

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 32

Виконав студент ІП-13, Черкасов Станіслав Олексійович

(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 набути практичних навичок їх використання під час складання циклічних програмних специфікацій.

Варіант 32

32. Наближено (із заданою точністю ε) обчислити $(1 + X)^m$, за формулою $S = 1 + m \cdot X + \frac{m(m-1)X^2}{2!} + \frac{m(m-1)(m-2)X^3}{3!} + \frac{m(m-1)(m-2)(m-3)X^4}{4!} + \dots$

Постановка задачі

Для обчислення заданої формули побудуємо цикл. Із кожною ітерацією циклу будемо множити змінну ADD (елемент послідовності) на певний вираз, залежний від $I \in \mathbb{N}$ (лічильник циклу), та додавати її до суми SUM. Щоб обчислення було вірним, цикл має повторитися M (показник степеня) разів.

Побудова математичної моделі

Точність E

Точність запропоновано використовувати наступним чином: якщо Δ (модуль різниці попереднього та поточного елемента заданої множини) менше за E, припиняємо обчислення. Проте це працює лише з нескінченно малими множинами, в яких Δ прямує до нуля.

Задана у завданні множина не є нескінченно малою, тому точність E використовуємо іншим чином: замість того, щоб повторювати цикл M разів (що дає найточніший результат), повторюємо його **$\text{floor}(M * E)$** разів, де:
 $E \in [1; 0]$; $E \in \mathbb{R}$;

$\text{floor}(A)$ - функція, яка округлює значення A до найближчого цілого числа меншого за A

Тоді, якщо $E == 1$, результат буде найточнішим. Якщо $E == 0,5$, ми проведемо лише половину обчислень. Якщо $E == 0$, обчислень проведено не буде

Типи даних аргументів формули

X:

Якщо $E == 1$, формула знаходить точне значення при будь-якому дійсному X . Проте значення $E < 1$ призводять до помилок при обчисленні від'ємних іксів. Поступаємо наступним чином:

Для зручності просимо користувача одразу вводити значення $X+1$.

Якщо значення $X < 0$, домножаємо його на -1 та присвоюємо змінній NEG_X значення `True` (початкове значення $\text{NEG_X} == \text{False}$). Потім віднімаємо від отриманого значення 1 (бо користувач ввів значення на 1 більше потрібного)

Після обчислень, якщо $\text{NEG_X} == \text{False}$ та показник степеня є непарним ($M \% 2 == 1$), $\text{Sum} *= -1$.

$(A)\%(B)$ - функція, що знаходить остачу від ділення A на B

M:

Від'ємні та дробові значення M призводять до помилок при обчисленні формули. Отже, дозволяємо використовувати лише цілі значення M .

Якщо значення $M < 0$, домножаємо його на -1 та присвоюємо змінній NEG_M значення `True` (початкове значення $\text{NEG_M} == \text{False}$). Після обчислень, якщо $\text{NEG} == \text{True}$, $\text{SUM} = 1/\text{SUM}$.

Таблиця змінних

Змінна	Тип	Ім'я	Призначення
Задане число X	Дійсне	A	Початкове дане
Задане число M	Ціле	B	Початкове дане
Задана точність	Дійсне	E	Початкове дане
Індикатор $X < 0$	Булевий	NEG_X	Проміжне дане
Індикатор $M < 0$	Булевий	NEG_Y	Проміжне дане
Лічильник циклу	Натуральне	I	Лічильник циклу
Елемент суми	Дійсне	ADD	Проміжне дане
Шукана сума	Дійсне	SUM	Кінцеве дане

Нехай початкові значення Sum та ADD дорівнюють 1, 1 відповідно.

Із кожною ітерацією циклу змінюємо їх наступним чином:

$ADD *= ((M - (I - 1)) * X) / I;$

$SUM += ADD;$

Розв'язання

Програмні специфікації запишемо у формі псевдокоду та у вигляді блок-схеми.

