*Додаток 1*

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 № 4 з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«Дослідження арифметичних циклічних алгоритмів»

Варіант 24

Виконав студент ІП-15 Омран Мохамад

(шифр, прізвище, ім'я, по батькові)

Перевірив

(прізвище, ім'я, по батькові)

Київ 2021

# Лабораторна робота 4

**Дослідження арифметичних циклічних алгоритмів**

**Мета** – дослідити особливості роботи арифметичних циклів та набути практичних навичок їх використання під час складання програмних специфікацій.

# Варіант 24

Визначити різницю між результатами обчислень числа  за допомогою ряду

Грегорі:  = 1 − 1 + 1 − 1 + ⋯ (−1)𝑘+1 ∗ 1

та добутком Валіса: 

= 2 ∗ 2 ∗

4

4 4 6

3 5 7

2𝑘+1

2 1 3

∗ ∗

3 5 5

∗ … для n ітерацій, вважаючи ітерацією одну операцію додавання або

множення. Порівняти одержані результати з точним значенням числа .

# Побудова математичної моделі:

Необхідно за домопогою двох арифметичних циклів та заданих формул обчислити значення  за рядом Грегорі (gre := gre + pow(-1, i) \* (1 / (2\* i +1))) та за добутком Валіса (val := val \* (2 \* (i // 2)) / (2 \* (i // 2) + 1)) та в кінці, використовуючи умовні оператори, порівняти їх з точним значенням  = 3,141592653589… При тестуванні будемо округлювати це значення.

# Таблиця змінних

|  |  |  |  |
| --- | --- | --- | --- |
| Опис | Тип | Ім’я | Призначення |
| Кількість ітерацій | int | n | Вхідні дані |
| Значення  із формули Грегорі | float | gre | Проміжні дані |
| Значення  із формули Валіса | float | val | Проміжні дані |
| Лічильник | int | i | Проміжні дані |
| Різниця точного значення  і gre | float | gre\_dif | Вихідні дані |
| Різниця точного значення  і val | float | val\_dif | Вихідні дані |

**Таблиця констант**

|  |  |  |
| --- | --- | --- |
| Опис | Позначення | Значення |
| Число пі |  | 3,1415926535897932384626433… |

# Таблиця функцій

|  |  |  |
| --- | --- | --- |
| Назва | Позначення | Опис |
| Піднесення до степеня | pow | pow(a, b) == 𝑎𝑏 |
| Модуль числа | abs | Якщо a >= 0 то abs(a) == a  Якщо а < 0 то abs(a) == -a |

1. **Розв’язання:**

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

Крок 1. Визначення основних дій;

Крок 2. Деталізуємо дію знаходження числа  за допомогою ряду Грегорі; Крок 3. Деталізуємо дію знаходження числа  за допомогою добутку Валіса; Крок 4. Деталізуємо дію порівняння отриманих значень і точного значення ;

# Псевдокод:

*Крок 1.*

# початок

**введення** n

знаходження числа  за допомогою ряду Грегорі знаходження числа  за допомогою добутку Валіса порівняння отриманих значень і точного значення  **виведення** gre\_dif, val\_dif

# кінець

*Крок 2.*

# початок

**введення** n gre := 0

# повторити для i від 0 до n - 1

gre := gre + pow(-1, i) \* (1 / (2\* i +1))

# все повторити

gre := gre \* 4

знаходження числа  за допомогою добутку Валіса порівняння отриманих значень і точного значення  **виведення** gre\_dif, val\_dif

# кінець

*Крок 3.*

# початок

**введення** n gre := 0

val := 1

# повторити для i від 0 до n - 1

gre := gre + pow(-1, i) \* (1 / (2\* i +1))

# все повторити

gre := gre \* 4

# повторити для i від 1 до n якщо i % 2 == 1

**то**

val := val \* (2 \* (i // 2 +1)) / (2 \* (i // 2 + 1) - 1)

