

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

3bit

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

## «Дослідження ітераційних циклічних алгоритмів»

Вариант 18

Виконав студент ІП-12 Кушнір Ганна Вікторівна  
(шифр, прізвище, ім'я, по батькові)

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

Київ 2021

### Лабораторна робота 3

#### Дослідження ітераційних циклічних алгоритмів

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

#### Варіант 18

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

$$a_n = \frac{x^n}{(2n)!}, n = 1, 2, \dots$$

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.** Початковим даним є дійсне число  $x$ , яке вводиться користувачем з клавіатури. Результатом розв'язку є дійсне число Sum – сума  $n$  членів послідовності, заданих формулою  $n$ -го члена  $a_n$ .
- Побудова математичної моделі.** Складемо таблицю імен змінних.

| Змінна                        | Тип     | Ім'я | Призначення       |
|-------------------------------|---------|------|-------------------|
| Дійсне число $x$              | Дійсний | $x$  | Початкове дане    |
| Член послідовності            | Дійсний | $a$  | Проміжна величина |
| Номер члену послідовності     | Цілий   | $n$  | Лічильник         |
| Сума $k$ членів послідовності | Дійсний | Sum  | Результат         |

Таким чином, математичне формулювання задачі зводиться до задання початкових значень змінних:  $n:=1, a:=x/2, \text{Sum}:=a$ , а далі циклічне повторення виконання дій:  $n:=n+1, a:=(a*x)/(2*n*(2*n-1)), \text{Sum}:=\text{Sum}+a$ , поки не буде виконано дві умови:  $n > 10$  та  $|a_n| < 0.00001$ . Але оскільки перевірку виконання обох цих умов важко задати в одному циклі, необхідно розбити алгоритм на два цикли з повторенням тих самих дій, але за різних умов. Для розв'язання цієї задачі буде використано цикл з передумовою (основну схему).

Додаткові функції: для перевірки другої умови буде використано функцію  $\text{abs}()$ , яка повертає модуль числа.

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

**Крок 1.** Визначимо основні дії.

*Крок 2.* Деталізуємо дію присвоєння початкових значень змінним.

*Крок 3.* Деталізуємо дію обчислення суми перших 11 членів послідовності (для виконання умови  $k > 10$ ).

*Крок 4.* Деталізуємо дію обчислення суми для всіх членів послідовності від  $a_1$  до  $a_n$ , де  $|a_n| < 10^{-5}$ .