Крок 1: визначимо основні дії

Крок 2: задамо початкові значення SUM, ADD, NEG_X, NEG_M

Крок 3: деталізуємо перевірку від'ємності X

Крок 4: деталізуємо перевірку від'ємності M

Крок 5: деталізуємо знаходження "Sum"

Крок 6: деталізуємо модифікацію "Sum" при від'ємному X

Крок 7: деталізуємо модифікацію "Sum" при від'ємному M

Псевдокод

Крок 1:

початок

введення X, M, E

задання SUM, ADD, NEG_X, NEG_M

перевірка від'ємності X

перевірка від'ємності M

знаходження "Sum"

модифікація "Sum" при

від'ємному X

модифікація "Sum" при

від'ємному M

виведення SUM

кінець

Крок 2:

початок

введення X, M, E

SUM = 1, ADD = 1, NEG_X = False,

NEG_M = False

перевірка від'ємності X

перевірка від'ємності M

знаходження "Sum"

модифікація "Sum" при

від'ємному X

модифікація "Sum" при

від'ємному M

виведення SUM

кінець

Крок 3:

початок

введення X, M, E

SUM = 1, ADD = 1, NEG_X = False,

NEG_M = False

якщо X < 0:

X *= -1

NEG_X = True

X -= 1

перевірка від'ємності M

знаходження "Sum"

модифікація "Sum" при

від'ємному X

модифікація "Sum" при

від'ємному M

виведення SUM

кінець

Крок 4:

початок

введення X, M, E

SUM = 1, ADD = 1, NEG_X = False,

NEG_M = False

якщо X < 0:

X *= -1

NEG_X = True

X -= 1

якщо M < 0:

M *= -1

NEG_X = True

знаходження "Sum"

модифікація "Sum" при

від'ємному X

модифікація "Sum" при

від'ємному M

виведення SUM

кінець

Крок 5:

початок

введення X, M, E

SUM = 1, ADD = 1, NEG_X = False, NEG_M = False

якщо X < 0:

X *= -1

NEG_X = True

X -= 1

якщо M < 0:

M *= -1

NEG_X = True

для I = 0 до floor(M * E) шаг 1:

I += 1

ADD *= ((M - (I - 1)) * X) / I

SUM += ADD

все для

модифікація "Sum" при від'ємному X

модифікація "Sum" при від'ємному M

виведення SUM

кінець

Крок 6

початок

введення X, M, E

SUM = 1, ADD = 1, NEG_X = False, NEG_M = False

якщо X < 0:

X *= -1

NEG_X = True

X -= 1

якщо M < 0:

M *= -1

NEG_X = True

для I = 0 до floor(M * E) шаг 1:

I += 1

ADD *= ((M - (I - 1)) * X) / I

SUM += ADD

все для

якщо NEG_X та M % 2 == 1:

SUM *= -1

модифікація “Sum” при від’ємному M

виведення SUM

кінець

Крок 7

початок

введення X, M, E

SUM = 1, ADD = 1, NEG_X = False, NEG_M = False

якщо X < 0:

X *= -1

NEG_X = True

X -= 1

якщо M < 0:

M *= -1

NEG_X = True

для I = 0 до floor(M * E) шаг 1:

I += 1

ADD *= ((M - (I - 1)) * X) / I

SUM += ADD

все для

якщо NEG_X та M % 2 == 1:

SUM *= -1

якщо NEG_M:

