Лабораторна робота № 5. Циклічні структури

1 ВИМОГИ

1.1 **Розробник**

* Фоменко Гліб Володимирович;
* студент групи КІТ-121Г;
* 08-гру-2021.

1.2 **Загальне завдання**

Розробити програму використовуючи функції

1.3 **Індивідуальне завдання**

Сформувати функцію, яка буде знаходити найбільший загальний дільник двох цілих чисел

2 ОПИС ПРОГРАМИ

2.1 **Функціональне призначення**

Програма призначена для знаходження найбільшого загального дільника двох цілих чисел

При знаходженні числа діють наступні *обмеження*:

* числа які передаються в функцію повинні бути int

Результат зберігається у змінній *greatest\_common\_divisor*.

Подивитися результат виконання можна тільки в режимі дебагінгу

2.2 **Опис логічної структури**

Програма бере два числа, більшому з них привласнює залишок від ділення більшого на менше. Це повторюється доки в одному з чисел не буде нуль.

2.2.1 Функція знаходження найбільшого загального дільника

int get\_greatest\_common\_divisor(int first\_num, int second\_num)

*Призначення*: повертання найбільшого загального дільника двох чисел які передані як аргументи.

*Опис роботи*: бере два переданих в аргумент числа, методом ейлера знаходить найбільший загальний дільник та повертає його.

2.2.2 Основна функція

int main(int argc, char \*\*argv)

*Призначення*: головна функція.

*Опис роботи*:

* якщо через аргументи командної строки нічого не передасться, то для аргументів *first\_num* та *second\_num* створюються випадкові числа.
* числа, які були передані через аргументи строки або випадкові, передаються в функцію *get\_greatest\_common\_divisor*.
* результат виконання записується в змінну *greatest\_common\_divisor*.

2.2.3 Структура проекту

lab05

├── Doxyfile

├── Makefile

├── doc

│ ├── lab05.docx

│ ├── lab05.md

│ └── lab05.pdf

└── src

└── main.c

2.3 **Важливі фрагменти програми**

Метод ейлера

int get\_greatest\_common\_divisor(int first\_num, int second\_num)

{

int greatest\_common\_divisor = 0;

while (greatest\_common\_divisor == 0) {

if (first\_num > second\_num) {

first\_num = first\_num % second\_num;

} else {

second\_num = second\_num % first\_num;

}

if (first\_num == 0 || second\_num == 0) {

greatest\_common\_divisor = first\_num == 0 ? second\_num first\_num;

}

}

return greatest\_common\_divisor;

}

3 ВАРІАНТ ВИКОРИСТАННЯ

Для демонстрації результатів задачі використовується:

* покрокове виконання програми в утиліті lldb;

**Варіант використання**: послідовність дій для запуску програми у режимі відлагодження:

* запустити програму у відлагоднику lldb;
* поставити точку зупинки на функції main (строка з return 0;);
* запустити програму;
* подивитись результати виконання програми, оглядаючи значення змінної greatest\_common\_divisor

Process 4494 stopped

\* thread #1, name = 'main.bin', stop reason = breakpoint 1.1

frame #0: 0x00000000004011e5 main.bin`main(argc=1, argv=0x00007fffffffe2e8) at main.c:55:2

52 }

53 int greatest\_common\_divisor = get\_greatest\_common\_divisor(first\_num, second\_num);

54

-> 55 return 0;

56 }

57

58 int get\_greatest\_common\_divisor(int first\_num, int second\_num)

(lldb) p greatest\_common\_divisor

(int) $0 = 4

(lldb) p first\_num

(int) $1 = 1975032828

(lldb) p second\_num

(int) $2 = 169278508

ВИСНОВКИ

При виконанні даної лабораторної роботи було набуто практичного досвіду роботи із методом ейлера для знаходження найбільшого загального дільника.