Додаток 1

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 № 3 з дисципліни
«Алгоритми та структури даних-1.
Основи алгоритмізації»

«Дослідження ітераційних циклічних алгоритмів»

Варіант 29

Виконав студент ІП-15 Рибалка Ілля Сергійович
(шифр, прізвище, ім'я, по батькові)

Перевірив ________________________________
(прізвище, ім'я, по батькові)

Київ 2021

Лабораторна робота 3

Дослідження ітераційних циклічних алгоритмів

Мета – дослідити подання операторів повторення дій та набути практичних навичок їх використання під час складання циклічних програмних специфікацій.

Індивідуальне завдання

Варіант 29

Наближено (із заданою точністю ε) обчислити інтеграл $\int\limits_{0}^{\pi} Ln(2 + Sinx)dx$, використовуючи формулу прямокутників:

$\int\limits_{a}^{b} f(x)dx \approx h \cdot (f(x_1) + f(x_2) +...+ f(x_n))$, де h = (b - a) / n, $x_i = a + i \cdot h - h/2$.

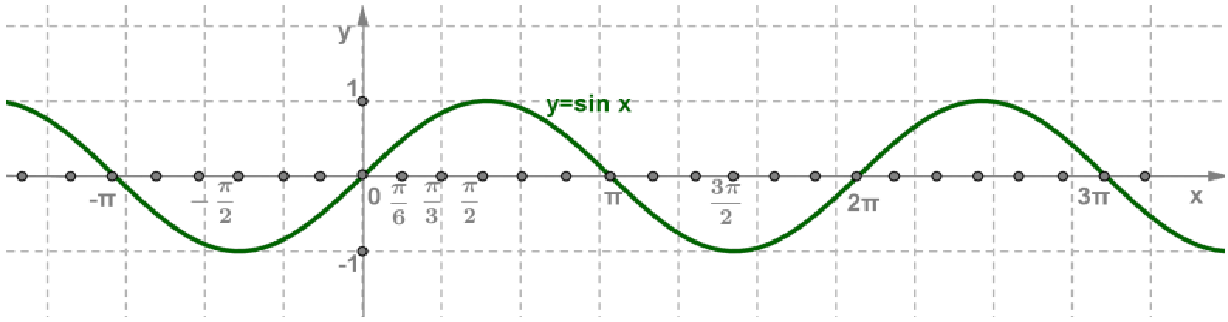
1. Постановка задачі

Обчислити інтеграл з заданою точністю (ε та n) за допомогою ітераційного циклу, що виконує дію суми.

2. Побудова математичної моделі

Змінна	**Тип**	**Ім'я**	**Призначення**
Точність ε	Дійсне	eps	Вхідні дані
Змінна n	Натуральне	n	Проміжні дані
Змінна h	Дійсне	h	Проміжні дані
Лічильник	Натуральне	i	Проміжні дані
Змінна x	Дійсне	x	Проміжні дані
Функція f(x)	Дійсне	fx	Проміжні дані
Сума функцій $f(x_i)$	Дійсне	sumfx	Проміжні дані
Результат минулої ітерації	Дійсне	res1	Проміжні дані
Результат	Дійсне	res	Проміжні, Вихідні дані

Інтеграл задано на проміжку $[0; \pi]$, через це, в програмі буде використано число $\pi(pi) \approx 3,14159...$. За допомогою n вираховується змінна $h = \frac{\pi - 0}{n} = \frac{\pi}{n}$, та відповідно $x_i = 0 + i \cdot h - \frac{h}{2} = i \cdot h - \frac{h}{2}$, x підставляється в функцію $f(x) = Ln(2 + Sinx)$. В функції використовується логарифм натуральний (*log()*), який в основі має число $e \approx 2,71828...$, також в функції використано синус числа *x* (*sin(x)*), що відповідає точці ординат на графіку синуса.



Результатом є змінна *res* що знаходиться за формулою $res = h \cdot \sum_{i=1}^{n} f(x_i)$, суму в коді замінено на цикл де лічильник $i \le n$, результатом цієї дії є проміжна змінна *sumfx*. Точність ε вводиться з клавіатури, для отримання результату з заданою точністю в коді буде використано модуль (*abs()*).

