# **Лабораторна робота №**10**.** Вступ до документації проекту**.**

## **1. Вимоги**

* *Розробник*: Зозуля Ігор
* *Загальне завдання*: реалізувати програму з використанням функцій.
* *Індивідуальне завдання*: розробити програму яка знаходить НСК для двох чисел.

## **2. Опис програми**

* *Функціональне призначення*: Визначення найбільшого спільного дільника для двох випадкових чисел.
* *Опис логічної структури*:
  + - Функція `main`. Формує два випадкових числа. Викликає функцію `findGCF`. Схема алгоритму функції main наведена на рис. 1.

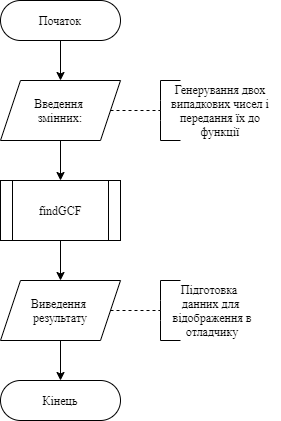


Рисунок 1 - блок-схема алгоритму функції main

* Функція `findGCF`. Знаходить НСК для двох чисел. Схема алгоритму функції findGCF наведена на рис. 2.

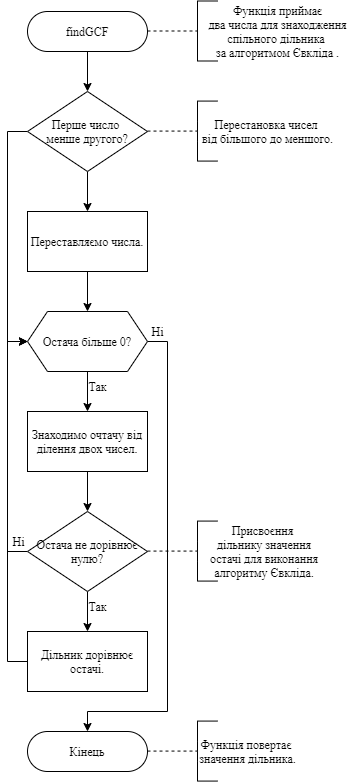


Рисунок 2 - блок-схема алгоритму функції findGCF

* *Структура програми*:

└── lab10/

├── Doxyfile

├── Makefile

├── README.md

├── doc/

│ ├── lab10.md

│ ├── lab10.docx

│ ├── lab10-doc.pdf

│ └── lab10-md.pdf

├── src/

└── main.c

* *Важливі елементи програми*:
  + Формування масиву. Формуємо два випадкових числа з випадковими значеннями.

int rand\_max = 100;

const int a = rand() % rand\_max + 1;

const int b = rand() % rand\_max + 1;

* Знаходження спільного дільника методом Евкліда.

while (remainder > 0) {

remainder = num1 % num2;

num1 = num2;

num2 = remainder;

if (remainder != 0) {

divider = remainder;

}

}

## **3. Варіанти використання**

З допомогою відлогаднику ставимо точку зупинки на строках виклику функції (для визначення початкового стану масиву) та `return 0;` (для визначення стану відсортованого масиву).

Виклик функції findGCF з початковими даними, що зображені на рис. 3.

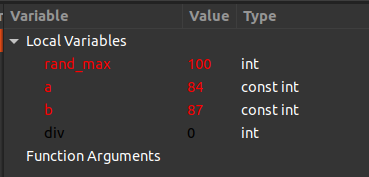


Рисунок 3 — початкові дані

Результати виконання функції наведені на рис. 4.

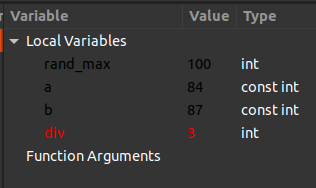


Рисунок 4 — результат виклику функції findGCF

## **Висновки**

Для виконання лабораторної роботи я навчився створювати та реалізовувати алгоритми функції, створювати схеми алгоритмів, та оформлювати документацію.