

# Лабораторна робота № 11. Вступ до показчиків

## 1 Вимоги

### 1.1 Розробник

- Радєвич Владислав Романович;
- студент групи КІТ – 320;
- 15.12.2020 р.

### 1.2 Загальне завдання

Розробити програми, умови яких надано у лабораторному практикуму. Мною було взято умови для розробки з розділу на оцінку «відмінно».

### 1.3 Індивідуальне завдання

Зробити звіт за обраним мною варіантом. На даний момент це завдання номер 3.

## 2 Опис програми

### 2.1 Функціональне призначення

Програма призначена для визначення кількості ділянок, які утворюють безперервні послідовності чисел з не-зменшуваними значеннями та знаходження найбільшої ділянки з такими числами, використовуючи показники. При цьому значення найбільшої ділянки будуть записані в окремий масив.

### 2.2 Опис логічної структури

#### Основна функція

```
int main
```

*Призначення:* головна функція

*Схема алгоритму функції* подана на рис. 1

*Опис роботи:* задає розмір масивам, змінним, виділяючи їм пам'ять. Викликає функції `fillingNumber`, `findsize_start`, `fillingResult`.

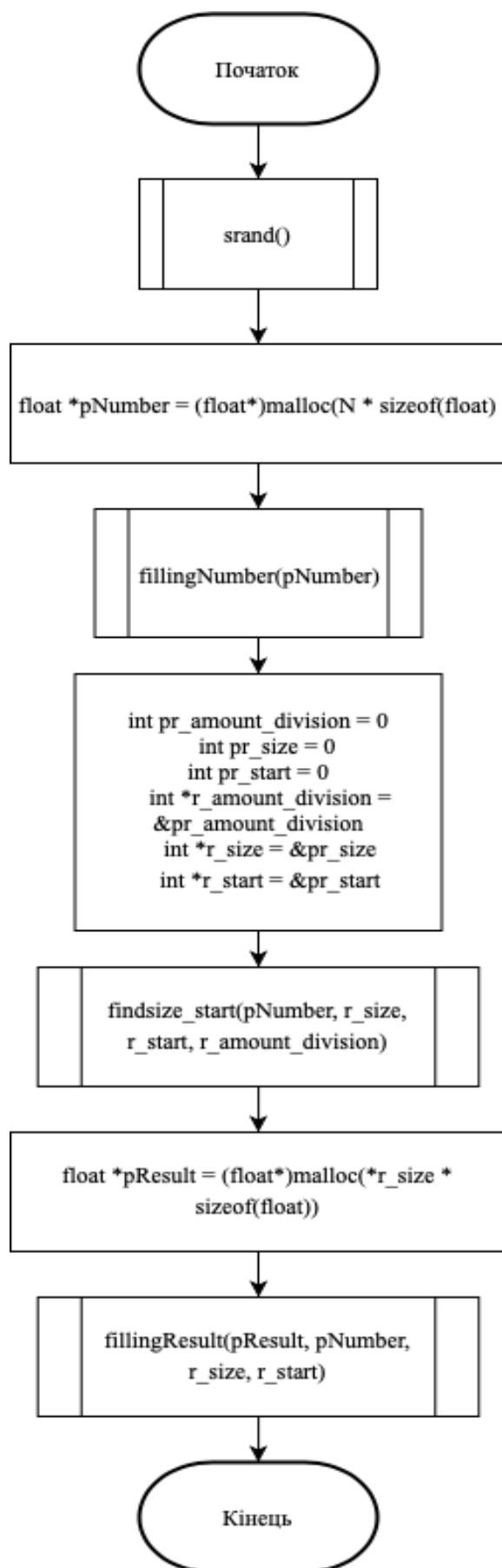


Рисунок 1 – схема алгоритму функції main

## Функція заповнення масиву

```
void fillingNumber(float *pNumber);
```

*Призначення:* заповнення масиву випадковими речовинними числами.

*Схема алгоритму функції* подана на рис. 2

*Опис роботи:* кожному елементу масиву, які перебираються за допомогою циклу, присвоюється випадкове значення завдяки функції rand().

