

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 № 6 з дисципліни  
«Алгоритми та структури даних-1.  
Основи алгоритмізації»  
«Дослідження рекурсивних алгоритмів»

Варіант \_\_16\_\_

Виконав студент \_\_ІП-15, Куманецька І. В.\_\_\_\_  
Перевірив \_Вечерковська А. С.\_\_\_\_\_

Київ 2021

## Лабораторна робота 6

### Дослідження рекурсивних алгоритмів

**Мета** – дослідити особливості роботи рекурсивних алгоритмів та набути практичних навичок їх використання під час складання програмних специфікацій підпрограм.

### Індивідуальне завдання

#### Варіант 16

16. Дано натуральне число  $n$ . Отримати всі прості дільники цього числа.

#### Постановка задачі

Знайти всі прості дільники заданого натурального числа. Результатом виведення мають бути власне дільники.

#### Побудова математичної моделі

Змінна	Тип	Ім'я	Призначення
Задане натуральне число	Ціле натуральне	chislo	Вхідні дані
Число, для якого виводимо прості дільники	Ціле натуральне	number	Проміжні дані, аргумент функції delim
Можливий простий дільник	Ціле натуральне	divider	Проміжні дані, аргумент функції delim

Розрахування та вивід простих дільників виконується за допомогою функції `delim(number, divider)`. Термінальна гілка виконується, якщо саме число дорівнює 1, тобто простих дільників немає. У рекурсивній гілці знаходиться можливий простий дільник, після чого число ділиться на максимальний степінь

отриманого дільника. Потім рекурсивно запускається ця ж функція від вже зменшеного числа та наступного можливого значення дільника.

Через `sqrt()` позначається знаходження квадратного кореня від числа. Через `a /= b` позначається операція  $a = a / b$ . Через `%` позначається операція знаходження остачі від ділення. Через `!=` позначається «не дорівнює.»

## **Розв'язання**

Програмні специфікації запишемо у псевдокодi, графічній формi у вигляді блок-схеми та у вигляді коду.

Крок 1. Визначення основних дій.

Крок 2. Деталізація розрахування та виведення простих дільників числа за допомогою рекурсивної функції.

## **Псевдокод**

*Крок 1*

**початок**

введення `chislo`

розрахування та виведення простих дільників `chislo`

**кінець**

*Крок 2*

**початок**

введення `chislo`

`delim(chislo, 2)`

**кінець**

*Підпрограма*

**delim(number, divider)**

**якщо** number = 1

**то**

виведення «Більше простих дільників немає»

**інакше**

**поки** number % divider != 0

**повторити**

divider += 1

**все повторити**

виведення divider

**повторити**

number /= divider

**поки** number % divider = 0

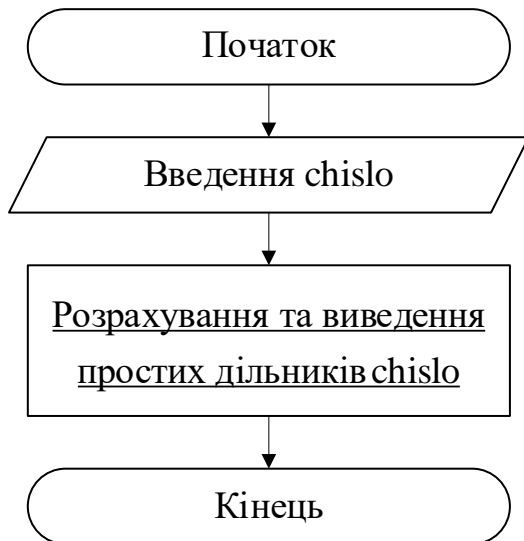
**все повторити**

delim(number, divider + 1)