### *3. Псевдокод алгоритму.*

#### *Крок 1*

**початок**

введення  $x$

присвоєння початкових значень змінним  $n$ ,  $a$  та  $Sum$

обчислення суми  $Sum$  перших 11 членів послідовності

обчислення суми  $Sum$  всіх членів послідовності від  $a_1$  до  $a_n$

виведення  $Sum$

**кінець**

#### *Крок 2*

**початок**

введення  $x$

$n := 1$

$a := x / 2$

$Sum := a$

обчислення суми  $Sum$  перших 11 членів послідовності

обчислення суми  $Sum$  всіх членів послідовності від  $a_1$  до  $a_n$

виведення  $Sum$

**кінець**

#### *Крок 3*

**початок**

введення  $x$

$n := 1$

$a := x / 2$

$Sum := a$

**повторити**

**поки**  $n < 11$

$n := n + 1$

$a := (a * x) / (2 * n * (2 * n - 1))$

$Sum := Sum + a$

**все повторити**

обчислення суми  $Sum$  всіх членів послідовності від  $a_1$  до  $a_n$

виведення  $Sum$

**кінець**

#### *Крок 4*

**початок**

введення  $x$

$n := 1$

$a := x / 2$

$Sum := a$

**повторити**

**поки**  $n < 11$

$n := n + 1$

$a := (a * x) / (2 * n * (2 * n - 1))$

$Sum := Sum + a$

**все повторити**

**повторити**

**поки**  $abs(a) \geq 0.00001$

$n := n + 1$

$a := (a * x) / (2 * n * (2 * n - 1))$

$Sum := Sum + a$

