Лабораторна робота № 9. Вступ до документації коду(частина 1)

1 ВИМОГИ

1.1 Розробник

- Михневич Владислав Вікторович
- Студент 1-го курсу
- Групи КІТ-120а

1.2 Загальне завдання

Для попереднього проекту необхідно додати можливість генерації Doxygen документації.

2 ОПИС ПРОГРАМ

2.1 Функціональне призначення

Основу програм переробити так, щоб головна функція main() викликала розроблену функцію в залежності від умови.

3 РОЗРОБКА ПРОГРАМИ

3.1 Редагуємо код на титульну сторінку

Додаємо зміни до коду (див. рис.1)

```
/**
 * @mainpage
 * # Загальне завдання
 * 1. **Розробити** функцию, що буде рухувати квадратний корінь з числа, без допомоги зовнішних бібліотек
 * 2. **Розробити** функцію, що буде премножати матрицю саму на себе
 * 3. **Розробити** варіаивну функцію, що буде рахувати кількість пар чисел, в яких наступне число меньше попереднього
 *
 * @author Mykhnevych V.V.
 * @date 14-dec-2020
 * @version 1.0
 */
```

Рисунок 1 – Код титульної сторінки

3.2 Вигляд титульної сторінки

Згенеруємо наш файл за допомогою команди `make doxygen` та побачимо зміни в файлі index.html (див. рис. 2).

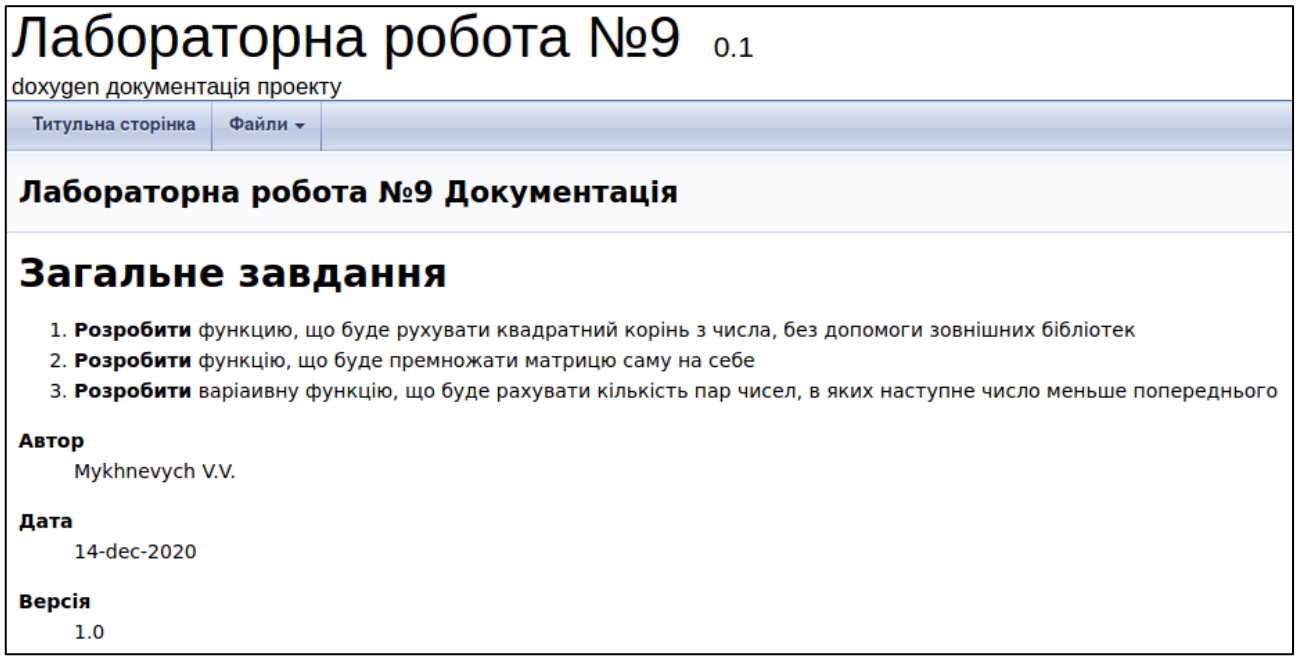


Рисунок 2 – Вигляд титульної сторінки

3.3 Опис функцій

В коді описуємо всі функції (див. рис. 3).

```
/**
 * @file main_1.c
 * @brief Квадратний корінь числа
 * @author Mykhnevych V.V.
 * @date 20-dec-2020
 * @version 1.0
 */

#include <stdio.h>
#include <time.h>
#include <stdlib.h>

/**
 * функція для розрахунку квадратного кореня з числа
 * @return root
 * @param root добутий корінь із заданого числа
 */

double result_root(double root, int n);

/**
 * Головна функція.
 *
 * Послідовність дій:
 * - створення змінної n, в яку буде записане число
 * - створення змінної root, в яку буде записаний результат виконання функіїї
 * - виклик функціїї result_root для згенерованого числа змінної n
 * @return повертає(0)
 * @param n число, я якого отримаємо корінь
```

Рисунок 3 – Приклад опису функцій

3.4 Згенеровані описи функцій

Після команди `make doxygen` перейдемо до результуючого файлу (index.html) та побачимо результат (див. рис.4-6).

