**Лабораторна робота № 12, 14. Взаємодія з користувачем шляхом механізму введення/виведення та взаємодія з файлами**

1. ВИМОГИ
   1. Розробник

**** Радєвич Владислав Романович;

**** студент групи КІТ – 320;

**** 19.02.2021 р.

* 1. Загальне завдання

Переробити програми з попередніх робіт, згідно умов та розробити програму, умови якої надано у лабораторному практикуму. Мною було взято умови для розробки з розділу на оцінку «добре».

* 1. Індивідуальне завдання

Зробити звіт за обраним мною варіантом. На даний момент це завдання номер 2 (лабораторна робота №14).

1. ОПИС ПРОГРАМИ
   1. Функціональне призначення

Програма призначена для визначення детермінанта матриці, розмір якої не перевищую 3 на 3. При цьому використовуючи, введення та виведення даних з файлу/ у файл та взаємодія з користувачем за допомогою консолі. Дані про матрицю: а саме: розмір матриці та сама матриця повинні бути записані у файл заздалегідь.

* 1. Опис логічної структури
     1. Основна функція

int main

*Призначення*: головна функція.

*Схема алгоритму функції* подана на рис. 1.

*Опис роботи*: задає розмір масивам, змінним, виділяючи їм пам'ять та викликає функції check\_file, check\_matrix, filling\_matrix, find\_det.

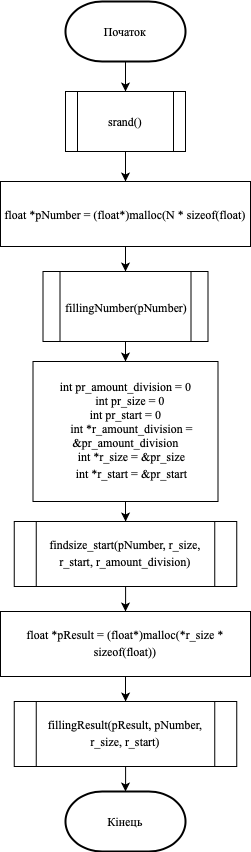


Рисунок 1 – Схема алгоритму функції main

* + 1. Функція перевірки файлу

void check\_file (FILE \*fmatrix);

*Призначення*: перевірка наявності файлу по заданій користувачем директорії.

*Схема алгоритму функції* подана на рис. 2

*Опис роботи*: функція перевіряє наявність файлу з матрицею по заданому користувачем шляху.

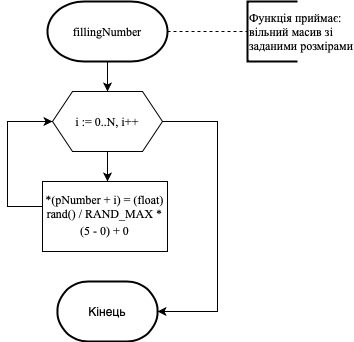


Рисунок 2 – Схема алгоритму функції check\_file

* + 1. Функція перевірки матриці

void check\_matrix (int column, int row, FILE \*fmatrix);

*Призначення*: перевірка чи є матриця квадратною та перевіряє її розмір для знаходження детермінанту.

*Схема алгоритму функції* подана на рис. 3.

*Опис роботи*: функція перевіряє чи є задана матриця квадратною та не перевищує вона обмежений розмір 3 на 3. При виявлення порушення однієї з заданих умов – програма повідоміть про порушення користувача.

