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 № 5 з дисципліни

«Алгоритми та структури даних.

Основи алгоритмізації»

«Дослідження складних циклічних алгоритмів» Варіант 10

Виконав студент ІП-12 Горобець Олексій Сергійович

(шифр, прізвище, ім'я, по батькові)

Перевірив

( прізвище, ім'я, по батькові)

Київ 202 1

**Лабораторна робота 5**

**Дослідження складних циклічних алгоритмів**

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

**Варіант 10**

***Задача 1.10***.

10. Дано натуральне число n. Знайти всі числа Мерсена, що не перевищують число n. Просте число називається числом Мерсена, якщо його можна представити у вигляді 2^p −1, де p — теж просте число.

***Постановка задачі*.** Результатом розв’язку є числа, кожне з яких менше заданого n та відповідають умові: число можна подати у вигляді an=2^p-1, де p та an – прості числа. Для визначення цих чисел нам необхідне задане значення цілого числа n. Інших початкових даних для розв’язку не потрібно.

***Математична побудова.*** Складемо таблицю змінних.

|  |  |  |  |
| --- | --- | --- | --- |
| **Змінна** | **Тип** | **Ім’я** | **Призначення** |
| **Число n** | Цілий | n | Початкове дане |
| **Число divisor** | Цілий | divisor | Проміжна змінна |
| **Число mers\_numb** | Цілий | mers\_numb | Проміжна змінна |

Таким чином, математичне формулювання задачі зводиться до розрахування an за відповідною формулою an = 2^p -1 , де p – просте, та перевірки an на простоту, тобто якщо число an просте, то воно є шуканим числом Мерсена.

