**ЛАБОРАТОРНА РОБОТА №11.**

**ВСТУП ДО ПОКАЖЧИКІВ.**

1. **Вимоги**
   1. Розробник

* Бельчинська Катерина Юріївна;
* студентка групи КІТ-320;
* 10-dec-2020.
  1. Загальне завдання

Реалізувати програму з використанням покажчиків.

* 1. Індивідуальне завдання

Дано масив з N речовинних чисел. Підрахувати кількість ділянок, які утворюють безперевні послідовності чисел з незменшуваними значеннями. Максимальну ділянку переписати у інший масив.

1. **Опис програми**
   1. Функціональне призначення

Покажчики доцільно використовувати для працювання не з копією елементу, а безпосередньо з самим елементом, за допомогою його адреси знаходження у пам'яті.

* 1. Опис логічної структури

Функція 'main' виділяє пам'ть для заданого і результуючого масиву, викликає усі функції для заповнення заданного і результуючого масивів, визначення найбільшої незменшуваної послідовності. Схема алгоритму функції наведена на рис. 1.

Функція 'fillArrOne' заповнює заданий масив псевдовипадковими числами. Схема алгоритму функції наведена на рис. 2.

Функція 'countOfIncreasingSequences' рахує кількість незменшуваних послідовностей. Схема алгоритму функції наведена на рис. 3.

Функція 'findMaxIncreasingSequence' шукає найдовшу ділянку. Схема алгоритму функції наведена на рис. 4.

Функція 'fillArrayOut' заповнює результуючий масив. Схема алгоритму функції наведена на рис. 6.