SUM = 1/SUM

виведення SUM

кінець

Блок-схема

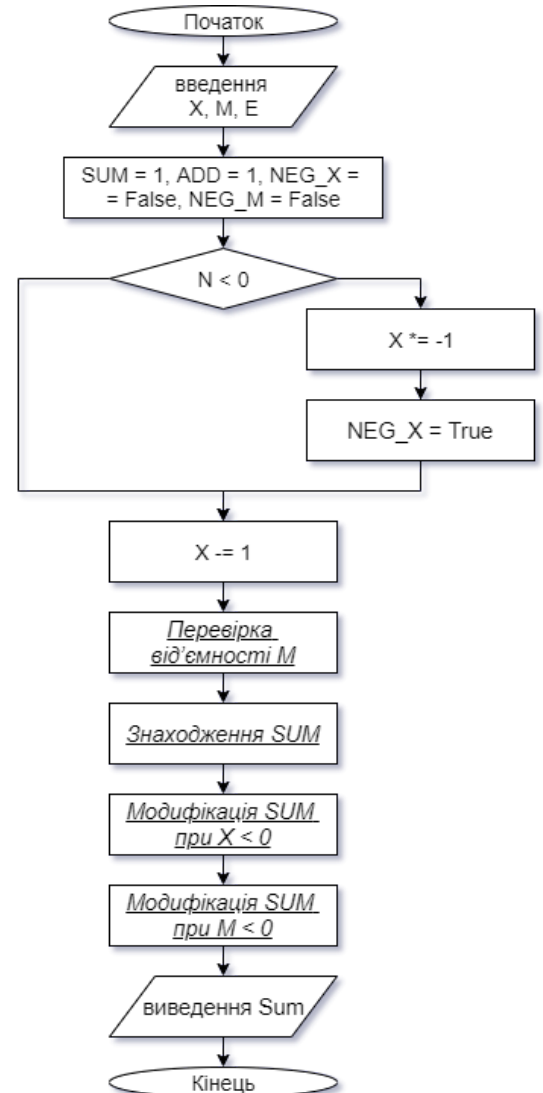
Крок 1



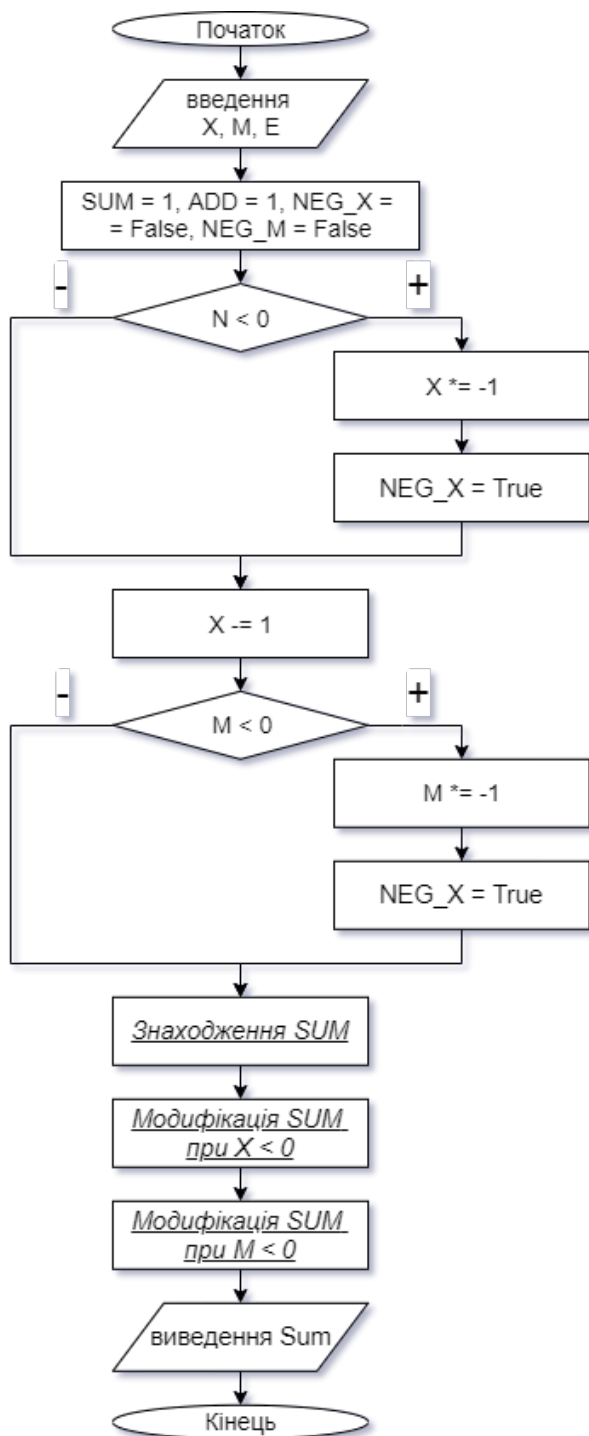
Крок 2



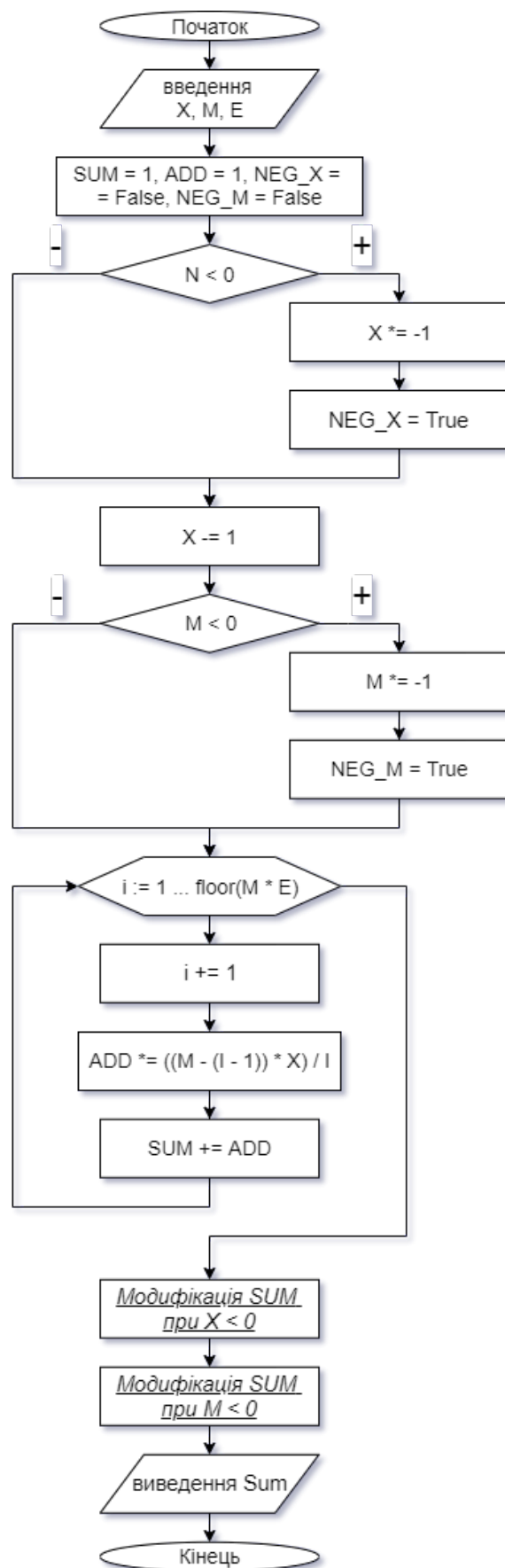
Крок 3



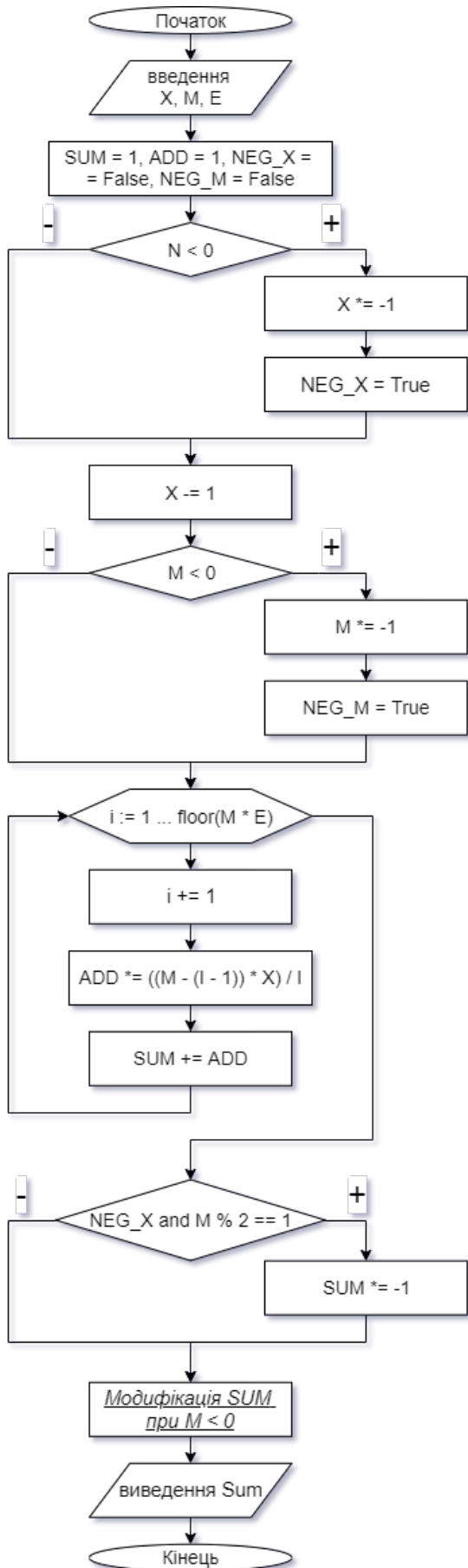
Крок 4



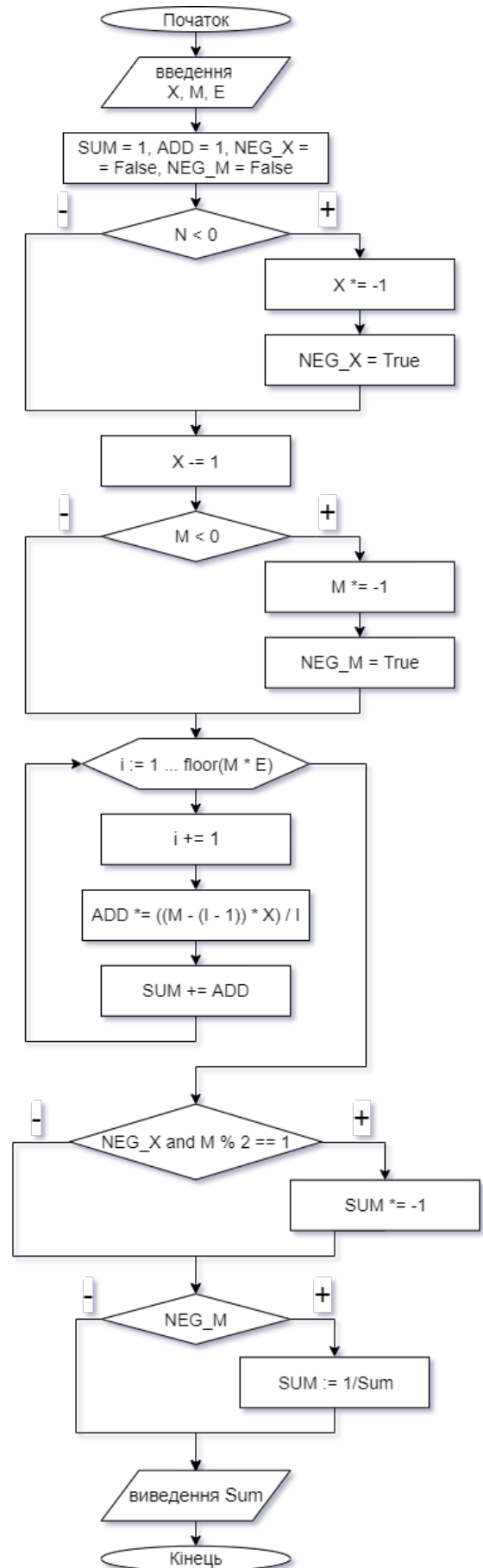
Крок 5



Крок 6



Крок 7



Випробування алгоритму

Блок	Дія
	початок
1	введення $X := 2$, $M := 5$, $E := 0.9$
2	$SUM = 1$; $ADD = 1$; $NEG_X = False$; $NEG_M = False$
3	пропустити крок $\text{бо } 2 < 0 == False$
4	$X = 1$ $2 - 1$
5	пропустити крок $\text{бо } 5 < 0 == False$
6	$SUM := 31$ $1 + 5 + 10 + 10 + 5$