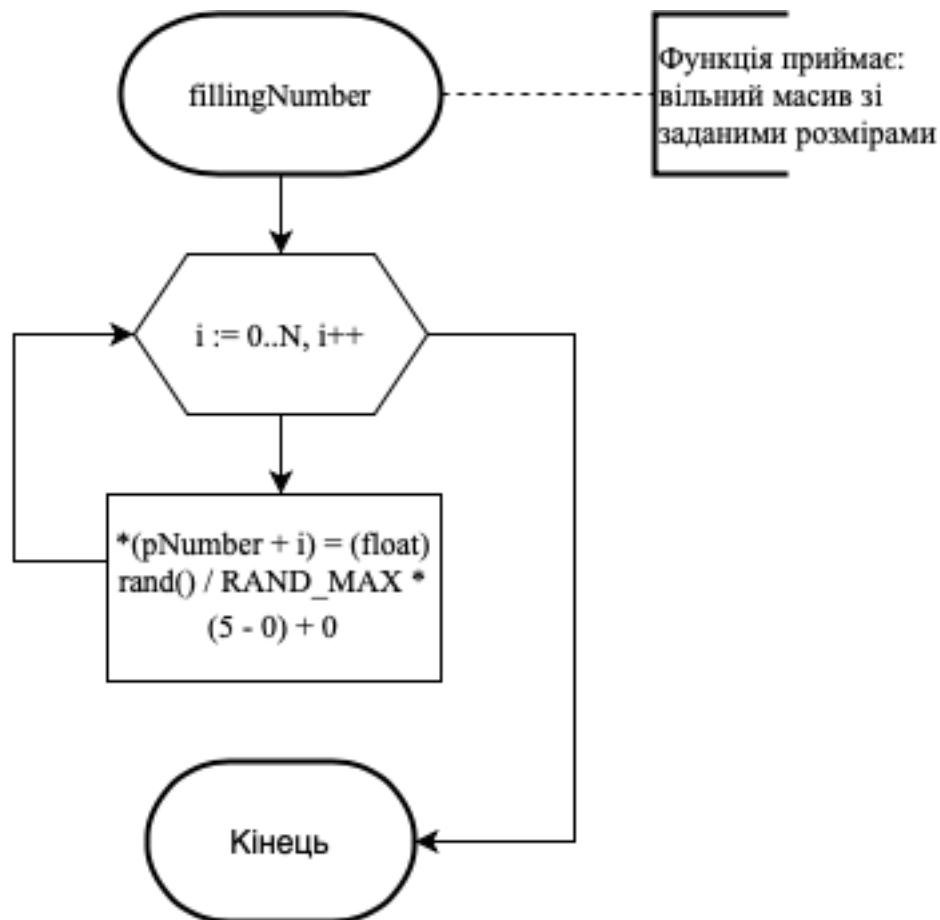


Рисунок 2 – схема алгоритму функції `fillingNumber`

## Функція визначення розміру та кількості значень результуючого масиву та кількості з не-зменшуваними речовинними числами

```
void findsize_start(float *pNumber, int *r_size, int *r_start, int *r_amount_division);
```

*Призначення:* знаходження розміру результуючого масиву, номер елемента з якого починається послідовності чисел з не-зменшуваними

значеннями та кількість ділянок з не-зменшуваними речовинними числами.

*Схема алгоритму функції* подана на рис. 3

*Опис роботи:* функція знаходить кількість ділянок з не-зменшуваними речовинними числами, розмір максимальної ділянки та елемент масиву з якого починається максимальна ділянка з не-зменшуваними значеннями.