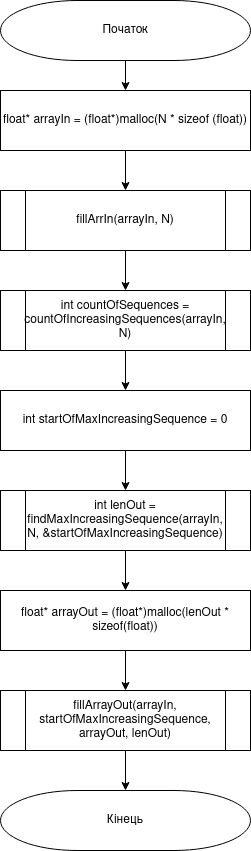


Рис. 1. Схема алгоритму функції main

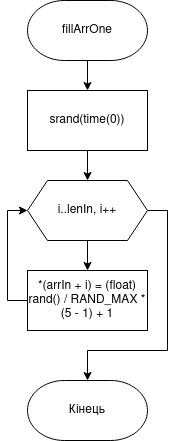
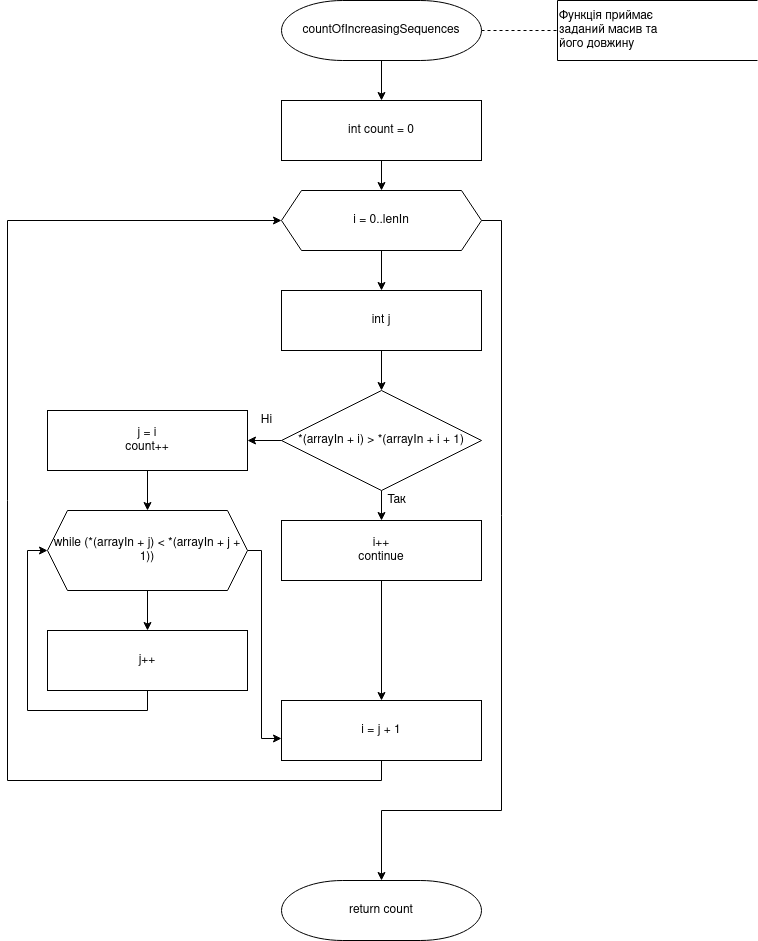
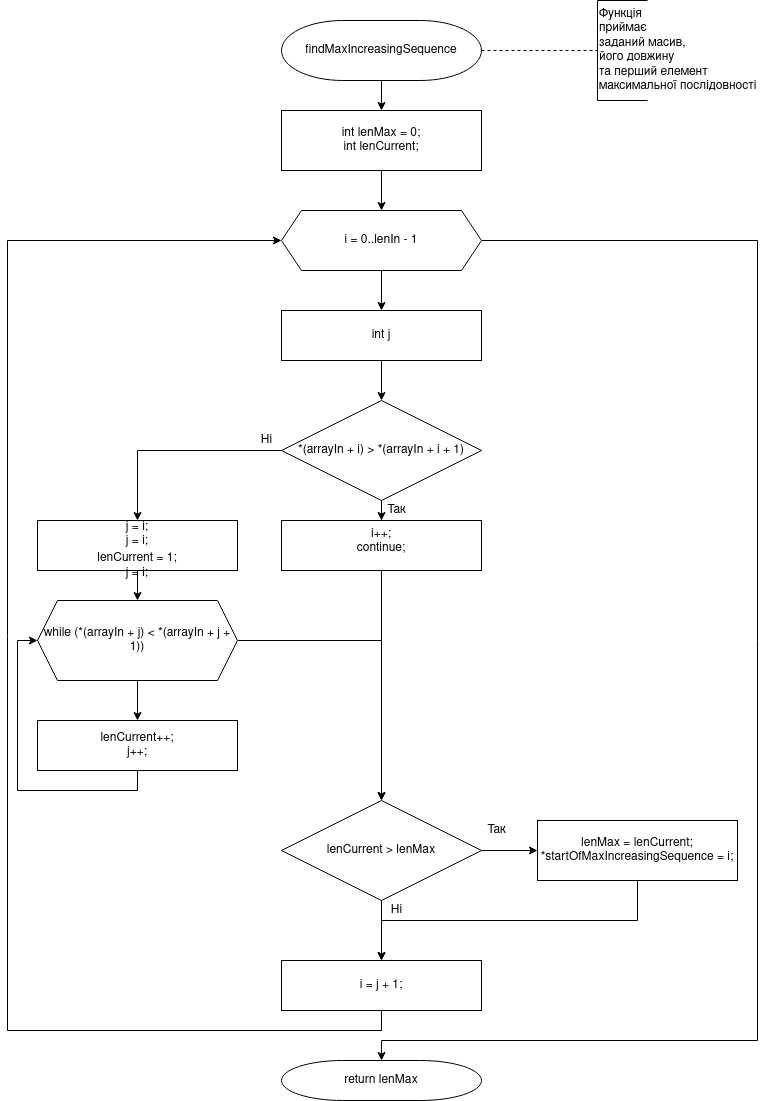


Рис. 2. Схема алгоритму функції fillArrOne

Рис. 3. Схема алгоритму функції countOfIncreasingSequences

Рис. 4. Схема алгоритму функції findMaxIncreasingSequence

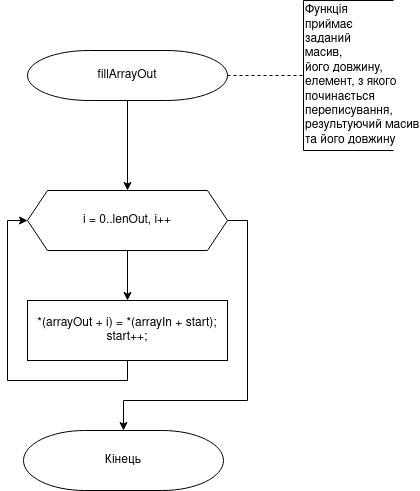


Рис. 5. Схема алгоритму функції fillArrayOut

* 1. Структура проекту

.

├── doc

│   ├── assets

│   │   ├── countOfIncreasingSequences.png

│   │   ├── fillArrayOut.png

│   │   ├── fillArrayOut\_.png

│   │   ├── fillArrOne.png

│   │   ├── findMaxIncreasingSequence.png

│   │   └── main.png

│   ├── lab11.docx

│   ├── lab11.md

│   └── lab11.pdf

├── Doxyfile

├── Makefile

├── README.md

├── task1

│   ├── README.md

│   └── src

│   ├── lib.c

│   ├── lib.h

│   └── main.c

├── task2

│   ├── README.md

│   └── src

│   ├── lib.c

│   ├── lib.h

│   └── main.c

├── task3

│   ├── README.md

│   └── src

│   ├── lib.c

│   ├── lib.h

│   └── main.c

└── task4

├── README.md

└── src

├── lib.c

├── lib.h

└── main.c

* 1. Генерування Doxygen-документації

Doxygen-документація представлена на рисунках 6, 7, 8.