# інакше

val := val \* (2 \* (i // 2)) / (2 \* (i // 2) + 1)

**все якщо все повторити** val := val \* 2

порівняння отриманих значень і точного значення 

**виведення** gre\_dif, val\_dif

# кінець

*Крок 4.*

# початок

**введення** n gre := 0

val := 1

# повторити для i від 0 до n - 1

gre := gre + pow(-1, i) \* (1 / (2\* i +1))

# все повторити

gre := gre \* 4

# повторити для i від 1 до n якщо i % 2 == 1

**то**

val := val \* (2 \* (i // 2 +1)) / (2 \* (i // 2 + 1) - 1)

# інакше

val := val \* (2 \* (i // 2)) / (2 \* (i // 2) + 1)

**все якщо все повторити** val := val \* 2

gre\_dif := abs( - gre) val\_dif := abs( - val) **виведення** gre\_dif, val\_dif

# кінець

1. **Блок – схема**

|  |  |
| --- | --- |
| Знаходження числа π за  допомогою ряду Грегорі | |
|  |  |
| Знаходження числа π за допомогою  добутку Валіса | |
|  |  |
| Порівняння отриманих  значень і точного  значення π | |
|  |  |

Початок

Введення n

gre := 0

Початок

Введення n

i := 0, -

n-1, 1

+

gre := gre + pow(-

1, i) \* (1 / (2\* i +1))

val := 1

gre := 0

Початок

Введення n

i := 0, -

n-1, 1

+

gre := gre + pow(-

1, i) \* (1 / (2\* i +1))

val := 1

gre := 0

Початок

Введення n

i := 1, n, 1

Виведення gre\_dif, val\_dif

Кінець

i := 1, n, 1

-

+

+

-

i % 2 == 1

Виведення gre\_dif, val\_dif

Кінець

val := val \* 2

val := val \* (2 \* (i // 2)) / (2 \* (i // 2) +

1)

val := val \* (2 \* (i // 2 +1)) / (2 \* (i // 2

+ 1) - 1)

gre := gre \* 4

i := 0, -

n-1, 1

+

gre := gre + pow(-

1, i) \* (1 / (2\* i +1))

Порівняння отриманих

значень і точного значення π

Знаходження числа π за допомогою

добутку Валіса

gre := gre \* 4

gre := gre \* 4

-

+

+ -

i % 2 == 1

val := val \* 2

val := val \* (2 \* (i // 2)) / (2 \* (i // 2) +

1)

val := val \* (2 \* (i // 2 +1)) / (2 \* (i // 2

+ 1) - 1)

Виведення gre\_dif, val\_dif

Кінець

Порівняння

отриманих

значень і точного

значення π

Виведення gre\_dif, val\_dif

Кінець

val\_dif := abs(π - val)

gre\_dif :=

abs(π - gre)

# Тестування

|  |  |
| --- | --- |
| Блок | Дія |
|  | Початок |
| 1 | Введення:  n = 4 |
| 2 | i = 0: gre = 1  i = 1: gre = 0.6667  i = 2: gre = 0.8667  i = 3: gre = 0.7238  gre = 0.7238 \* 4 = 2.8952 |
| 3 | i = 1: val = 2  i = 2: val = 1.3333  i = 3: val = 1.7777  i = 4: val = 1.4223  val = 1.4223 \* 2 = 2.8446 |
| 4 | gre\_dif = abs (3.1416 - 2.8952) = 0.2464  val\_dif = abs (3.1416 - 2.8446) = 0.297 |
| 5 | Виведення: gre\_dif = 0.2464  val\_dif = 0.297 |
|  | Кінець |

**Висновки:**

Я дослідив подання операторів повторення дій та набув практичних навичок їх використання під час складання циклічних програмних специфікацій. В процесі виконання лаболаторної роботи я навчився використовувати цикли з лічильником, описувати їх у псевдокоді та позначати на блок-схемі.