

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 5

Виконав студент

ІП-15, Буяло Дмитро Олександрович
(шифр, прізвище, ім'я, по батькові)

Перевірів

(прізвище, ім'я, по батькові)

Київ 2021

Лабораторна робота 3

Дослідження алгоритмів розгалудження

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

Індивідуальне завдання

Варіант 5

Завдання

Обчислити відрізок ряду:

$$S = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n} + 1}{2^n + 1} \quad \text{для } x = 0,56$$

Обчислення завершити, якщо

$$\left| \frac{x^{2n} + 1}{2^n + 1} \right| \leq 10^{-4}$$

1. Постановка задачі

1) Термінологія в формуванні задачі повністю зрозуміла та не потребує пояснень. 2) Маємо нескінченний ряд, значення x та умову, при якій маємо завершити роботу. 3) Необхідно обчислити суму заданого ряду при $x=0,56$, використовуючи вхідні дані та обмеження. 4) Як загальну властивість можна виділити те, що через раз у виразі буде змінюватися знак. 5) Існує багато розв'язків даної задачі, будемо використовувати, на мою думку, найпростіший, найефективніший та найшвидший з них. 6) Даних цілком достатньо та всі потрібні. 7) Зробимо пропущення, що n – множина натуральних чисел.

2. Побудова математичної моделі

Складемо таблицю імен змінних

<i>Змінна</i>	<i>Тип</i>	<i>Ім'я</i>	<i>Призначення</i>
Змінна x	Дійсне	x	Вхідне дане
Натуральні числа	Ціле	n	Вхідне дане
Змінна k	Дійсне	k	Проміжні дані
Елемент ряду	Дійсне	s1	Проміжні дані
Сума	Дійсне	S	Результат

Змінна x нам задана за умовою. Щоб знайти суму S, нам потрібно створити цикл, який буде працювати, поки не виконається умова $\left| \frac{x^{2n+1}}{2^{n+1}} \right| \leq 10^{-4}$, що ми замінили на $k \leq 10^{-4}$, також ця умова зазначає, що цикл не нескінченний. Для реалізації введемо змінну s1, яка буде оновлюватися при кожному збільшенні n, і буде рівна наступному елементу ряду суми. Тобто, щоб знайти суму S, ми маємо скласти всі отримані s1.

У роботі потрібно використовувати модуль та піднесення до степеня. Для позначення модуля у псевдокодi будемо використовувати функцію Abs(), а для блок-схеми позначення «| |», щоб економити місце. Abs/|| повертає модуль (абсолютну величину, Abs) числа. Абсолютна величина числа - це число без знака. Для позначення степеня у псевдокодi будемо використовувати функцію Pow(), а для блок-схеми позначення «^», щоб економити місце. Pow/^ повертає значення числа, піднесеного до степеня. Також зазначимо, що оператор += - це оператор складання з привласненням, додає значення правого операнда до змінної і привласнює змінній результат.

Розв'язання

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

Крок 1. Визначимо основні дії

Крок 2. Деталізуємо дію знаходження суми S .