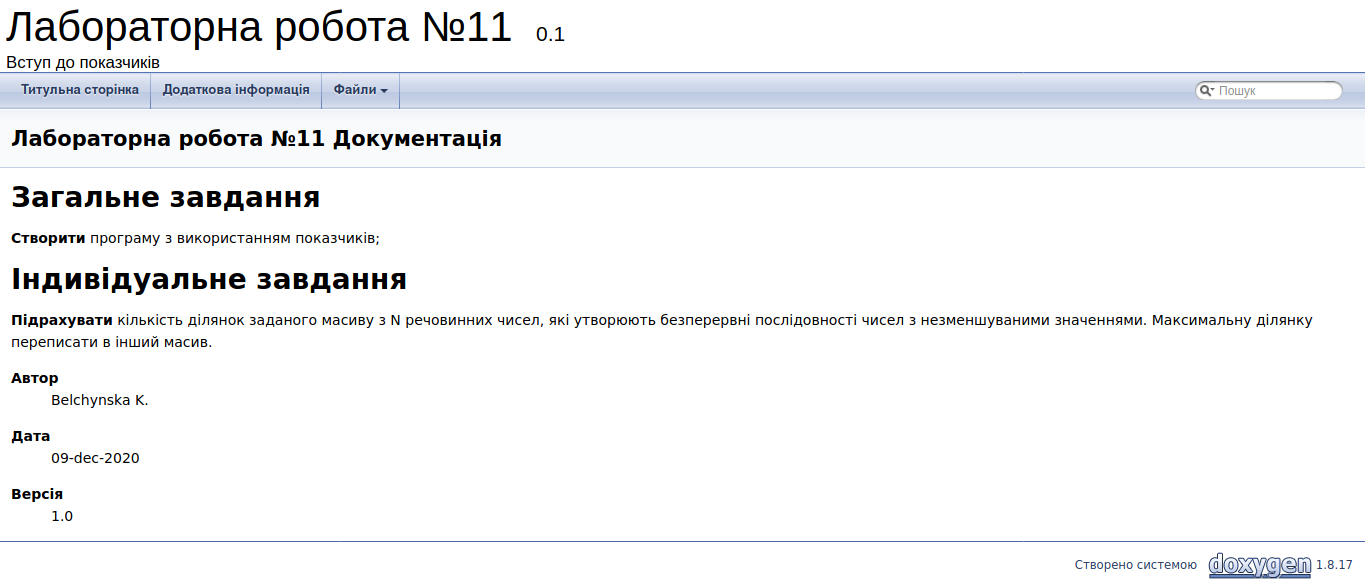
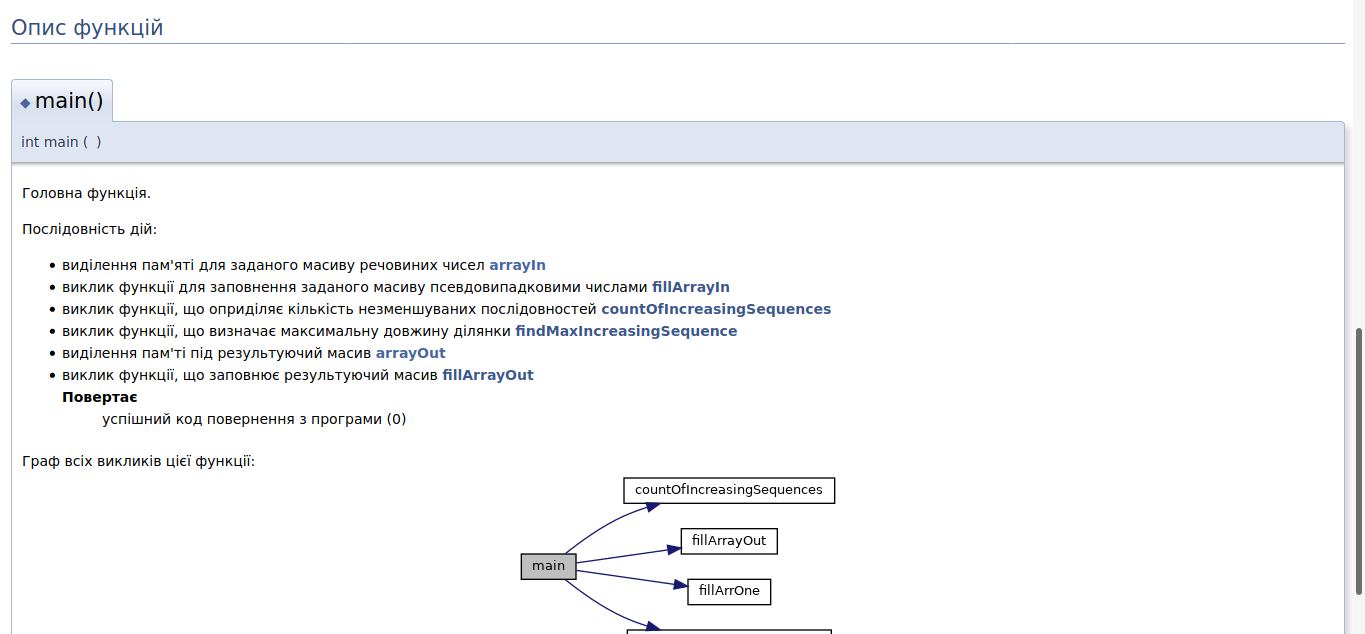


