**Лабораторна робота №11. Вступ до показчиків**

**Мета:** розробити програму із застосуванням показчиків.

**1 Вимоги**

**1.1 Розробник:**

Мосійчук Артем Михайлович

Студент групи КІТ-120а

07.01.2021

**1.2 Загальне завдання**

Розробити програму із застосуванням показчиків.

**1.3 Індивідуальне завдання**

1. Розробити функцію, яка буде визначати, чи є в масиві елементи, що повторюються, та скільки разів, які елементи повторюються.
2. Розробити функцію, яка буде знаходити безперервну послідовність позитивних чисел у вхідному масиві.
3. Розробити функцію, яка буде підраховувати кількість ділянок, які утворюють безперервні послідовності речовинних чисел.
4. Розробити функцію, яка буде записувати елементи головної діагоналі матриці в одномірний масив за зростанням.

**2 Опис програми**

**2.1 Функціональне призначення:**

1. Перша задача розроблена для визначення елементів масива, що повторюються, та скільки разів, які елементи повторюються.
2. Друга задача розроблена для знаходження безперервної послідовності позитивних чисел.
3. Третя задача розроблена для підрахування кількості ділянок, які утворюють безперервні послідовності речовинних чисел.
4. Четверта задача розроблена для виписування елементів головної діагоналі матриці в одномірний масив.
   1. **Опис логічної структури**

Структура проекту (див. рис. 1).

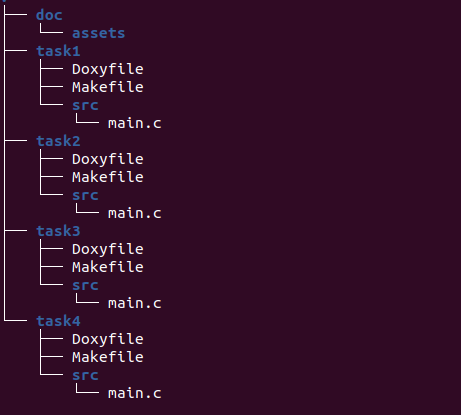
****

Рисунок 1 – структура проекту.

1. Створюємо вхідний масив, та заповнюємо його за допомогою функції fill\_initial\_array. Створюємо масив того ж розміру, що й вихідний, та за допомогою функції fill\_array\_of\_repeats заповнюємо його елементами, які вказують на кількість повторів відповідного елемента у вхідному масиві. Потім за допомогою функції size\_of\_result дізнаємось розмір результуючого масиву та створюємо його. Далі за допомогою функції fill\_result\_array заповнюємо результуючий масив необхідними елементами.

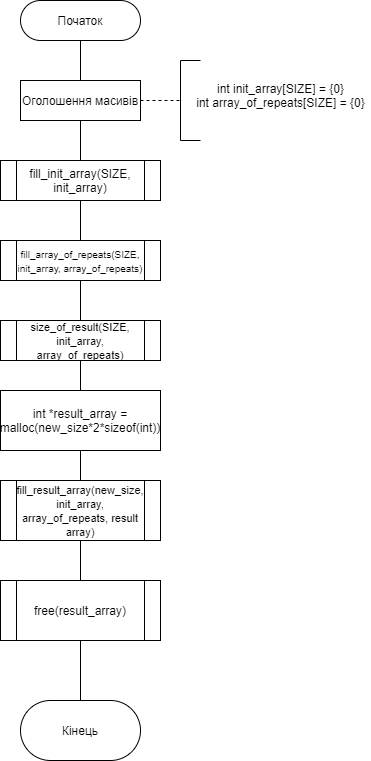


Рисунок 2 — Схема алгоритму функції main.

2. Функція, що заповнює вхідний масив fill\_init\_array. Заповнює вхідний масив випадковими числами від 1 до 5. Схема алгоритму функції подана на рис. 3.

