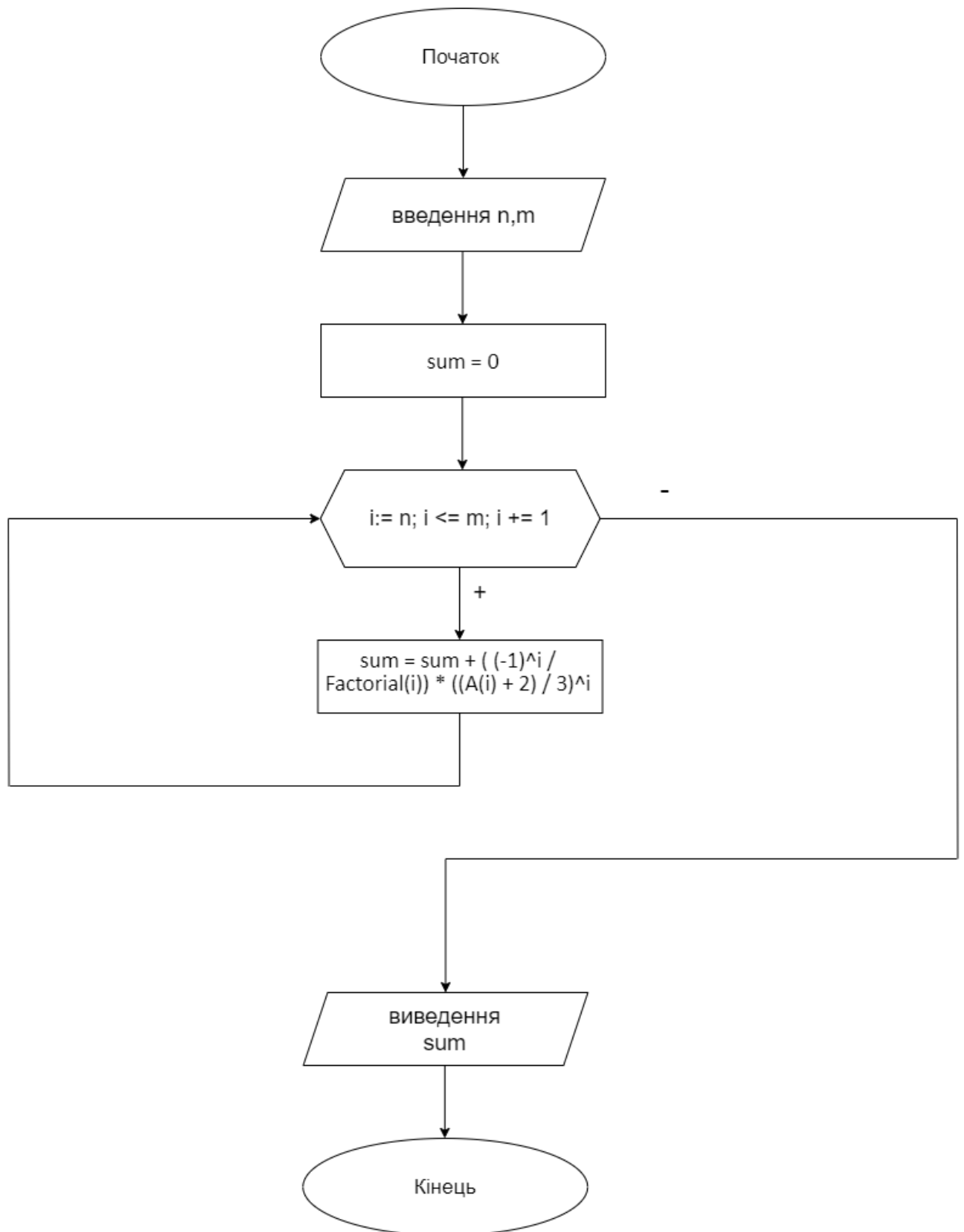
кінець

Блок-схема

Крок 1

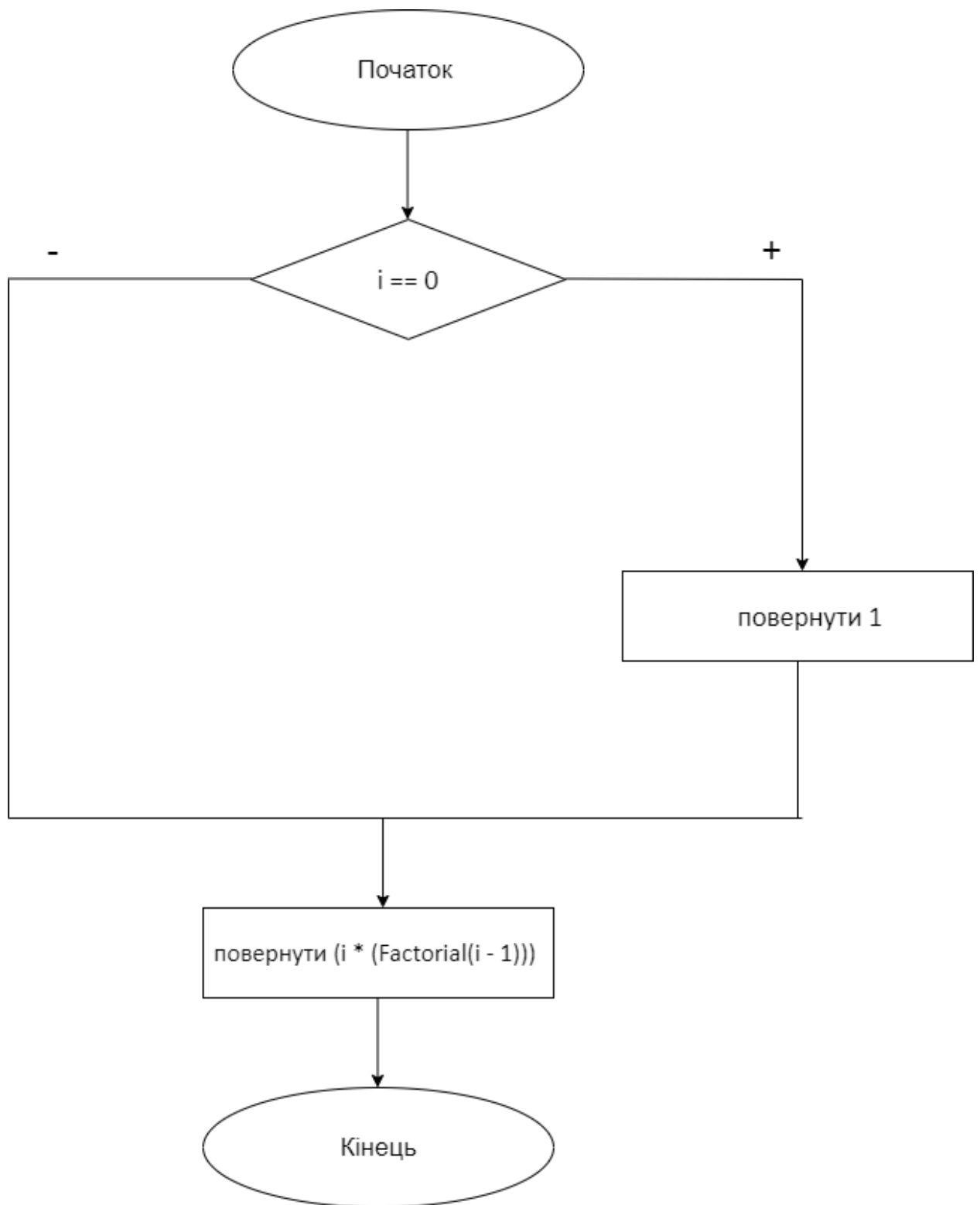


Крок 2

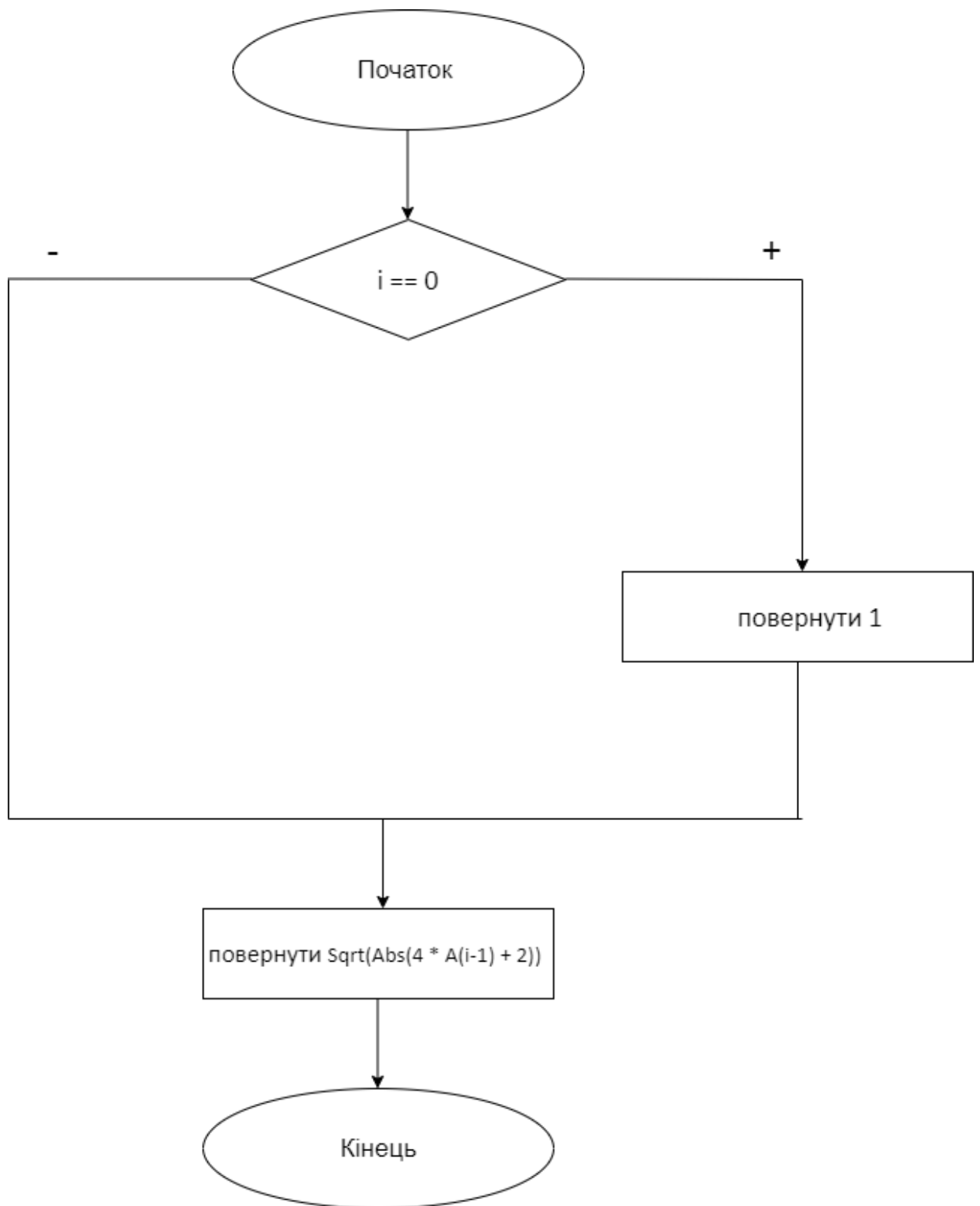


Підпрограма

Factorial(i)



A(i)



Код програми

```
class Program
{
    Ссылка: 2
    private static double A (int i)
    {
        if (i == 0) { return (1); }
        return (Math.Sqrt(Math.Abs(4 * A(i-1) + 2)));
    }

    Ссылка: 2
    private static double Factorial(int i)
    {
        if (i == 0) return (1);
        return (i * (Factorial(i - 1)));
    }

    Ссылка: 0
    static void Main(string[] args)
    {
        int n, m;
        double sum;
        Console.WriteLine("Enter n");
        n = Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("Enter m");
        m = Convert.ToInt32(Console.ReadLine());
        sum = 0;
        for (int i=m; i<=n; i++)
        {
            sum = sum + (Math.Pow(-1, i) / Factorial(i)) * (Math.Pow(((A(i)
        }

        Console.WriteLine(Convert.ToString(sum));
    }
}
```

Випробування алгоритму

Блок	Дія основна програма (цикл 1)		Дія основна програма (цикл 2)	Підпрограма
	Початок			
1	$n = 0 \ m = 1$			
2	$sum = 0$			
3	$i = 0$	$Factorial(i) = 1$	$i = 1$	$Factorial(i) = 1$
4		$A(i) = 1$		$A(i) = 2,449$
5	$sum = 1$		$sum = -0,4832$	
6			виведення(-0,4832)	
7			кінець	

Висновки

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