

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

«Дослідження ітераційних циклічних  
алгоритмів»

Варіант 6

Виконав студент ІП-13 Вдовиченко Станіслав Юрійович  
(шифр, прізвище, ім'я, по батькові)

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

Київ 20211

### 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.

- **Постановка задачі (Варіант 6)**

6. З точністю  $\varepsilon = 10^{-5}$  обчислити значення функції  $e^x$ :

$$e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots + \frac{x^n}{n!} + \dots$$

Задаємо значення епсилон (точність), задаємо значення змінної  $x$  і значення ітераційної змінної (початкове значення = 1).

Запишемо рекурентне відношення для кожного наступного члена і знайдемо значення функції за допомогою ітераційного циклу(до заданої точності).

- **Математична модель**

Змінна	Тип	Ім'я	Призначення
Епсилон	Дійсний	eps	Вхідні дані
Аргумент	Дійсний	x	Вхідні дані
Попереднє значення компоненту послідовності	Дійсний	previousComponent	Проміжний результат
Поточний компонент послідовності	Дійсний	currentComponent	Проміжний результат
Різниця поточного і попереднього значення	Дійсний	diff	Проміжний результат
Ітераційна змінна	Цілий	n	Ітераційна змінна
Функція $e^x$	Дійсний	functE	Вихідні дані

Розв'язання.

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

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо дію обчислення першого члена

Крок 3. Деталізуємо дію обчислення n-го члена послідовності.

n-ий член послідовності знайдемо за формулою:

$$x_n = x_{n-1} * \frac{x}{n}$$

Тоді до попереднього значення функції будемо додавати n-ий член, поки різниця(diff) не буде менша за точність(eps).

Різницю (diff) знайдемо за формулою  $x_n - x_{n-1}$ .

Щоб знайти модуль різниці використовуємо функцію abs().

Факторіал розписуємо у формулі n-го члена (добуток попереднього члена на наступний), тобто:

$$\frac{x^3}{3!} = \frac{x^2}{2} * \frac{x}{3}$$

Початкове значення функції задаємо 1, номер ітерації – 1, значення початкового члена послідовності – 1.

- **Псевдокод**

**Крок 1.****Початок**

Введення даних

Декларування змінних

Обчислення значення функції за заданою точністю

Виведення даних

**Кінець****Крок 2.****Початок**

Введення  $x$

$\text{eps} := 0.00001, n := 1, \text{functE} := 1, \text{previousComponent} := 1, \text{diff} := 1.$

Обчислення значення функції за заданою точністю

Виведення даних

**Кінець****Крок 3.****Початок**

Введення  $x$

$\text{eps} := 0.00001, n := 1, \text{functE} := 1, \text{previousComponent} := 1, \text{diff} := 1.$

**поки** ( $\text{diff} \geq \text{eps}$ )

Знаходження  $n$ -го члена

Знаходження нового значення функції

Знаходження різниці членів

Задання значень для продовження циклу

**все поки**

Виведення даних

**Кінець**

#### **Крок 4.**

##### **Початок**

Введення  $x$

$\text{eps} := 0.00001, n := 1, \text{functE} := 1, \text{previousComponent} := 1, \text{diff} := 1.$

**поки** ( $\text{diff} \geq \text{eps}$ )

$\text{currentComponent} := \text{previousComponent} * (x / n);$

Знаходження нового значення функції

Знаходження різниці членів

Задання значень для продовження циклу

**все поки**

Виведення даних

**Кінець**

#### **Крок 5.**

##### **Початок**

Введення  $x$

$\text{eps} := 0.00001, n := 1, \text{functE} := 1, \text{previousComponent} := 1, \text{diff} := 1.$

**поки** ( $\text{diff} \geq \text{eps}$ )

$\text{currentComponent} := \text{previousComponent} * (x / n);$

$\text{functE} := \text{functE} + \text{currentComponent};$

Знаходження різниці членів

Задання значень для продовження циклу

**все поки**

Виведення даних

**Кінець**

#### **Крок 6.**

##### **Початок**

Введення  $x$

$\text{eps} := 0.00001, n := 1, \text{functE} := 1, \text{previousComponent} := 1, \text{diff} := 1.$

**поки** ( $\text{diff} \geq \text{eps}$ )

$\text{currentComponent} := \text{previousComponent} * (x / n);$

$\text{functE} := \text{functE} + \text{currentComponent};$

$\text{diff} := \text{abs}(\text{currentComponent} - \text{previousComponent});$

Задання значень для продовження циклу

**все поки**

Виведення даних

**Кінець**

## **Крок 7.**

### **Початок**

Введення  $x$

$\text{eps} := 0.00001$ ,  $n := 1$ ,  $\text{functE} := 1$ ,  $\text{previousComponent} := 1$ ,  $\text{diff} := 1$ .