Розв’язання

Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді блок-схеми.
Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо дію введення n, i, res.

Крок 3. Деталізуємо крок знаходження res за рахунок циклу.

Крок 4. Деталізуємо тіло циклу.

Крок 5. Деталізуємо дію знаходження sumfx за рахунок циклу.

Крок 6. Деталізуємо тіло циклу.

Псевдокод

крок 1
початок
Введення eps
<u>*Введення* n, i, res</u>
Знаходження res з заданою точністю
Виведення res
кінець

крок 2
початок
Введення eps
n = 1
i = 1
res = 0
<u>*Знаходження* res *з заданою точністю*</u>
Виведення res
кінець

крок 3
початок
Введення eps
n = 1
i = 1
res = 0
повторити
    Введення res1, sumfx
    Розрахунок h
    Знаходження sumfx
    Розрахунок res
    i = 0
    n++
поки abs(res - res1) > eps
все повторити
Виведення res
кінець

крок 4
початок
Введення eps
n = 1
i = 1
res = 0
повторити
    res1 = res
    sumfx = 0
    h = pi / n
    Знаходження sumfx
    res = h * sumfx
    i = 0
    n++
поки abs(res - res1) > eps
все повторити
Виведення res
кінець

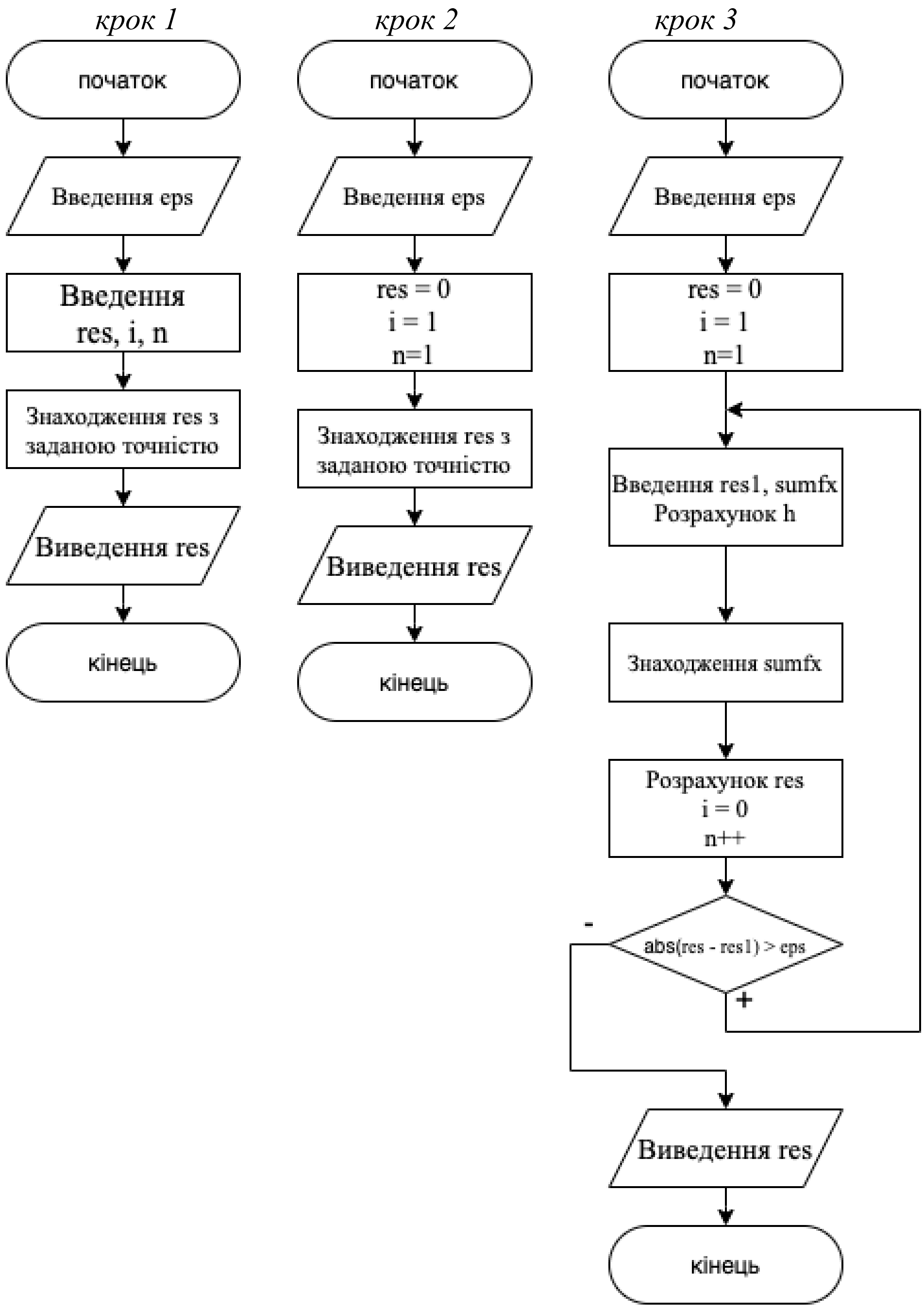
крок 5

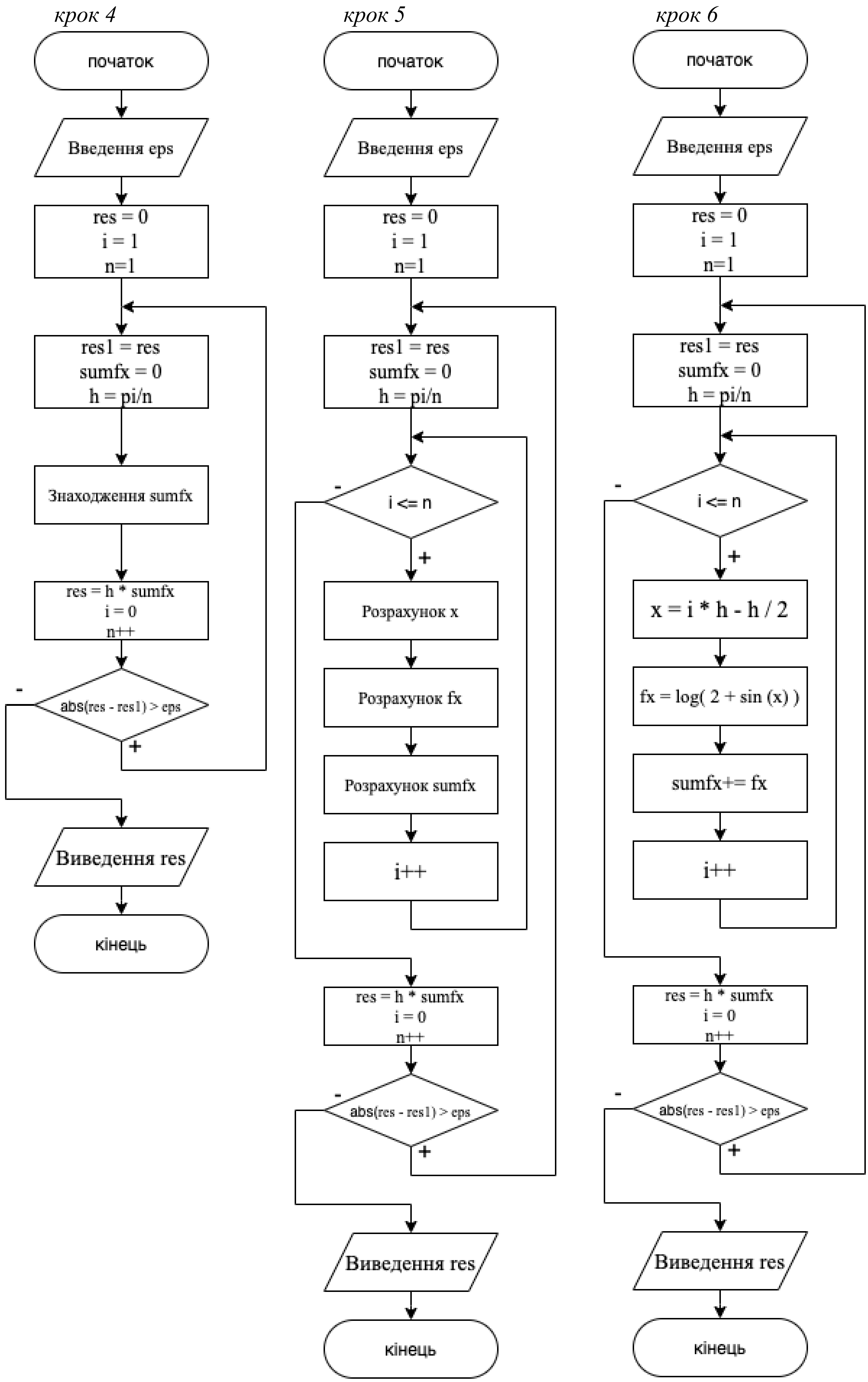
```
початок
Введення eps
n = 1
i = 1
res = 0
повторити
        res1 = res
        sumfx = 0
        h = pi / n
        поки i <= n повторити
                Знаходження x
                Знаходження fx
                Знаходження sumfx
                i++
        все повторити
        res = h * sumfx
        i = 0
        n++
поки abs(res - res1) > eps
все повторити
Виведення res
кінець
```

