Міністерство освіти і науки України

Національний технічний університет України “Київський політехнічний

інститут імені Ігоря Сікорського ˮ

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 5 з дисципліни

“Алгоритми та структури даних-1.

Основи алгоритмізації ˮ

“ Дослідження лінійних алгоритмів ˮ

Варіант:12

Виконав студент: ІП-12 Єльчанінов Артем Юрійович

(шифр, прізвище, ім'я, по батькові)

Перевірив: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(прізвище, ім'я, по батькові)

Київ 2021

**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 12**

**Задача:** Дано натуральне число ***n***. Визначити всі натуральні числа, менші за ***n*** і взаємно прості з ним.

**Постановка задачі**

Результатом розв’язку задачі є отримання натуральних чисел, які менші за введене число ***n*** і є взаємно прості з ним. Враховуючи специфіку задачі спершу ми визначаємо найбільший спільний дільник(НСД) ***n*** та меншого за нього натурального числа за алгоритмом Евкліда, а потім визначаємо чи число є взаємно простим з ним.

Задача буде виконана тоді, коли всі натуральні числа, які менші за ***n*** та мають з ним НСД, яке є **1,** будуть визначені.

**Математична модель**

|  |  |  |  |
| --- | --- | --- | --- |
| **Змінна** | **Тип** | **Ім’я** | **Призначення** |
| Натуральне число | Цілий | n | Вхідне дане |
| Допоміжна змінна, якій присвоюють значення ***n*** для обчислення НСД | Цілий | N | Проміжне дане |
| Допоміжна змінна, якій присвоюють значення натуральних чисел менших за ***n*** | Цілий | m | Проміжне дане |
| Номер ітерацій | Цілий | i | Проміжне дане |
| Натуральне число, яке менше за ***n*** та взаємно просте з ним | Цілий | num | Вихідне дане |

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

Крок 1. Визначимо основні дії.

Крок 2. Деталізуємо дію перебору натуральних чисел менших за n

Крок 3. Деталізуємо дію обчислення НСД за алгоритмом Евкліда

Крок 4. Деталізуємо дію визначення num

**Псевдокод алгоритму**

**Крок 1:**

**Початок**

**Введення** n

Декларування змінної

Перебір натуральних чисел менших за n

Обчислення НСД

Визначення num

**Виведення** num

**Кінець**

**Крок 2:**

**Початок**

**Введення** n

N:= n

Перебір натуральних чисел менших за n

Обчислення НСД

Визначення num

**Виведення** num

**Кінець**

**Крок 3:**

**Початок**

**Введення** n

N:= n

**для** і **від** 1 **до** n-1 **з кроком** 1

Обчислення НСД

Визначення num

**Виведення** num

**все повторити**

**Кінець**

**Крок 4:**

**Початок**

**Введення** n

N:= n

**для** і **від** 1 **до** n-1 **з кроком** 1

m:= i

**поки** m!= 0 && N!= 0

**повторити**

**якщо** m > N

**то**

m:= m % N;

**інакше**

N:= N % m;

**все якщо**

**все повторити**

Визначення num

**Виведення** num

**все повторити**

**Кінець**

**Крок 5:**

**Початок**

**Введення** n

N:= n

**для** і **від** 1 **до** n-1 **з кроком** 1

m:= i

**поки** m!= 0 && N!= 0

**повторити**

**якщо** m > N

**то**

m:= m % N;

**інакше**

N:= N % m;

**все якщо**

**все повторити**

**якщо** (N + m) == 1

**то**

num:= i

**Виведення** num

**все якщо**

N:= n;

**все повторити**

**Кінець**

**Блок-схема**

**Крок 1: Крок 2:**

 

**Крок 3:**



**Крок 4:**



**Крок 5:**



**Випробування алгоритму:** Перевіримо правильність алгоритму на довільних конкретних значеннях початкових даних.

|  |  |
| --- | --- |
| **Блок** | **Дія** |
|  | **Початок** |
| 1 | Введення n = 4 |
| 2 | N = 4 |
| 3 | i = 1; i <= 3; +1 == true |
| 4 | m = 1 |
| 5 | 1!= 0 && 4!= 0 == true |
| 6 | 1 > 4 == false |
| 7 | N = 4 % 1 = 0 |
| 8 | 1!= 0 && 0!= 0 == false |
| 9 | (0 + 1) == 1 true |
| 10 | num = 1 |
| 11 | *Виведення:* 1 |
| 12 | N = 4 |
| 13 | I = 2; i <= 3; +1 == true |
| 14 | m = 2 |
| 15 | 2!= 0 && 4!= 0 == true |
| 16 | 2 > 4 == false |
| 17 | N = 4 % 2 = 0 |
| 18 | 2!= 0 && 0!= 0 == false |
| 19 | (0 + 2) == 1 false |
| 20 | N = 4 |
| 21 | i = 3; i <= 3; +1 == true |
| 22 | m = 3 |
| 23 | 3!= 0 && 4!= 0 == true |
| 24 | 3 > 4 == false |
| 25 | N = 4 % 3 = 1 |
| 26 | 3!= 0 && 1!= 0 == true |
| 27 | 3 > 1 == true |
| 28 | m = 3 % 1 = 0 |
| 29 | 0!= 0 && 1!= 0 == false |
| 30 | (1 + 0) == 1 true |
| 31 | num = 3 |
| 32 | *Виведення:* 3 |
| 33 | N = 4 |
| 34 | i = 4; i <= 3; +1 == false |
|  | **Кінець** |

**Висновок.**

У результаті лабораторної роботи було розроблено математичну модель, що відповідає постановці задачі; псевдокод та блок-схеми, які пояснюють логіку алгоритму. Було набуто практичного новичок у складанні складних циклічних алгоритмів та їх інтерпретації у блок-схеми і псевдокод.

Алгоритм був випробуваний з введенням значень: n = 4 , у підсумку було отримано, що num = 1 і 3. Таким чином, було доведено вірність складеного алгоритму. Отже, його можна застосовувати для визначення всіх натуральних чисел, менших за натуральне число(***n***)і взаємно простих з ним.