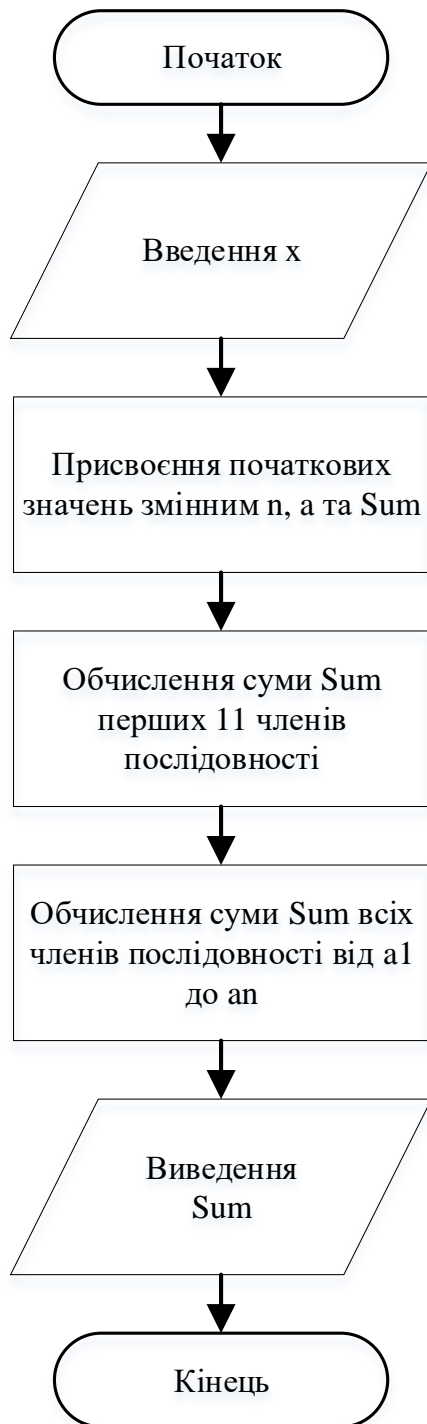
**все повторити**

виведення  $Sum$

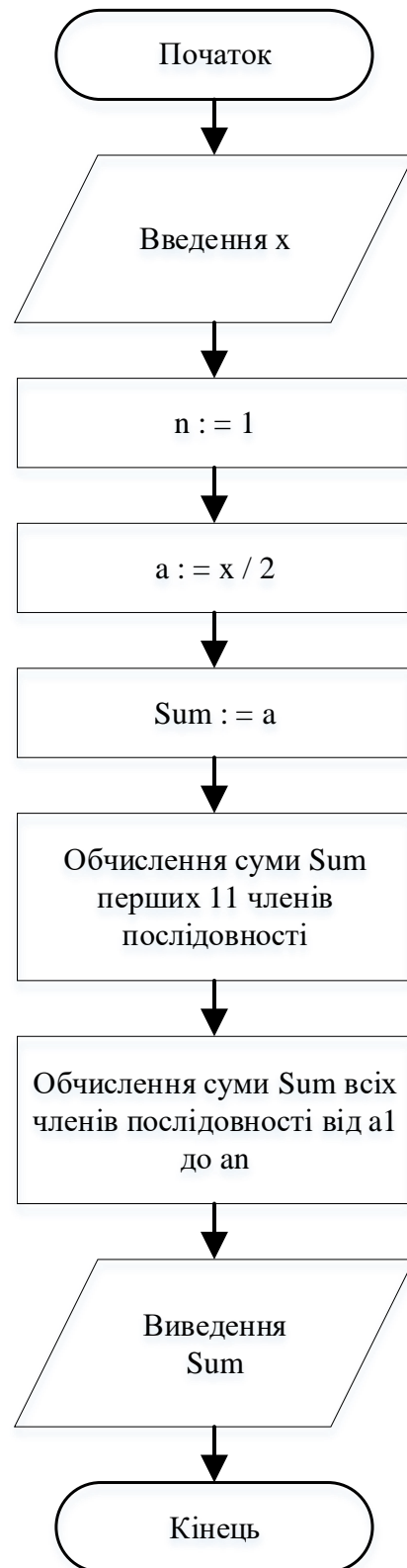
**кінець**

#### 4. Блок-схема алгоритму.

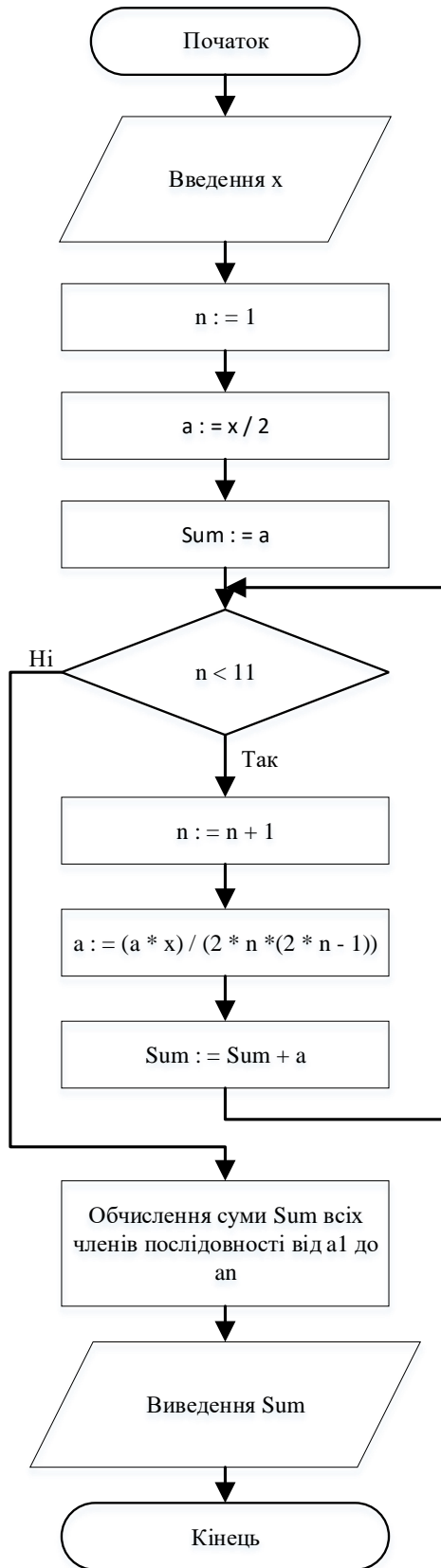
Крок 1



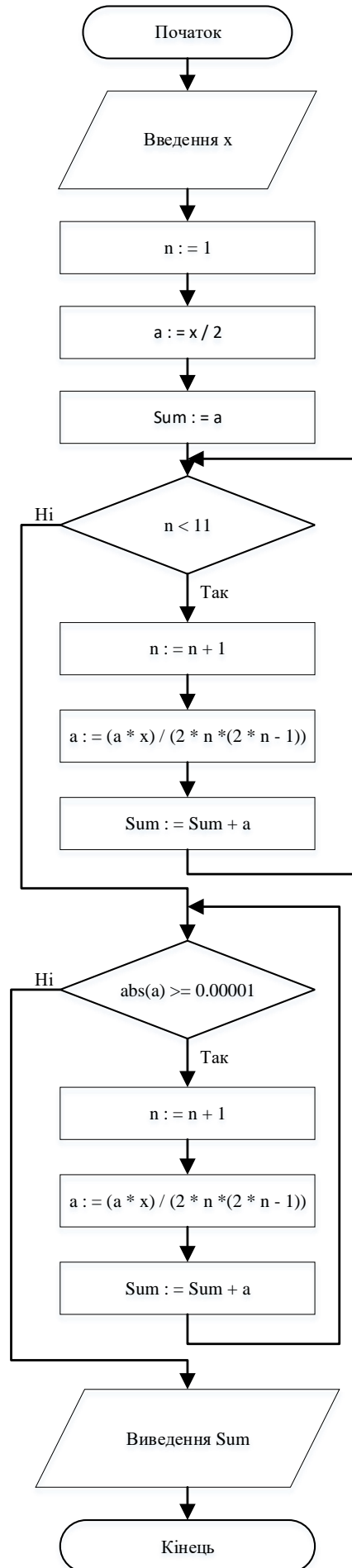
Крок 2



### Крок 3



### Крок 4



5. *Випробування алгоритму.* Перевіримо правильність алгоритму на довільних конкретних значеннях початкових даних:

| Блок       | Дія  |
|------------|--|
|            | Початок  |
| <b>1</b>   | Введення $x = 11$  |
| <b>2</b>   | $n = 1$  |
| <b>3</b>   | $a = 11/2 = 5.5$   |
| <b>4</b>   | $Sum = 5.5$  |
| <b>5.1</b> | $n < 11$ – так<br>Тоді<br>$n = 1+1 = 2$<br>$a = (5.5*11)/(2*2*(2*2-1)) = 5.04167$<br>$Sum = 10.54167$              |
| <b>5.2</b> | $n < 11$ – так<br>Тоді<br>$n = 2+1 = 3$<br>$a = (5.04167*11)/(2*3*(2*3-1)) = 1.84861$<br>$Sum = 12.390278$         |
| <b>5.3</b> | $n < 11$ – так<br>Тоді<br>$n = 3+1 = 4$<br>$a = (1.84861*11)/(2*4*(2*4-1)) = 0.36312$<br>$Sum = 12.753398$         |
| <b>5.4</b> | $n < 11$ – так<br>Тоді<br>$n = 4+1 = 5$<br>$a = (0.36312*11)/(2*5*(2*5-1)) = 0.04438$<br>$Sum = 12.79778$          |
