

Міністерство освіти і науки України  
Національний технічний університет України «Київський політехнічний  
інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Катедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 3 з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«Дослідження ітераційних циклічних алгоритмів»

Варіант 18

Виконав студент ІП-11 Лесів Владислав Ігорович

Перевірив                      Мартинова О.П.

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.

#### Варіант №18.

18. Задане дійсне число  $x$ . Послідовність  $a_1, a_2, \dots, a_n$  утворена за законом

$$a_n = x^n / (2n)!, \quad n = 1, 2, \dots$$

Отримати суму  $a_1 + a_2 + \dots + a_k$ , де  $k$  - найменше ціле число, що задовольняє двом умовам:  $k > 10, |a_k| < 10^{-5}$ .

**Постановка задачі.** Результатом розв'язку є сума елементів заданої законом послідовності за виконання необхідним умов задачі, та число-умова виконання критеріїв. Для визначення результату повинне бути задане дійсне число  $x$ . Інших початкових даних для розв'язку не потрібно.

**Побудова математичної моделі.** Складемо таблицю імен змінних.

Змінна	Тип	Ім'я	Призначення
Задане дійсне число $x$	Дійсний	$x$	Початкове дане
Елемент послідовності	Дійсний	$a$	Проміжне значення
Значення факторіалу числа	Цілий	$fac$	Проміжне значення
Сума елементів послідовності	Дійсний	$s$	Результат
Число, яке задовільняє умови задачі	Цілий	$k$	Результат

Математичне формулювання задачі зводиться до перевірки виконання заданих умов. Якщо модуль значення елемента послідовності на кроці  $k$  більший або рівний за  $10^{-5}$ , тобто  $|a| < 10^{-5}$ , а саме число  $k > 10$ , отримуємо виконання умов, що спричинить завершення перевірки. У іншому випадку одна з умов не виконується, тому додаємо елемент до суми елементів та продовжуємо перевірку в циклі.

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

*Крок 1.* Визначимо основні дії.

*Крок 2.* Деталізуємо дію обчислення початкового значення суми елементів та першого члена послідовності.

*Крок 3.* Деталізуємо дію обчислення значення суми елементів послідовності за виконання заданих умов.