***Розв’язання:***

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

**Крок 1**. Визначимо основні дії.

**Крок 2.** Деталізуємо дію пошуку простих чисел, які задовольняють умову

**Крок 3**. Деталізуємо дію перевірки числа на простоту

**Крок 4.** Деталізуємо дію перевірки на просте число Мейсона

***Псевдокод***

*крок 1*

**початок**

Пошук простих чисел які задовільняють умову

Перевірка числа на простоту

Перевірка числа на просте число Мейсона

**Кінець**

*крок 2*

**початок**

**Ввід** n

**повторити**

**для i від 2 до log2(n+1)**

Перевірка числа на простоту

Перевірка числа на просте число Мейсона

**все повторити**

**кінець**

*крок 3*

**початок**

**Ввід** n

**повторити**

**для i від 2 до log2(n+1)**

divisor:=2

**поки** i % divisor != 0 && divisor <= sqrt(i)

**повторити**

divisor:=divisor+1

**все повторити**

**якщо** divisor>sqrt(i)

**то**

Перевірка числа на просте число Мейсона

**все якщо**

**все повторити**

**кінець**

**початок**

Крок 4

**Ввід** n

**повторити**

**для i від 2 до log2(n+1)**

divisor:=2

**поки** i % divisor != 0 && divisor <= sqrt(i)

**повторити**

divisor:= divisor + 1

**все повторити**

**якщо** divisor>sqrt(i)

**то**

mers\_numb:= pow(2, i) – 1

divisor:= 2

**поки** mers\_numb % divisor != 0 && divisor <= sqrt(mers\_numb)

**повторити**

divisor:=divisor+1

**все повторити**

**якщо** divisor > sqrt(mers\_numb)

**то**

**Вивід** mers\_numb

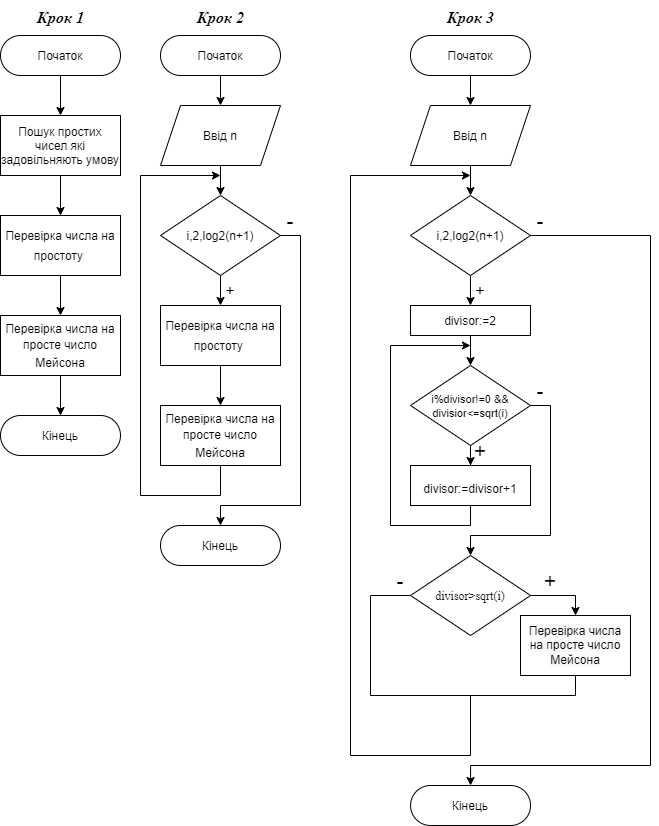
**все якщо**

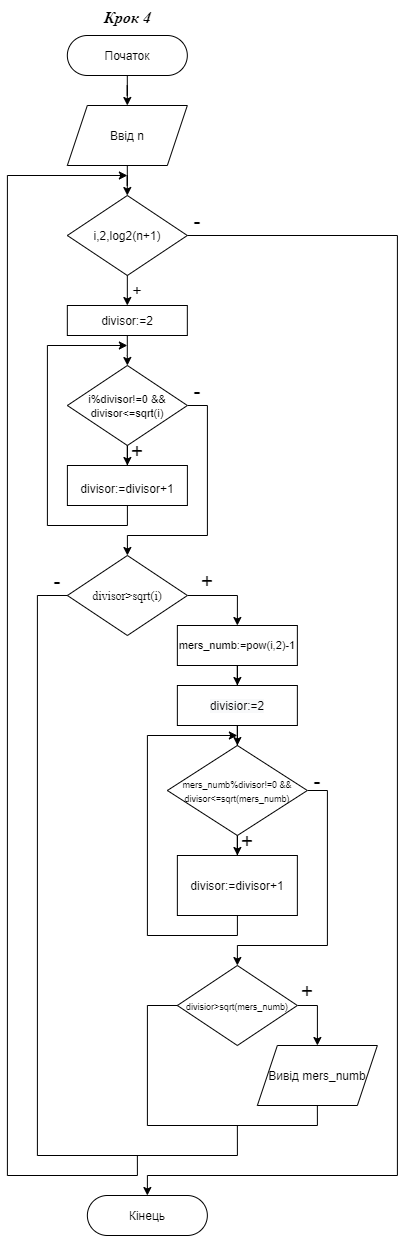
**все якщо**

**все повторити**

**кінець**

***Блок-схема***

**

**

***Випробування алгоритму***

Перевіримо правильність роботи алгоритму на довільних конкретних значеннях початкових даних:

|  |  |
| --- | --- |
| **Блок** | **Дія** |
|  | Початок |
| 1 | Введення: n = 4, x = 300 |
| 2 | поторити для i від 1 до 4 i=1 factorial(2\*1)=1\*2=2 val\_ai:= pow(x,i) / factorial = 300/2=150 |
| 3 | поторити для i від 1 до 4 i=2 factorial(2\*2)=1\*2\*3\*4=24 val\_ai:= pow(300, 2) / factorial(2\*2) = 3750 |
| 4 | поторити для i від 1 до 4 i=3 facorial(2\*3)=1\*2\*3\*4\*5\*6=720 val\_ai:= pow(300, 3) / factorial(2\*3) = 37500 |
| 5 | поторити для i від 1 до 4 i=4 factorial(2\*4)=1\*2\*3\*4\*5\*6\*7\*8=40320 val\_ai:= pow(300, 4) / factorial(2\*4) = 200892.85714285713 |
| 6 | Кінець |

***Висновок***

Під час виконання лабораторної було досліджено особливості роботи складних циклів та набуто практичних навичок їх використання під час складання програмних специфікацій. Розроблено алгоритм визначення членів ряду за заданою формулою. Покращено навички написання псевдокоду, побудови та тестування алгоритмів.