| <b>5.5</b> | $n < 11$ – так<br>Тоді<br>$n = 5+1 = 6$<br>$a = (0.04438*11)/(2*6*(2*6-1)) = 0.003698$<br>$Sum = 12.801476$        |
| <b>5.6</b> | $n < 11$ – так<br>Тоді<br>$n = 6+1 = 7$<br>$a = (0.003698*11)/(2*7*(2*7-1)) = 0.0002235$<br>$Sum = 12.8016997$     |
| <b>5.7</b> | $n < 11$ – так<br>Тоді<br>$n = 7+1 = 8$<br>$a = (0.0002235*11)/(2*8*(2*8-1)) = 0.0000102452$<br>$Sum = 12.8017099$ |
| <b>5.8</b> | $n < 11$ – так<br>Тоді<br>$n = 8+1 = 9$<br>$a = (0.0000102*11)/(2*9*(2*9-1)) = 3.68291503E-7$                      |

|             |  |
|-------------|--|
|             | Sum = 12.80171029  |
| <b>5.9</b>  | n < 11 – так<br>Тоді<br>n = 9+1 = 10<br>a = (3.68291503E-7*11)/(2*10*(2*10-1)) = 1.06610526E-8<br>Sum = 12.8017103   |
| <b>5.10</b> | n < 11 – так<br>Тоді<br>n = 10+1 = 11<br>a = (1.06610526E-8*11)/(2*11*(2*11-1)) = 2.53834587E-10<br>Sum = 12.8017103 |
| <b>5.11</b> | n < 11 – ні  |
| <b>6</b>    | abs(a)=abs(2.53834587E-10)>=0.00001 – ні   |
| <b>7</b>    | Виведення Sum = 12.80171   |
|             | Кінець   |

6. *Висновки.* На цій лабораторній роботі було досліджено подання операторів повторення дій та було набуто практичних навичок їх використання під час складання циклічних програмних специфікацій.