*Псевдокод*

*крок 1*

**початок**

введення  $x$

обчислення початкового значення суми та першого члена

обчислення значення суми елементів послідовності за виконання заданих умов

виведення  $s, k$

**кінець**

*крок 2*

**початок**

введення  $x$

$s := x/2;$

$a := x/2;$

обчислення значення суми елементів послідовності за виконання заданих умов

виведення  $s, k$

**кінець**

*крок 3*

**початок**

введення  $x$

$s := x/2;$

$a := x/2;$

$k := 1;$

**повторити**

**поки**  $|a| \geq 10^{-5}$  **або**  $k \leq 10$

$k := k + 1;$

$fac := 1;$

**повторити**

**для**  $i$  **від** 1 **до**  $2k$

$fac := fac * i;$

**все повторити**

$a := (x^k) / fac;$

$s := s + a;$

**все повторити**

вивести  $s, k$

**кінець**

## Блок-схема

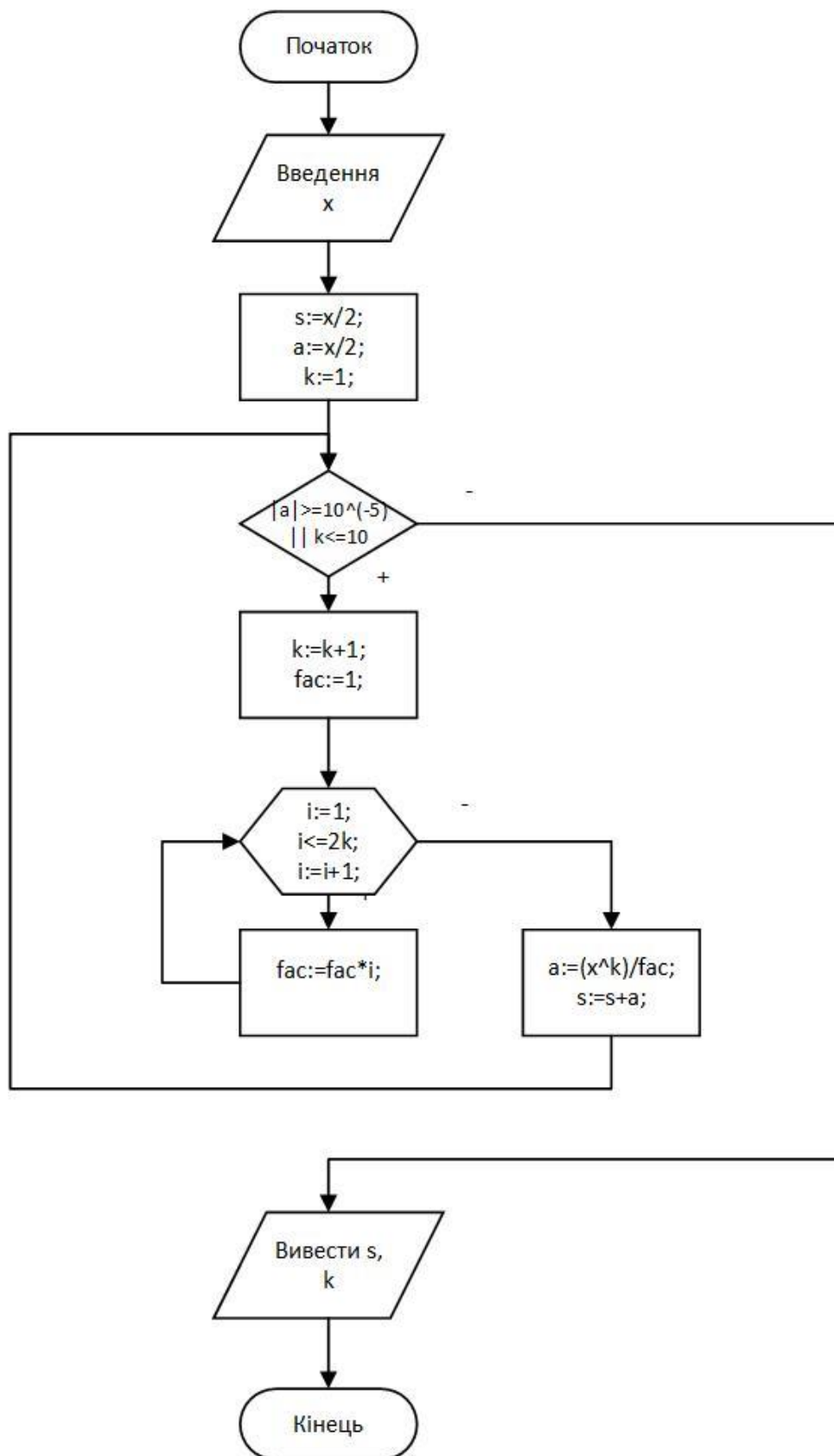
### крок 1



### крок 2



крок 3



**Випробування алгоритму.** Перевіримо правильність алгоритму на довільних конкретних значеннях початкових даних.  $n$  – загальна к-сть блоків.

Блок	Дія
	Початок
1	Введення $x=2$
2	$s:=1; a:=1; k:=1;$
3	$ a  \geq 10^{-5} \parallel k \leq 10$ - істина
4	$k:=2; fac:=1; i:=1;$
5	$i \leq 2*k - 1 \leq 4$ - істина
6	$fac:=1*i; i:=2;$
7	$i \leq 2*k - 2 \leq 4$ - істина
8	$fac:=1*i; i:=3;$
9-11	...
12	$fac:=6*i; i:=5;$
13	$i \leq 2*k - 5 \leq 4$ - хиба
14	$a:=(2^2)/24=1/6; s:=1+1/6;$
15	$ a  \geq 10^{-5} \parallel k \leq 10$ - істина
16	$k:=3; fac:=1; i:=1;$
...	...
$n-2$	$a:=1.82e-18;$ $s:=1.178183556608571;$
$n-1$	$ a  \geq 10^{-5} \parallel k \leq 10$ – хиба
$n$	Виведення $s=1.178183556608571,$ $k=11$
	Кінець

**Висновок.** Отже, у цій роботі я дослідив подання операторів повторення дій та набув практичних навичок їх використання під час складання циклічних програмних специфікацій. У результаті лабораторної роботи було розроблено математичну модель, що відповідає постановці задачі; псевдокод та блок-схеми, які пояснюють логіку алгоритму. Використовуючи ітераційний цикл з передумовою `while` для повторюваної перевірки умов, отримуємо коректний результат.