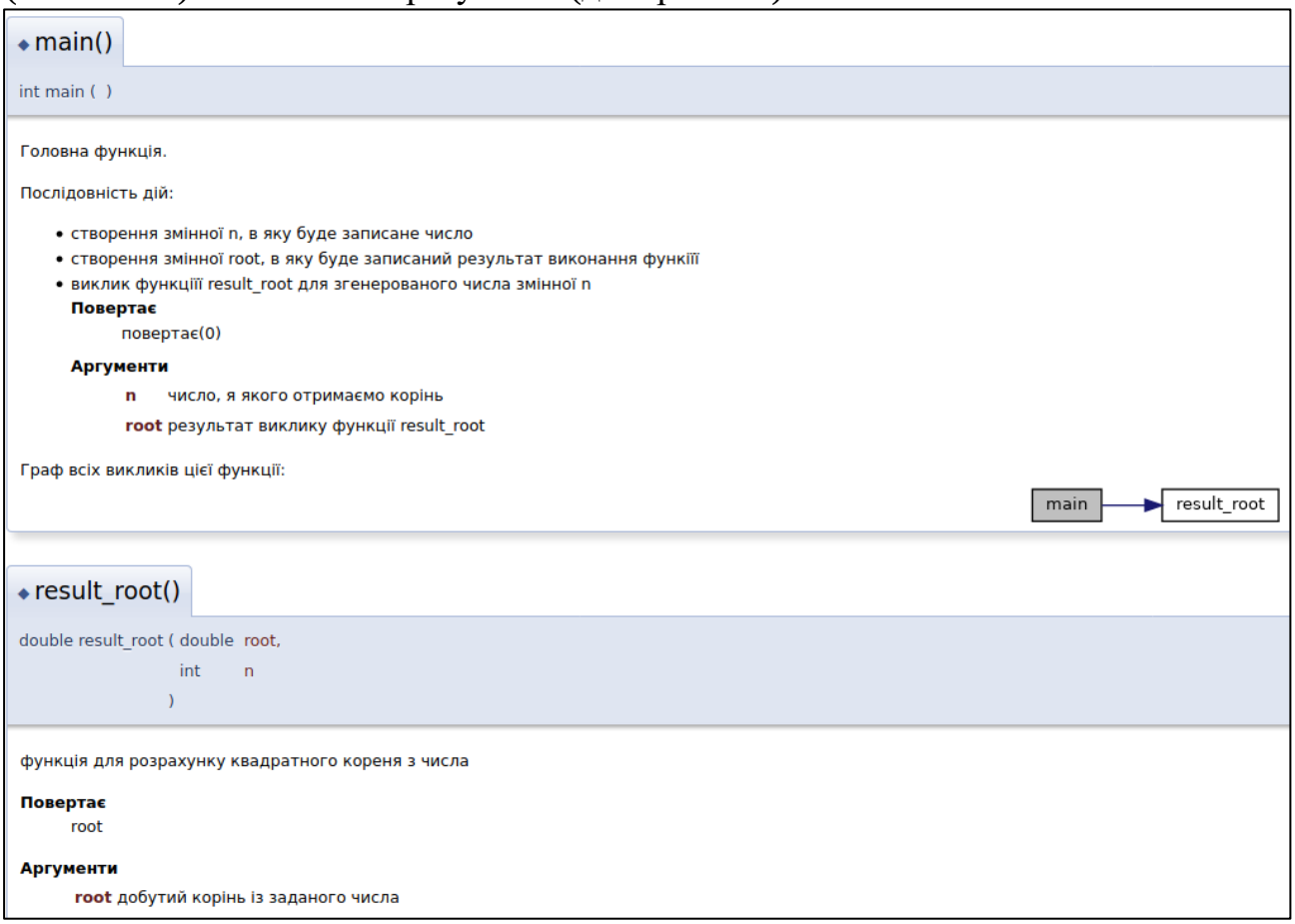


Рисунок 4 – Згенерований опис завдання main_1.c

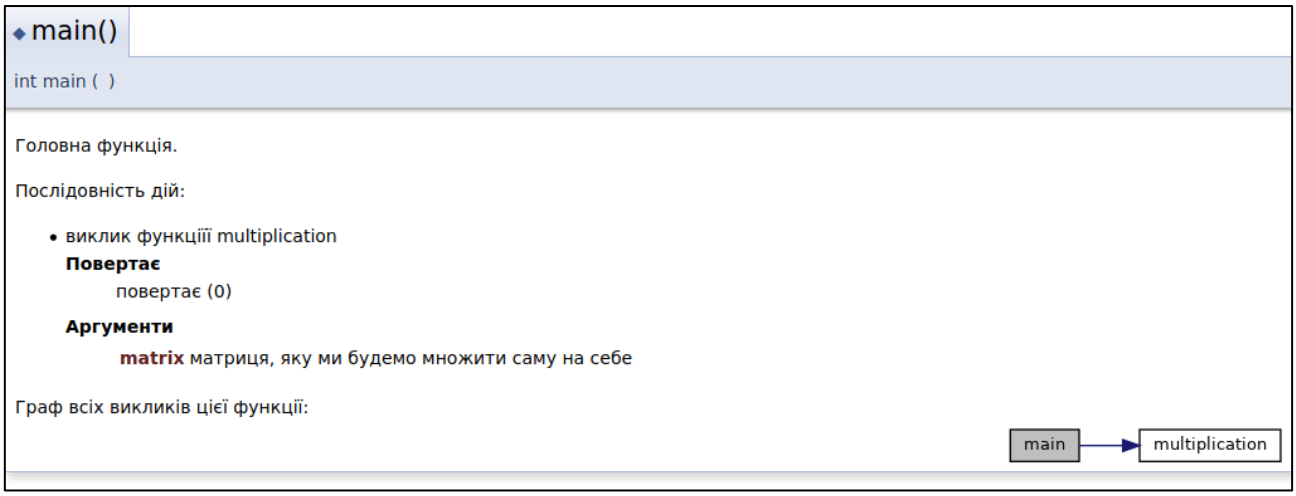
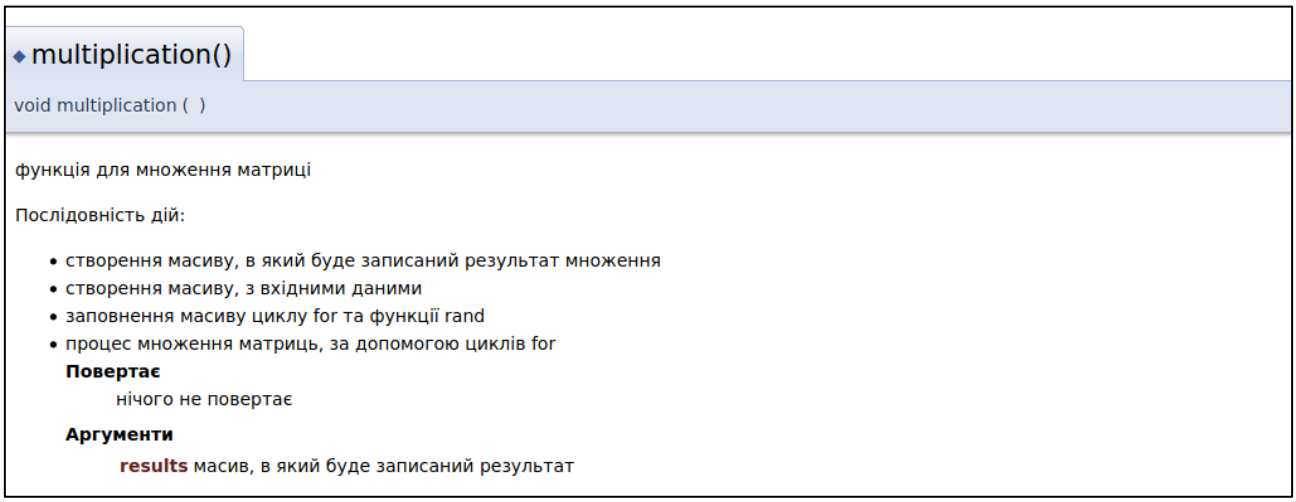


Рисунок 5.1 – Згенерований опис завдання main_2.c



◆ multiplication()

void multiplication ( )

функція для множення матриці

Послідовність дій:

- створення масиву, в який буде записаний результат множення
- створення масиву, з вхідними даними
- заповнення масиву циклу for та функції rand
- процес множення матриць, за допомогою циклів for