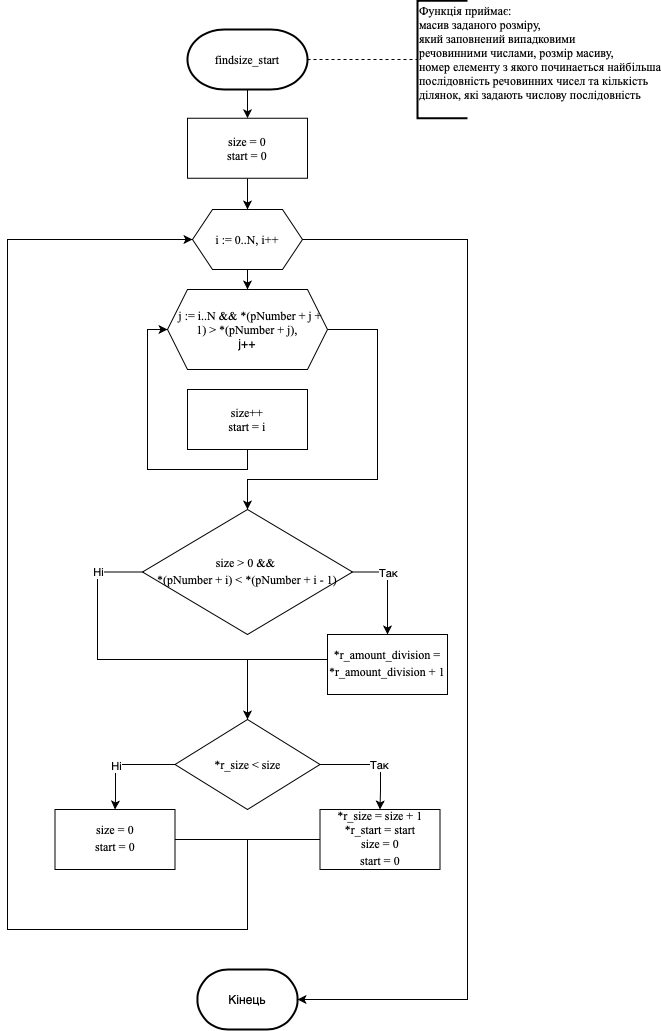


Рисунок 3 – Схема check\_matrix

* + 1. Функція заповнення матриці

void filling\_matrix (int\*\* p\_matrix, int column, int row, FILE \*fmatrix);

*Призначення*: заповнює масив інформацією о матриці, яка записана у файлі.

*Схема алгоритму функції* подана на рис. 4

*Опис роботи*: з файл, шлях якого був указаний користувачем, копіюється інформація про розмір матриці та значення самою матриці.

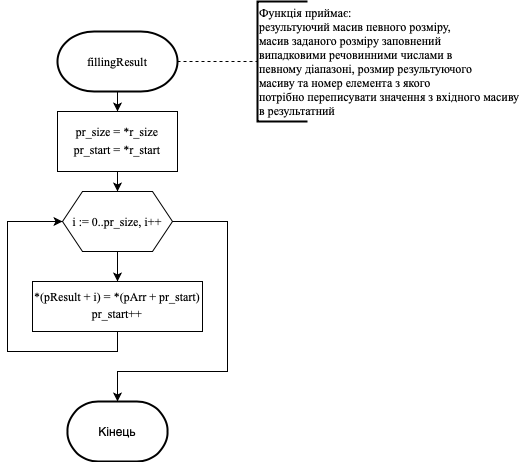


Рисунок 4 – схема алгоритму функції filling\_matrix

* + 1. Функція знаходження детермінанту

int find\_det (int\*\* p\_matrix, int column, FILE \*fmatrix);

*Призначення:* знаходження детермінанту матриці.

*Схема алгоритму функції* подана на рис. 5

*Опис роботи:* функція знаходить детермінант заданої матриці та виводить його значення у консоль і записує значення детермінанту до файлу.

Рисунок 4 – схема алгоритму функції find\_det

* + 1. Функція звільнення пам’яті

void free\_matrix(int\*\* p\_matrix, int column, int row););

*Призначення:* знаходження детермінанту матриці.

*Схема алгоритму функції* подана на рис. 6

*Опис роботи:* функція звільнює пам’ять, що використовувалася при виконання пригорами.

* 1. Структура проекту

.

├── 1

│   ├── README.md

│   └── src

│   ├── lib.c

│   ├── lib.h

│   └── main.c

├── 2

│   ├── README.md

│   └── src

│   ├── lib.c

│   ├── lib.h

│   └── main.c

├── 3

│   ├── README.md

│   └── src

│   ├── lib.c

│   ├── lib.h

│   └── main.c

├── 4

│   ├── README.md

│   └── src

│   ├── lib.c

│   ├── lib.h

│   └── main.c

├── doc

│   ├── assets

│   │   ├── 5.png

│   │   ├── 6.png

│   │   ├── 7.png

│   │   ├── flowchart\_fillingNumber.png

│   │   ├── flowchart\_fillingResult.png

│   │   ├── flowchart\_findsize\_start.png

│   │   └── flowchart\_main.png

│   ├── Lab11.md

│   └── Radievych11.pdf

├── Doxyfile

├── Makefile

└── README.md

1. Варіанти використання

Цю програму можна використовувати за для знаходження детермінанту матриці: максимальний розмір якої може становити 3 на 3.