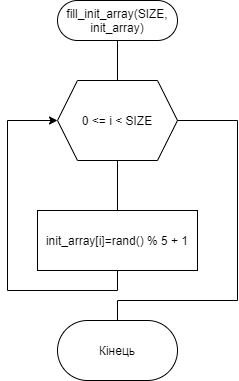


Рисунок 3 — Схема алгоритму функції fill\_init\_array

3. Функція, що заповнює масив повторів fill\_array\_of\_repeats. Заповнює створений масив кількістю повторів кожного елементу вхідного масиву відповідно. Схема алгоритму функції подана на рис. 4.

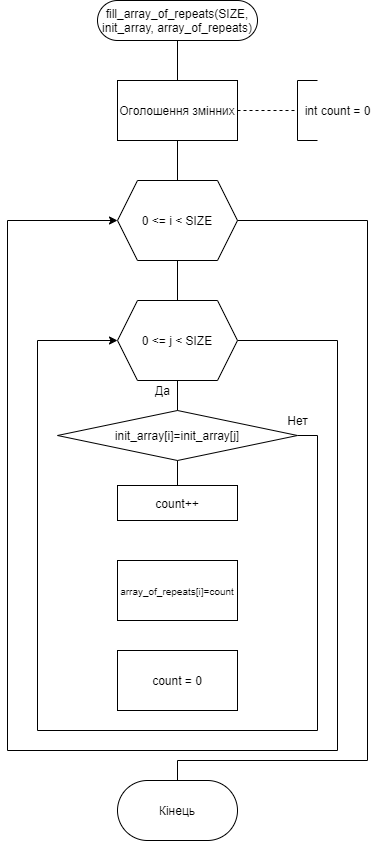


Рисунок 4 — Схема алгоритму функції fill\_array\_of\_repeats.

4. Функція, що рахує розмір результуючого самиву size\_of\_result. Фнукція рахує розмір результуючого масиву. Переметри: new\_size - розмір результуючого масиву, flag- змінна для перевірки умови. Функція повертає new\_size. Схема алгоритму функції подана на рис. 5.

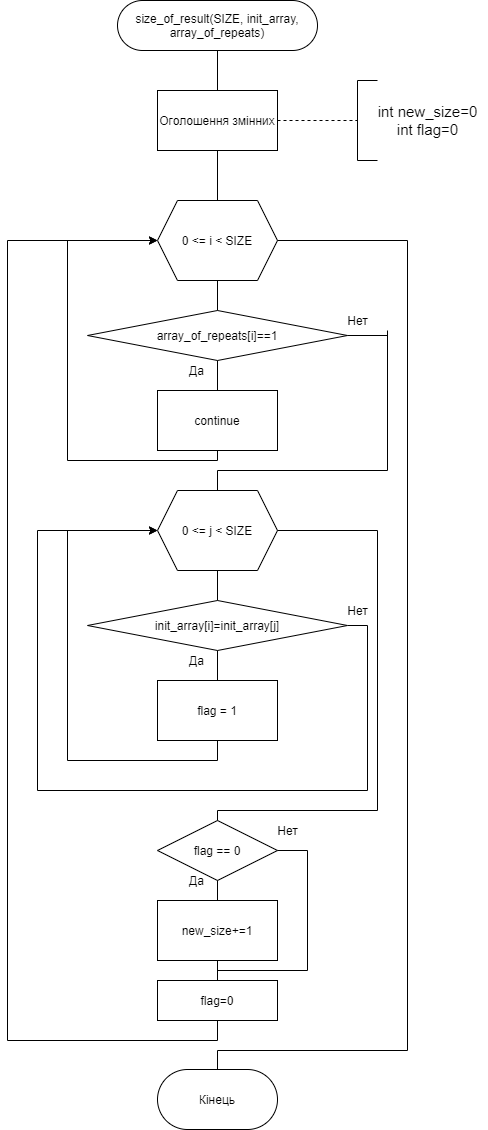


Рисунок 5 — Схема алгоритму функії size\_of\_result.

5. Функція, заповнює результуючий масивfill\_result\_array. Функція заповнює результуючий масив необхідними елементами. Переметри: k- змінна для правильного заповнення результуючого масиву, flag- змінна для перевірки умови. Схема алгоритму функціїподана на рис. 6.