Рис.6 Титульна сторінка Doxygen.

Рис.7 Коментарі до функції main.

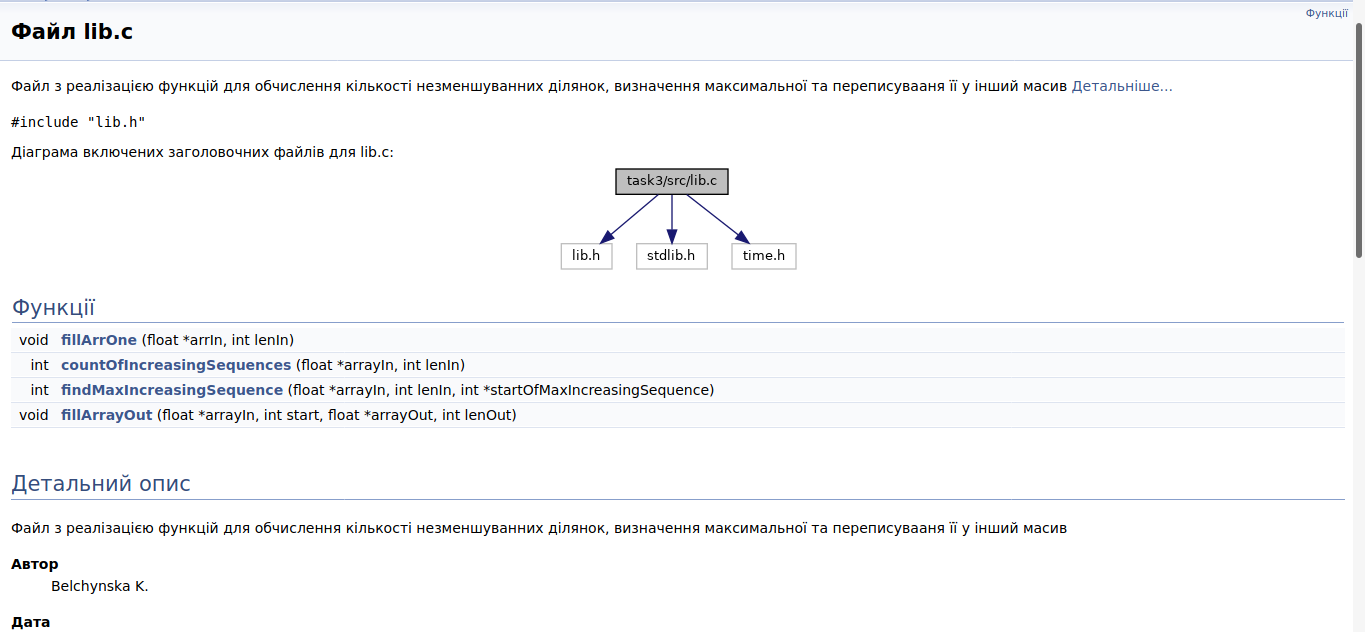


Рис. 8. Коментарі до функцій файлу lib.c

1. **Варіанти використання**

У відлагоджувачі nemiver можна спостерігати за поступовим заповненням результуючого масиву, підрахуванням кількості ділянок з незменшуваним значенням.(рис.9).



Рис. 9. Заповнення результуючого масиву, визначення кількості ділянок

**Висновки**

В ході виконання данної лабораторної роботи були використані покажчики для створення динамічних масивів та для оперування змінними за допомогою їхніх адрес зберігання у пам'яті.