Результат роботи з doxygen продемонстровано на рисунку 7, рисунку 8 та рисунку 9.

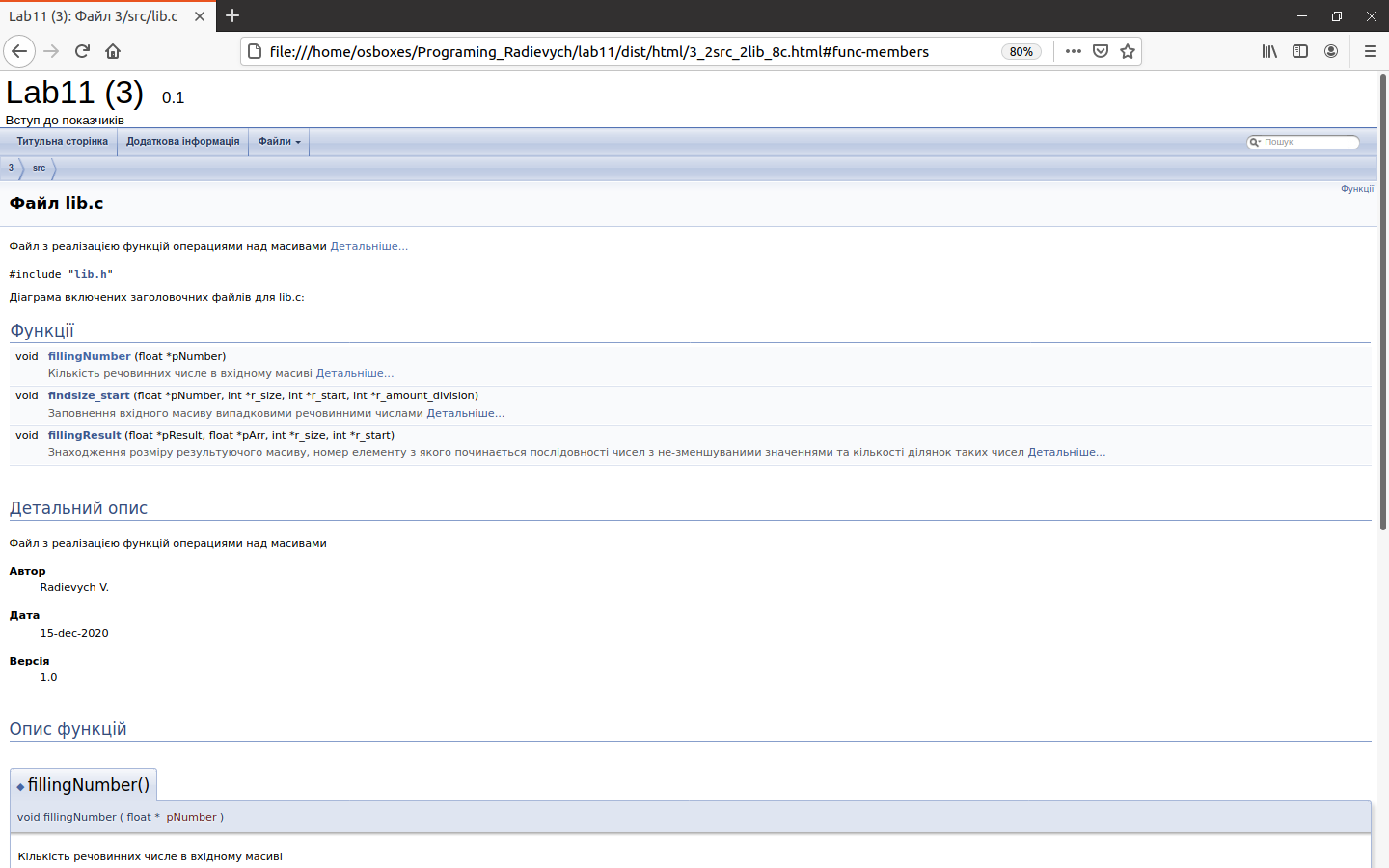


Рисунок 7 – робота з doxygen

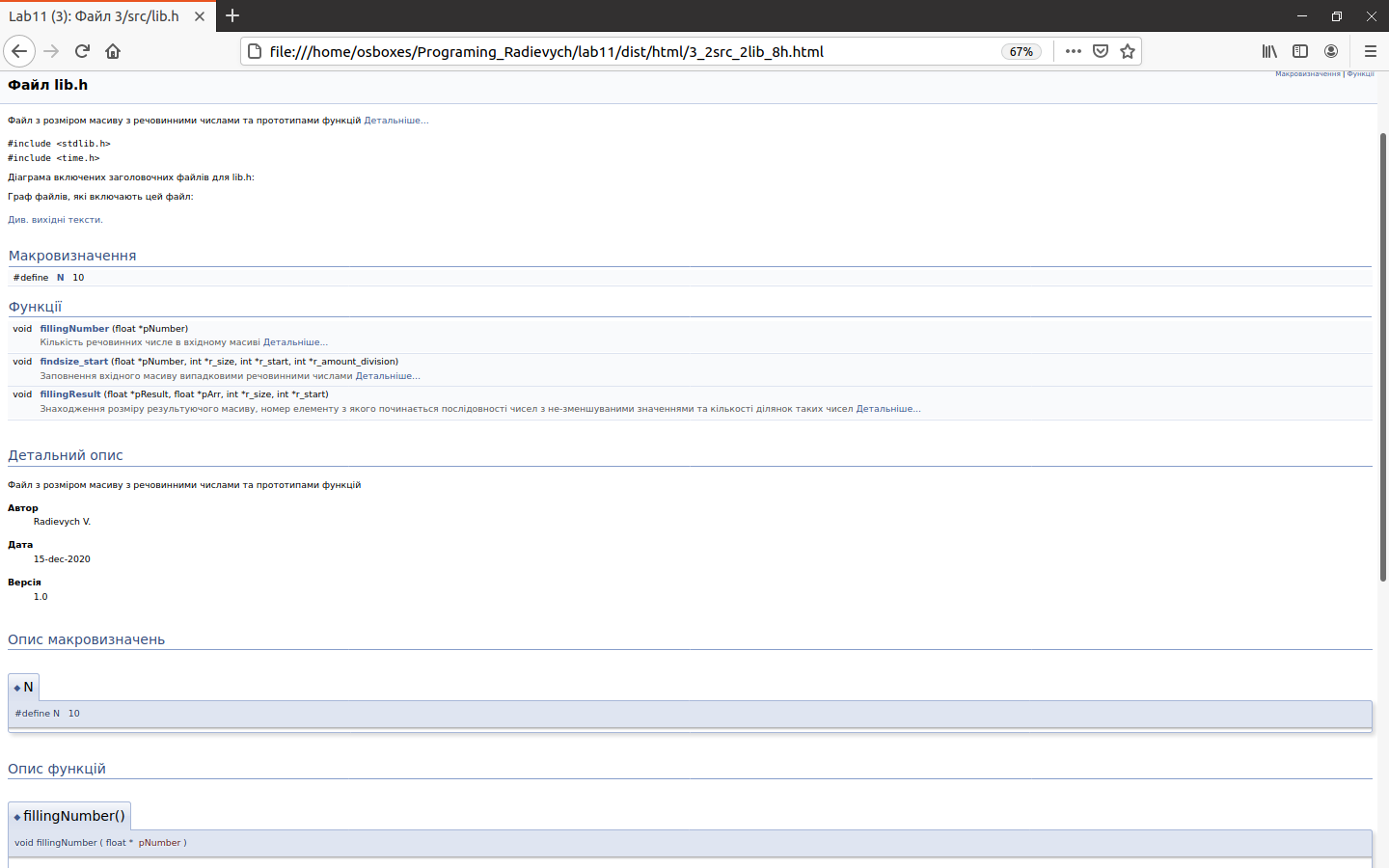