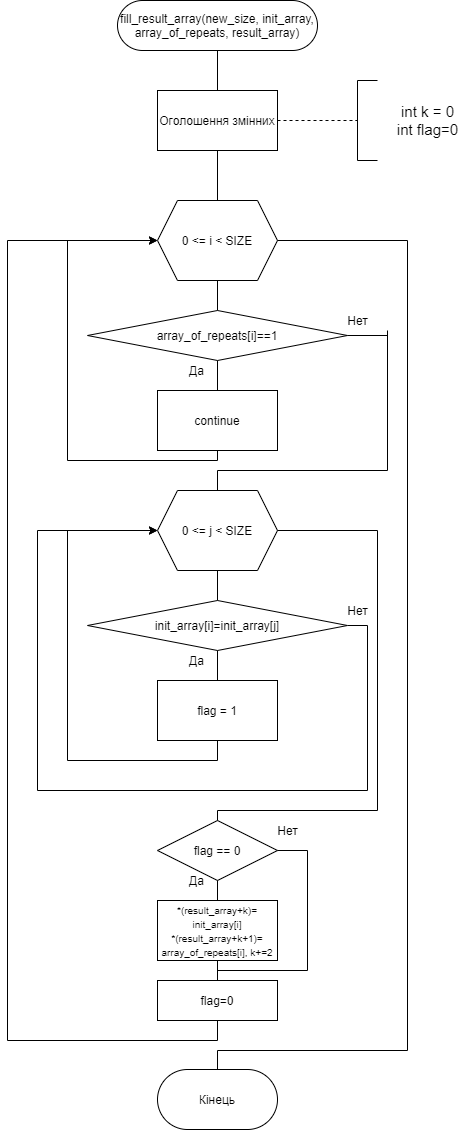


Рисунок 6 — Схема алгоритму функції fill\_result\_array.

* 1. **Важливі фрагменти програми**

Генеруємо *Doxyfile*  (див. рис. 7).

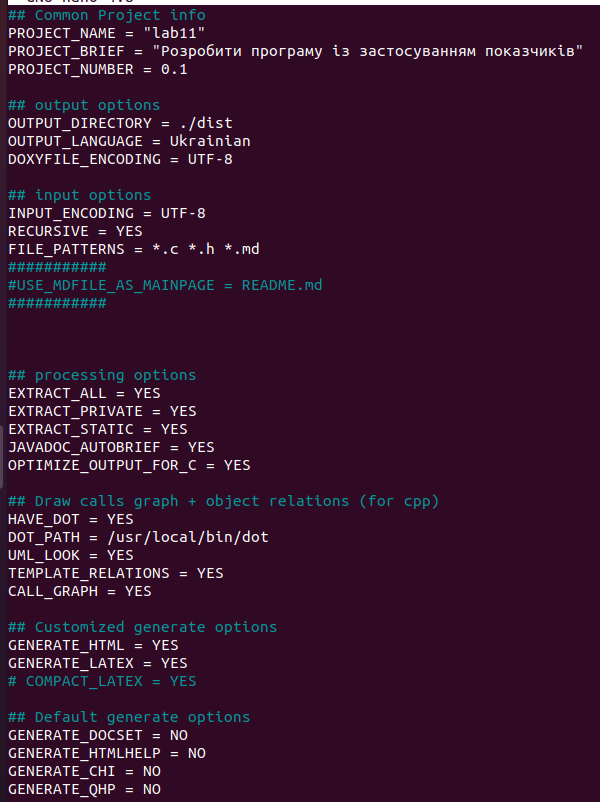


Рисунок 7 – згенерований Doxyfile.

Запускаємо Doxygen за допомогою виклику *dist/html => firefox index.html* (див. рис. 8)



Рисунок 8 – демонстрація документації Doxygen.

1. **Варіанти використання**

Демонструємо результат виконання програми.

**Задача №1** (див. рис. 13).



Рисунок 9 – результат виконання задачі №1.

**Висновок**

У ході виконання лабораторної роботи було набуто практичного досвіду

із розробки програм з показчиками.