## Основи алгоритмізації

Додаток 1

Министерство освіти і науки України
Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації і управління

#### Звіт

з лабораторної роботи №3 з дисципліни «Алгоритми та структури даних-1. Основи алгоритмізації» «Дослідження ітераційних циклічних алгоритмів» Варіант <u>6</u>

Виконав	П1-15, Волинець Кирило Михаилович
студент	(шифр, прізвище, ім'я, по батькові)
Перевірив	
1 1	( прізвище, ім'я, по батькові)

Лабораторна робота 3 Дослідження ітераційних циклічних алгоритмів

### Основи алгоритмізації

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

#### Індивідуальне завдання

#### Варіант 6

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

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

### Побудова математичної моделі.

Результат  $\epsilon$  нескінченною сумою, тому треба додавати елементи поки не буде досягнена потрібна точність. Спочатку суму sum визначимо першим доданком, тобто 1, а потім будемо заміняти sum її сумою з доданком drib. Оскільки drib  $\epsilon$  дробом, то доцільно зробити вирахування чисельника chisel окремо від знаменника znamen. Чисельник залежить від x, який вводиться людиною. Кожен крок циклу chisel треба множити на x, підвищуючи степінь. Оскільки знаменник  $\epsilon$  факторіалом, то нам треба знати на яке число його множити. Цим числом буде змінна factor, яка буде підвищуватись на 1 кожного кроку. За похибку будемо вважати зміну суми, тобто drib.

#### Складемо таблицю імен змінних.

Змінна	Tun	Ім'я	Призначення
Степінь числа Ейлера	3 плаваючою крапкою	x	Ввід, проміжні
Сума	3 плаваючою крапкою	sum	Проміжні, результат
Доданок до суми	3 плаваючою крапкою	drib	Проміжні
Чисельник доданку	3 плаваючою крапкою	chisel	Проміжні
Знаменник доданку	Цілочисельний	znamen	Проміжні
Множник факторіала	Цілочисельний	factor	Проміжні

#### Розв'язання

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

Крок 1. Визначимо стартові змінні

```
Основи алгоритмізації
Крок 2. Обрахуємо чисельник дробу
Крок 3. Обрахуємо знаменник дробу
Крок 4. Знайдемо результат кроку
Псевдокод
Крок 1:
початок
    введення х
    sum = 1
    chisel = 1
    znamen = 1
    factor = 1
    drib = 1
    поки не досягнена потрібна точність повторити:
        Знаходження чисельника
        Знаходження знаменника
        Знаходження результату
    все повторити
    вивести sum
кінець
Крок 2:
початок
    введення х
    sum = 1
    chisel = 1
    znamen = 1
    factor = 1
    drib = 1
    поки не досягнена потрібна точність повторити:
         chisel = chisel * x
        Знаходження знаменника
        Знаходження результату
    все повторити
    вивести sum
кінець
```

```
Основи алгоритмізації
Крок 3:
початок
    введення х
    sum = 1
    chisel = 1
    znamen = 1
    factor = 1
    drib = 1
    поки не досягнена потрібна точність повторити:
        chisel = chisel * x
        znamen = znamen * factor
        factor = factor + 1
        Знаходження результату
    все повторити
    вивести sum
кінець
Крок 4:
початок
    введення х
    sum = 1
    chisel = 1
    znamen = 1
    factor = 1
    drib = 1
    поки drib >= 0.00001 повторити:
        chisel = chisel * x
        znamen = znamen * factor
        factor = factor + 1
        drib = chisel/znamen
        sum = sum + drib
    все повторити
    вивести sum
кінець
```

## Основи алгоритмізації

Блок-схема

#### Крок 2: Крок 1: Крок 3: Крок 4: початок початок початок початок Введення х Введення х Введення х Введення х sum = 1 sum = 1 sum = 1sum = 1 drib = 1 drib = 1 drib = 1 drib = 1 chisel = 1 chisel = 1 chisel = 1 chisel = 1 znamen = 1 znamen = 1 znamen = 1znamen = 1 factor = 1 factor = 1 factor = 1 factor = 1 Потрібна точність Потрібна точність Потрібна точність не досягнена не досягнена не досягнена drib >=0.00001 Знахолження chisel = chisel \* x chisel = chisel \* x чисельника chisel = chisel \* x Знаходження Знаходження znamen = znamen \* znamen = znamen \* знаменника знаменника factor factor Знаходження Знаходження результату результату factor = factor + 1 factor = factor + 1 Знаходження drib = chisel / результату znamen Вивести sum Вивести sum sum = sum + dribкінець кінець Вивести sum кінець Вивести sum