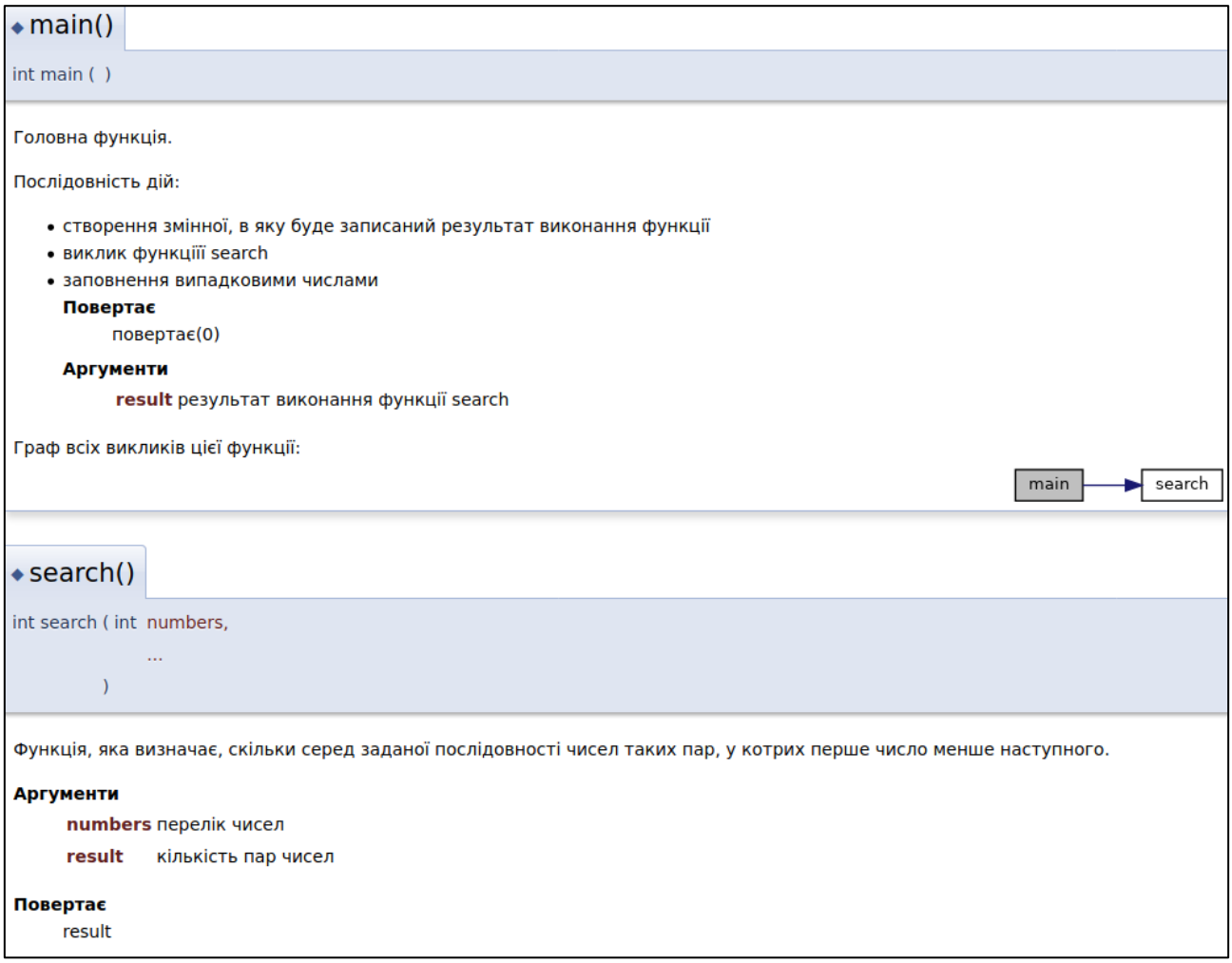
Повертає

нічого не повертає

Аргументи

results масив, в який буде записаний результат

Рисунок 5.2 – Згенерований опис завдання main_2.c



◆ main()

int main ( )

Головна функція.

Послідовність дій:

- створення змінної, в яку буде записаний результат виконання функції
- виклик функціїї search
- заповнення випадковими числами

Повертає

повертає(0)

Аргументи

result результат виконання функції search

Граф всіх викликів цієї функції:

main → search

◆ search()

int search (int numbers,
...
)

Функція, яка визначає, скільки серед заданої послідовності чисел таких пар, у котрих перше число менше наступного.

Аргументи

numbers перелік чисел

result кількість пар чисел

Повертає

result

Рисунок 6 – Згенерований опис завдання main_3.c

Висновки

На цій лабораторній роботі ми дізналися про систему Doxygen, та навчилися нею користуватися, описуючи змінні, та функції. Та генерувати їх в консольному режимі Linux.