**поки** ( $\text{diff} \geq \text{eps}$ )

$\text{currentComponent} := \text{previousComponent} * (x / n)$  ;

$\text{functE} := \text{functE} + \text{currentComponent}$ ;

$\text{diff} := \text{abs}(\text{currentComponent} - \text{previousComponent})$ ;

$\text{previousComponent} := \text{currentComponent}$ ;

$n++$ ;

**все поки**

Виведення даних

**Кінець**

## **Крок 8.**

### **Початок**

Введення  $x$

$\text{eps} := 0.00001$ ,  $n := 1$ ,  $\text{functE} := 1$ ,  $\text{previousComponent} := 1$ ,  $\text{diff} := 1$ .

**поки** ( $\text{diff} \geq \text{eps}$ )

$\text{currentComponent} := \text{previousComponent} * (x / n)$  ;

$\text{functE} := \text{functE} + \text{currentComponent}$ ;

$\text{diff} := \text{abs}(\text{currentComponent} - \text{previousComponent})$ ;

$\text{previousComponent} := \text{currentComponent}$ ;

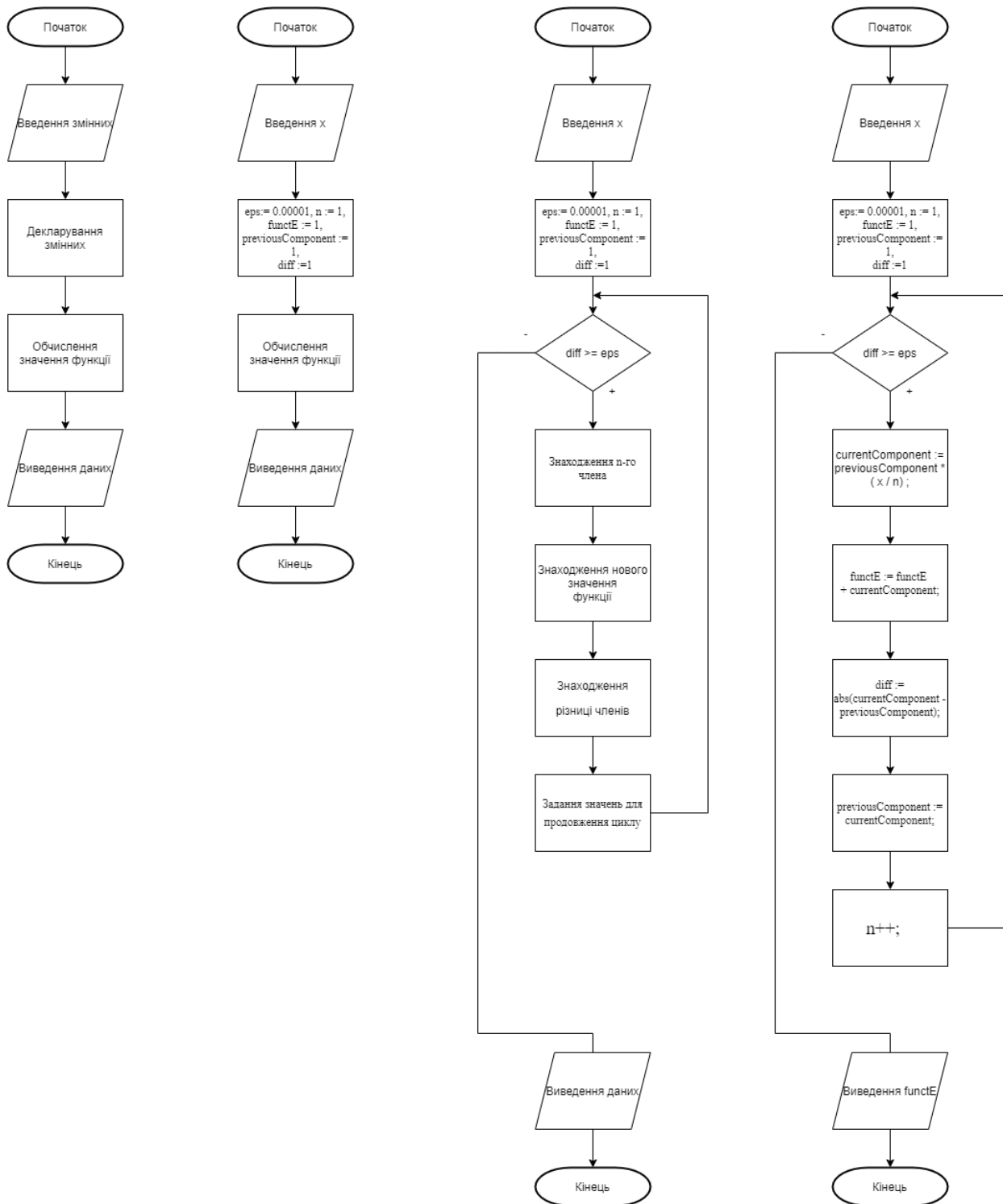
$n++$ ;

**все поки**

Виведення  $\text{functE}$

**Кінець**

## • Блок-схема



### • Випробування алгоритму

Перевіримо правильність алгоритму на довільних конкретних значеннях початкових даних.