Рисунок 8 – робота з doxygen

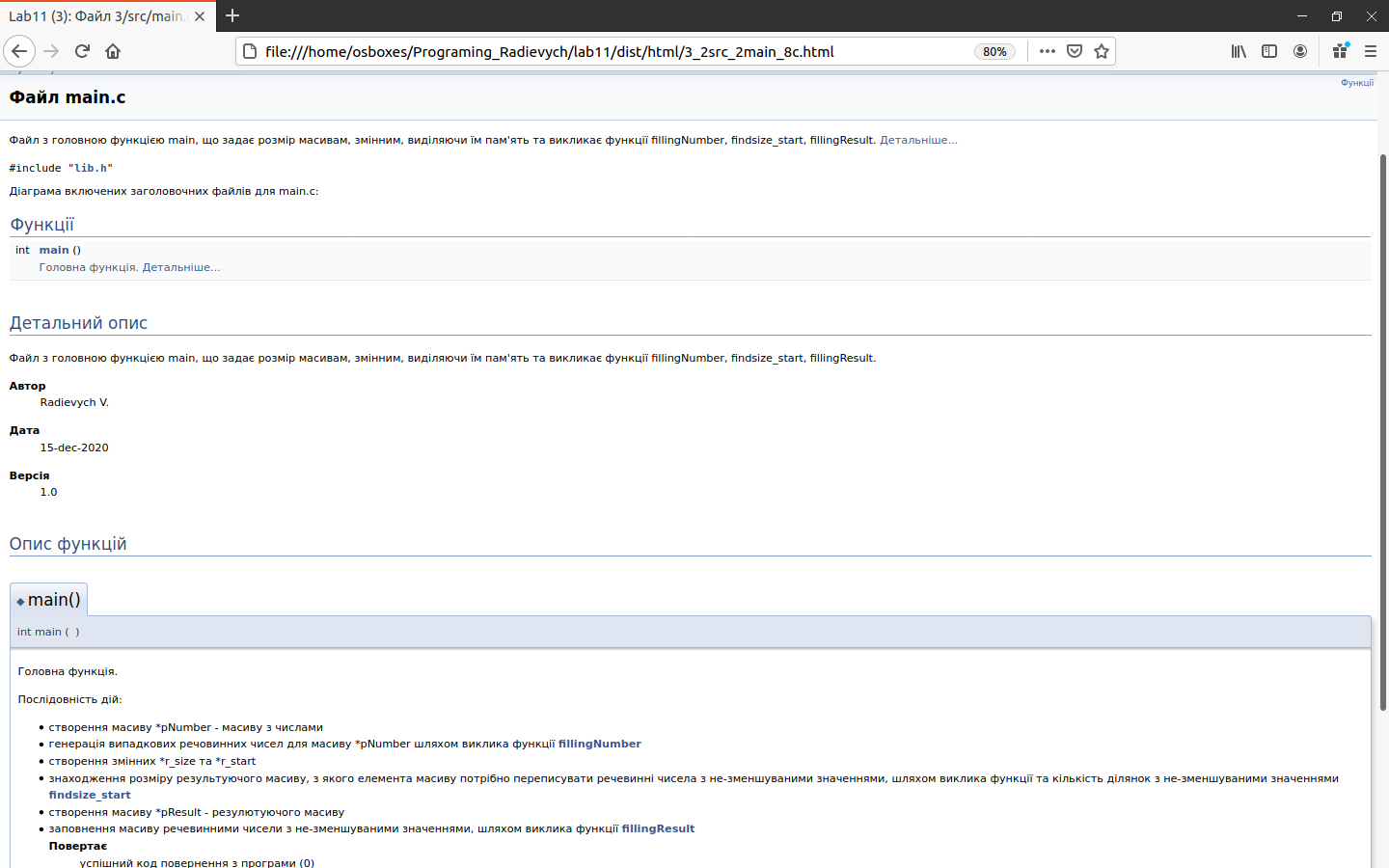


Рисунок 9 – робота з doxygen

1. Висновок

При виконанні даної лабораторної роботи я закріпив набуті мною навички, створення програми, використовуючи взаємодія з користувачем методом введення/виведення та взаємодія з файлами.

Посилання на GitHub, де знаходяться усі програми: <https://github.com/KotKHPI/Programming_Radievych>