3. Псевдокод алгоритму

Крок 1

Крок 2

Початок

Початок

Задання x, n

Задання x, n

Обчислення S

$k = \text{Abs}((\text{Pow}(x, 2) + 1) / 3)$

Повторити

Поки $k > \text{Pow}(10, -4)$

$s1 = \text{Pow}(-1, n) * (\text{Pow}(x, 2 * n) + 1) / (\text{Pow}(2, n) + 1)$

$S += s1$

$n = n + 1$

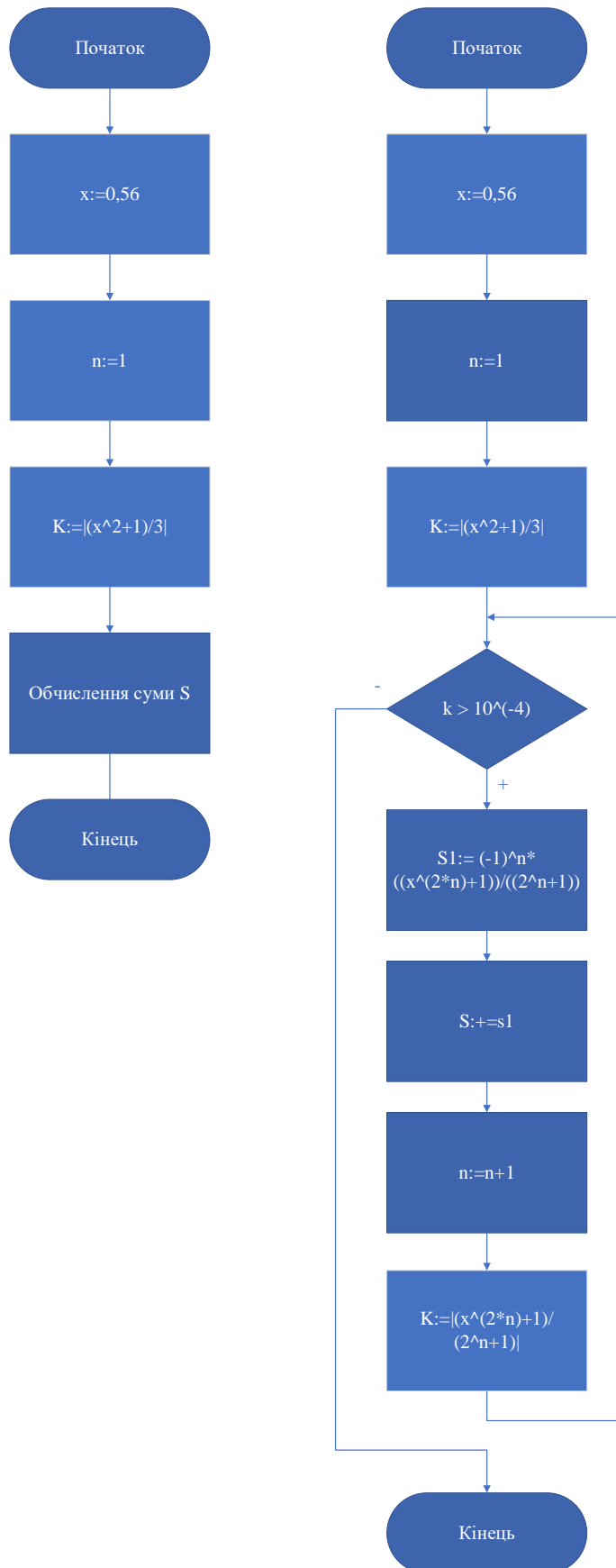
$k = \text{Abs}((\text{Pow}(x, 2 * n) + 1) / (\text{Pow}(2, n) + 1))$

Все повторити

Кінець

Кінець

4. Блок-схема алгоритму



5. Випробування алгоритму

Наведемо приклад виконання алгоритму.

Блок	Дія
	Початок
1	Задання $x=0.56$, $n=1$, $k=0.437867$
2	$k>0.0001$
	$s1=-0.437867$
	$S=-0.437867$
	$n=1+1$
	$k=0.219668992$
3	$k>0.0001$
	$s1=0.219668992$
	$S=-0.21819767467$
	$n=2+1$
	$k=0.11453788660622223$
4-13	...
14	$k>0.0001$
	$s1=-1.2205544778703576E-4$
	$S=-0.29354140930145156$
	$k = 6.103143661729241E-5$
	$k<0.0001$
	Кінець

Випробування алгоритм пройшов відмінно, видавши правильний кінцевий результат.

6. Висновки

Ми дослідили подання операторів повторення дій та набути практичних навичок їх використання під час складання циклічних програмних специфікацій. В результаті виконання лабораторної роботи ми отримали алгоритм для знаходження суми S за заданою формулою. Дискретували задачу на 2 кроки: визначили основні дії, потім деталізували визначення суми S . Алгоритм є ефективним та результативним, бо забезпечує розв'язок за мінімальний час із мінімальними витратами ресурсів та отримує чіткий кінцевий результат.