крок 6

```
початок
Введення eps
n = 1
i = 1
res = 0
повторити
        res1 = res
        sumfx = 0
        h = pi / n
        поки i <= n повторити
                x = i * h - h / 2
                fx = log( 2 + sin(x) )
                sumfx += fx
                i++
        все повторити
        res = h * sumfx
        i = 0
        n++
поки abs(res - res1) > eps
все повторити
Виведення res
кінець
```

Блок-Схема


крок 1
початок
Введення eps
Введення res, i, n
Знаходження res з заданою точністю
Виведення res
кінець
крок 2
початок
Введення eps
res = 0
i = 1
n=1
Знаходження res з заданою точністю
Виведення res
кінець
крок 3
початок
Введення eps
res = 0
i = 1
n=1
Введення res1, sumfx
Розрахунок h
Знаходження sumfx
Розрахунок res
i = 0
n++
abs(res - res1) > eps
-
+
Виведення res
кінець


крок 4
початок
Введення eps
res = 0
i = 1
n=1
res1 = res
sumfx = 0
h = pi/n
Знаходження sumfx
res = h * sumfx
i = 0
n++
abs(res - res1) > eps
-
+
Виведення res
кінець
крок 5
початок
Введення eps
res = 0
i = 1
n=1
res1 = res
sumfx = 0
h = pi/n
i <= n
-
+
Розрахунок x
Розрахунок fx
Розрахунок sumfx
i++
res = h * sumfx
i = 0
n++
abs(res - res1) > eps
-
+
Виведення res
кінець
крок 6
початок
Введення eps
res = 0
i = 1
n=1
res1 = res
sumfx = 0
h = pi/n
i <= n
-
+
x = i * h - h / 2
fx = log(2 + sin (x))
sumfx+= fx
i++
res = h * sumfx
i = 0
n++
abs(res - res1) > eps
-
+
Виведення res
кінець

Тестування

Блок	Дія
	Початок
1	ε = 0.1
2	res = 0, n = 1, i = 1
3.1	n = 1, res1 = 0, i(max) = 2, h = 3.14159, sumfx = 1.09861, res = 3.45139 3.45139 > 0.1 True
3.2	n = 2, res1 = 3.45139, i(max) = 3, h = 2.24864, sumfx = 3.45139, res = 3.53216 0.08077 > 0.1 False
4	Виведення 3.53216
	Кінець

Висновок

Я дослідив подання операторів повторення дій та набув практичних навичок їх використання під час складання циклічних програмних специфікацій. Було створено алгоритм для розв’язання заданого інтегралу з введеними точностями. Цей алгоритм було протестовано на точності обчислень 0.1, результатом слугувало число 3.53216.