кінець

# Основи алгоритмізації

## Перевірка

	Випадок 1	Випадок 2	Випадок 3	Випадок 4	Випадок 5
1	початок	початок	початок	початок	початок
2	Введення х = -1	Введення	Введення х = 0.5	Введення х = 1	Введення х = 2.5
		$\mathbf{x} = 0$			
9	chisel = $1 * (-1) = -1$	chisel = 1 *	chisel = $1 * 0.5 = 0.5$	chisel = 1 * 1 = 1	chisel = 1 * 2.5 = 2.5
		0 = 0			
10	znamen = $1 * 1 = 1$	znamen = 1	znamen = $1 * 1 = 1$	znamen = $1 * 1 = 1$	znamen = $1 * 1 = 1$
		* 1 = 1			
11	factor = $1 + 1 = 2$	factor = 1 +	factor = $1 + 1 = 2$	factor = $1 + 1 = 2$	factor = $1 + 1 = 2$
		1 = 2			
12	drib = -1 / 1 = -1	drib = 0 / 1	drib = 0.5 / 1 = 0.5	drib = 1 / 1 = 1	drib = 2.5 / 1 = 2.5
		= 0			
13	sum = 1 + -1 = 0	sum = 1 + 0	sum = 1 + 0.5 = 1.5	sum = 1 + 1 = 2	sum = 1 + 2.5 = 3.5
		= 1			
12	drib = 1 / 2 = 0.5		drib = $0.25 / 2 = 0.125$	drib = 1 / 2 = 0.5	drib = 6.25 / 2 = 3.125
13	sum = 0 + 0.5 = 0.5		sum = 1.5 + 0.125 =	sum = 2 + 0.5 = 2.5	sum = 3.5 + 3.125 = 6.625
			1.625		
12	drib = $-1 / 6 = -$		drib = 0.125 / 6 =	drib = 1 / 6 =	drib = 15.625 / 6 = 2.604167
	0.166666		0.020833	0.166667	
13	sum = 0.5 + -		sum = 1.625 +	sum = 2.5 + 0.166667	sum = 6.625 + 2.604167 =
	0.166666 = 0.3333333		0.020833 = 1.645833	= 2.666667	9.229167
12	drib = 1 / 24 =		drib = 0.0625 / 24 =	drib = 1 / 24 =	drib = 39.0625 / 24 =
	0.416666		0.002604	0.041667	1.627604
13	sum = $0.333333 +$		sum = 1.645833 +	sum = 2.666667 +	sum = 9.229167 + 1.627604
	0.416666 = 0.375		0.002604 = 1.648438	0.041667 = 2.708333	= 10.856771
12	drib = -1 / 120 = -		drib = 0.03125 / 120 =	drib = 1 / 120 =	drib = 97.6562 / 120 =
	0.008333		0.000260	0.008333	0.813802
13	sum = 0.375 + -		sum = 1.648438 +	sum = 2.708333 +	sum = 10.856771 + 0.813802
	0.008333 = 0.366666		0.000260 = 1.648698	0.008333 = 2.716667	= 11.670573
12	drib = 1 / 720 =		drib = 0.015625 / 720	drib = 1 / 720 =	drib = 244.141 / 720 =
	0.001388		= 0.000022	0.001389	0.339084
13	sum = 0.366666 +		sum = 1.648698 +	sum = 2.716667 +	sum = 11.670573 + 0.339084
	0.001388 = 0.368055		0.000022 = 1.64872	0.001389 = 2.718056	= 12.009657
12	drib = -1 / 5040 = -		drib = 0.0078125 /	drib = 1 / 5040 =	drib = 610.352 / 5040 =
	0.000198		5040 = 0.000002	0.000198	0.121102
13	sum = $0.368055 + -$		sum = 1.64872 +	sum = 2.718056 +	sum = 12.009657 + 0.121102
	0.000198 = 0.367857		0.000002 = 1.648721	0.000198 = 2.718254	= 12.130759
12	drib = 1 / 40320 =		drib = 0.00390625 /	drib = 1 / 40320 =	drib = 1525.88 / 40320 =
	0.000024		40320 = 0.000000	0.000025	0.037844
		<u> </u>			

# Основи алгоритмізації

	0 0 11	יו פ	алт орит	штоації	
13	sum = $0.367857 +$		sum = 1.648721 +	sum = $2.718254 +$	sum = 12.130759 + 0.037844
	0.000024 = 0.367881		0.000000 = 1.648721	0.000025 = 2.718279	= 12.168603
12	drib = -1 / 362880 =			drib = 1 / 362880 =	drib = 3814.7 / 362880 =
	0.000002			0.000003	0.010512
13	sum = $0.367881 +$			sum = 2.718279 +	sum = 12.168603 + 0.010512
	0.000002 = 0.367879			0.000003 = 2.718282	= 12.179115
12	drib = 1 / 3628800 =			drib = 1 / 3628800 =	drib = 9536.74 / 3628800 =
	0.000000			0.000000	0.002628
13	sum = $0.367879 +$			sum = 2.718282 +	sum = 12.179115 + 0.002628
	0.000000 = 0.367879			0.000000 = 2.718282	= 12.181743
12					drib = 23841.9 / 39916800 =
					0.000597
13					sum = 12.181743 + 0.000597
					= 12.18234
12					drib = 59604.6 / 479001600
					= 0.000124
13					sum = 12.18234 + 0.000124
					= 12.182465
12					drib = 149012 / 6227020800
					= 0.000024
13					sum = 12.182465 + 0.000024
					= 12.182489
12					drib = 372529 /
					87178291200 = 0.000004
13					sum = 12.182489 + 0.000004
					= 12.182493
12					drib = 931323 /
					1307674368000 = 0.000001
13					sum = 12.182493 + 0.000001
					= 12.182494
14	Вивести 0.36787	Вивести 1	Вивести 1.64872	Вивести 2.71828	Вивести 12.18249

## Основи алгоритмізації Висновки

Ми дослідили ітераційних циклічних алгоритмів та набули практичних навичок їх створення та використання під час складання програмних специфікацій. В результаті виконання лабораторної роботи ми отримали алгоритм для знаходження степені числа Ейлера, декомпозували задачу на 4 кроки: визначили стартові змінні, обрахували чисельник і знаменник, а потім і результат, перевірили потрібну точність і за недосягнення повторили. В процесі випробування ми розглянули випадки при від'ємному, нульовому, дробовому, одиничному та великому х і отримали результати, що збігаються з правильними на калькуляторі.