Блок	Дія
	Початок
1	Введення $x := 2.25$
2	$\text{eps} := 0.00001, n := 1, \text{functE} := 1, \text{previousComponent} := 1, \text{diff} := 1$
3	$(1 > 0.00001); \text{currentComponent} := 1 * (2.25/1) = 2.25; \text{functE} := 1 + 2.25 = 3.25; \text{diff} := (2.25 - 1) = 1.25 ; \text{previousComponent} := 2.25; n = 2;$
4	$(1.25 > 0.00001); \text{currentComponent} := 2.25 * (2.25/2) = 2.53125; \text{functE} := 3.25 + 2.53 = 5.78125; \text{diff} := (2.53125 - 2.25) = 0.28 ; \text{previousComponent} := 2.53125; n = 3;$
5	$(0.28 > 0.00001); \text{currentComponent} := 2.53125 * (2.25/3) = 1.89 ; \text{functE} := 5.78125 + 1.89 = 7.679; \text{diff} := (1.89 - 2.53125) = 0.63 ; \text{previousComponent} := 1.89; n = 4;$
6	$(0.63 > 0.00001); \text{currentComponent} := 1.89 * (2.25/4) = 1.067 ; \text{functE} := 7.679 + 1.067 = 8.7475; \text{diff} := (1.067 - 1.89) = 0.83 ; \text{previousComponent} := 1.067; n = 5;$
7	$(0.83 > 0.00001); \text{currentComponent} := 1.067 * (2.25/5) = 0.48 ; \text{functE} := 8.7475 + 0.48 = 9.228; \text{diff} := (0.48 - 1.067) = 0.587 ; \text{previousComponent} := 0.48; n = 6;$
8	$(0.587 > 0.00001); \text{currentComponent} := 0.48 * (2.25/6) = 0.18 ; \text{functE} := 9.228 + 0.18 = 9.408; \text{diff} := (0.18 - 0.48) = 0.3 ; \text{previousComponent} := 0.18; n = 7;$
9	$(0.3 > 0.00001); \text{currentComponent} := 0.18 * (2.25/7) = 0.057 ; \text{functE} := 9.408 + 0.057 = 9.4662; \text{diff} := (0.057 - 0.18) = 0.122 ; \text{previousComponent} := 0.057; n = 8;$
10	$(0.122 > 0.00001); \text{currentComponent} := 0.057 * (2.25/8) = 0.016 ; \text{functE} := 9.4662 + 0.016 = 9.4825; \text{diff} := (0.016 - 0.057) = 0.041 ; \text{previousComponent} := 0.016; n = 9;$
11	$(0.041 > 0.00001); \text{currentComponent} := 0.016 * (2.25/9) = 0.004 ; \text{functE} := 9.4825 + 0.004 = 9.48658; \text{diff} := (0.004 - 0.016) = 0.0122 ; \text{previousComponent} := 0.004; n = 10;$
12	$(0.0122 > 0.00001); \text{currentComponent} := 0.004 * (2.25/10) = 0.00091; \text{functE} := 9.48658 + 0.00091 = 9.4875; \text{diff} := (0.00091 - 0.004) = 0.0031 ; \text{previousComponent} := 0.00091; n = 11;$
13	$(0.0031 > 0.00001); \text{currentComponent} := 0.00091 * (2.25/11) = 0.00018; \text{functE} := 9.4875 + 0.00018 = 9.48769; \text{diff} := (0.00018 - 0.00091) = 0.00072 ; \text{previousComponent} := 0.00018; n = 12;$



14	$(0.00072 > 0.00001)$ ; $\text{currentComponent} := 0.00018 * (2.25/12) = 0.000035$ ; $\text{functE} := 9.48769 + 0.000035 = 9.48772$ ; $\text{diff} := (0.000035 - 0.00018) = 0.00015$ ; $\text{previousComponent} := 0.000035$ ; $n = 13$ ;
15	$(0.00015 > 0.00001)$ ; $\text{currentComponent} := 0.000035 * (2.25/13) = 0.000006$ ; $\text{functE} := 9.48772 + 0.000006 = 9.48773$ ; $\text{diff} := (0.000006 - 0.000035) = 0.00002$ ; $\text{previousComponent} := 0.000006$ ; $n = 14$ ;
16	$(0.00002 > 0.00001)$ ; $\text{currentComponent} := 0.000006 * (2.25/14) = 0.0000009$ ; $\text{functE} := 9.48773 + 0.0000009 = 9.48774$ ; $\text{diff} := (0.0000009 - 0.000006) = 0.000005$ ; $\text{previousComponent} := 0.0000009$ ; $n = 15$ ;
17	$(0.000005 < 0.00001)$
18	$\text{functE} := 9.48774$
19	Виведення $\text{functE}$
	Кінець

### • Висновок

Я дослідив подання операторів повторення дій та набув практичних навичок їх використання під час складання циклічних програмних специфікацій. Побудував алгоритм знаходження значення послідовності за допомогою циклу з передумовою, побудував псевдокод та блок-схему. Під час випробування дослідив значення послідовності після кожної ітерації.