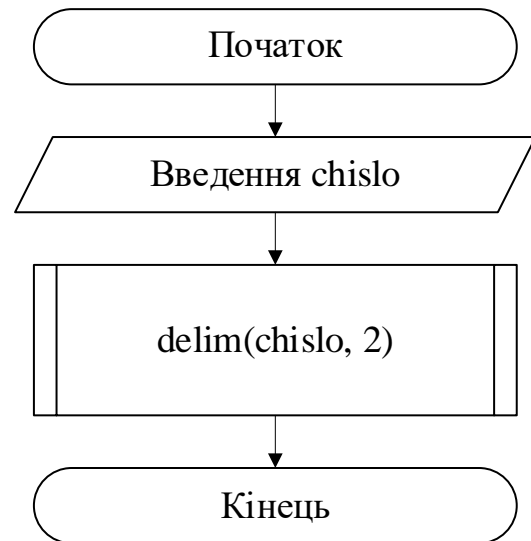
**кінець**

## Блок-схема

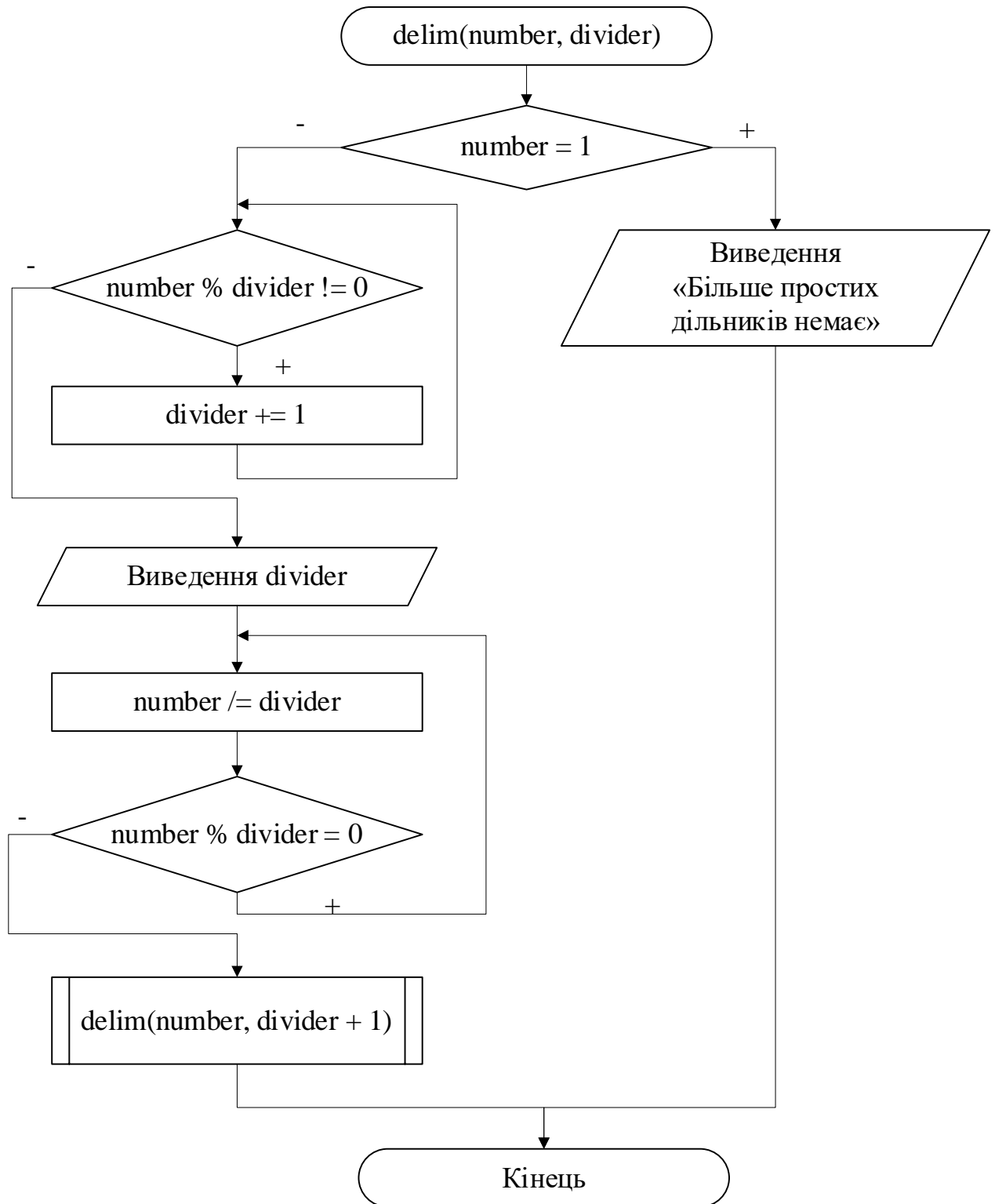
*Крок 1*



*Крок 2*



*Підпрограма*



## Код

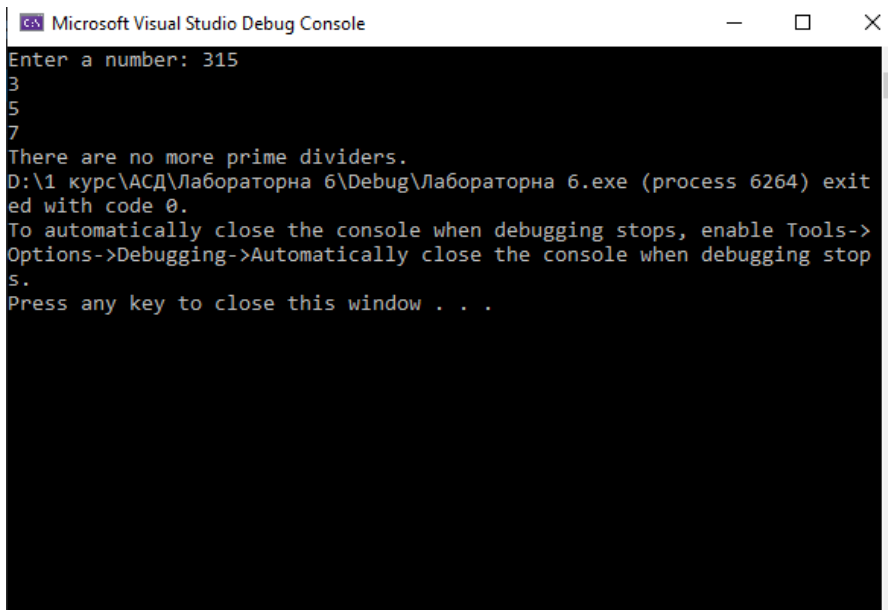
```
#include <stdio.h>
#include <iostream>
#include <math.h>

using namespace std;
void delim(int number, int divider);

int main()
{
    int chislo;
    cout << "Enter a number: ";
    cin >> chislo;

    delim(chislo, 2);
}

void delim(int number, int divider)
{
    if (number == 1) {
        cout << "There are no more prime dividers.";
        return;
    }
    else {
        while (number % divider != 0) {
            divider++;
        }
        cout << divider << endl;
        do
        {
            number /= divider;
        } while (number % divider == 0);
        delim(number, divider + 1);
    }
}
```



```
Microsoft Visual Studio Debug Console
Enter a number: 315
3
5
7
There are no more prime dividers.
D:\1 курс\АСД\Лабораторна 6\Debug\Лабораторна 6.exe (process 6264) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

## Висновок

Було досліджено особливості роботи рекурсивних алгоритмів та набути практичних навичок їх використання під час складання програмних специфікацій підпрограм. В результаті виконання лабораторної роботи було знайдено прості дільники числа, розділивши задачу на 2 кроки: визначення основних дій, деталізація розрахування та виведення простих дільників числа за допомогою рекурсивної функції. В процесі випробування було розраховано прості дільники числа 315: 3, 5, 7.