7	пропустити крок $\text{бо } NEG_X == False$
8	пропустити крок $\text{бо } NEG_M == False$
9	виведення SUM
	кінець

Блок	Дія
	початок
1	введення $X := 5$, $M := -2$, $E := 1$
2	$SUM = 1$; $ADD = 1$; $NEG_X = \text{False}$; $NEG_M = \text{False}$
3	пропустити крок бо $5 < 0 == \text{False}$
4	$X = 4$ $5 - 1$
5	$M = 2$; $NEG_M = \text{True}$ бо $5 < 0 == \text{False}$
6	$SUM := 25$ $1 + 8 + 16$
7	пропустити крок бо $NEG_X == \text{False}$
8	$SUM = 0.04$ ($1/25$) бо $NEG_M == \text{True}$
9	виведення SUM

Блок	Дія
	початок
1	введення $X := -10$, $M := -5$, $E := 1$
2	$SUM = 1$; $ADD = 1$; $NEG_X = False$; $NEG_M = False$
3	$X = 10$; $NEG_X = True$ бо $-10 < 0 == True$
4	$X = 9$ $10 - 1$
5	$M = 5$; $NEG_M = True$ бо $-5 < 0 == True$
6	$SUM := 100000$ $1 + 45 + 810 + 729 + 32805 + 59049$
7	$SUM := -100000$ бо $NEG_X == True$
8	$SUM = -0.00001$ ($-1/100000$) бо $NEG_M == True$
9	виведення SUM

Висновок

Під час виконання цієї лабораторної роботи я вдосконалив навички написання математичної моделі, праці з блок схемами та випробування алгоритму.

Дослідив подання операторів повторення дій та набув практичних навичок їх використання під час складання циклічних програмних специфікацій.