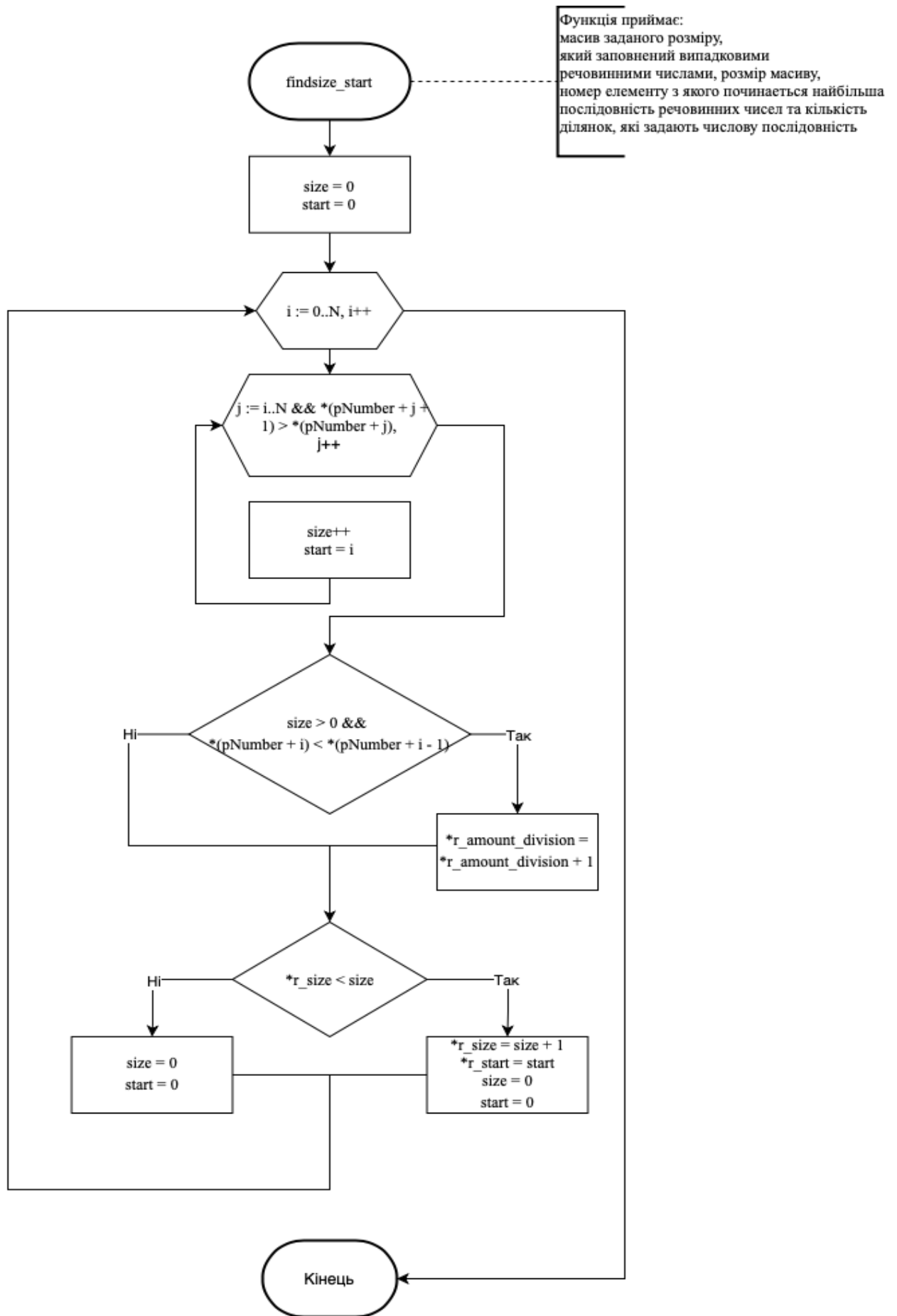


Рисунок 3 – схема findsize\_start

## Функція заповнення результатного масиву

```
void fillingResult(float *pResult, float *pArr, int *r_size, int  
*r_start);
```

*Призначення:* заповнення результуючого масиву числами з не-зменшуваними значеннями, послідовність, яких найбільша за розміром в масиві.

*Схема алгоритму функції* подана на рис. 4

*Опис роботи:* кожному елементу масиву присвоюється значення вхідного масиву, починаючи зі знайденого початку та упродовж знайденої кількості таких чисел.

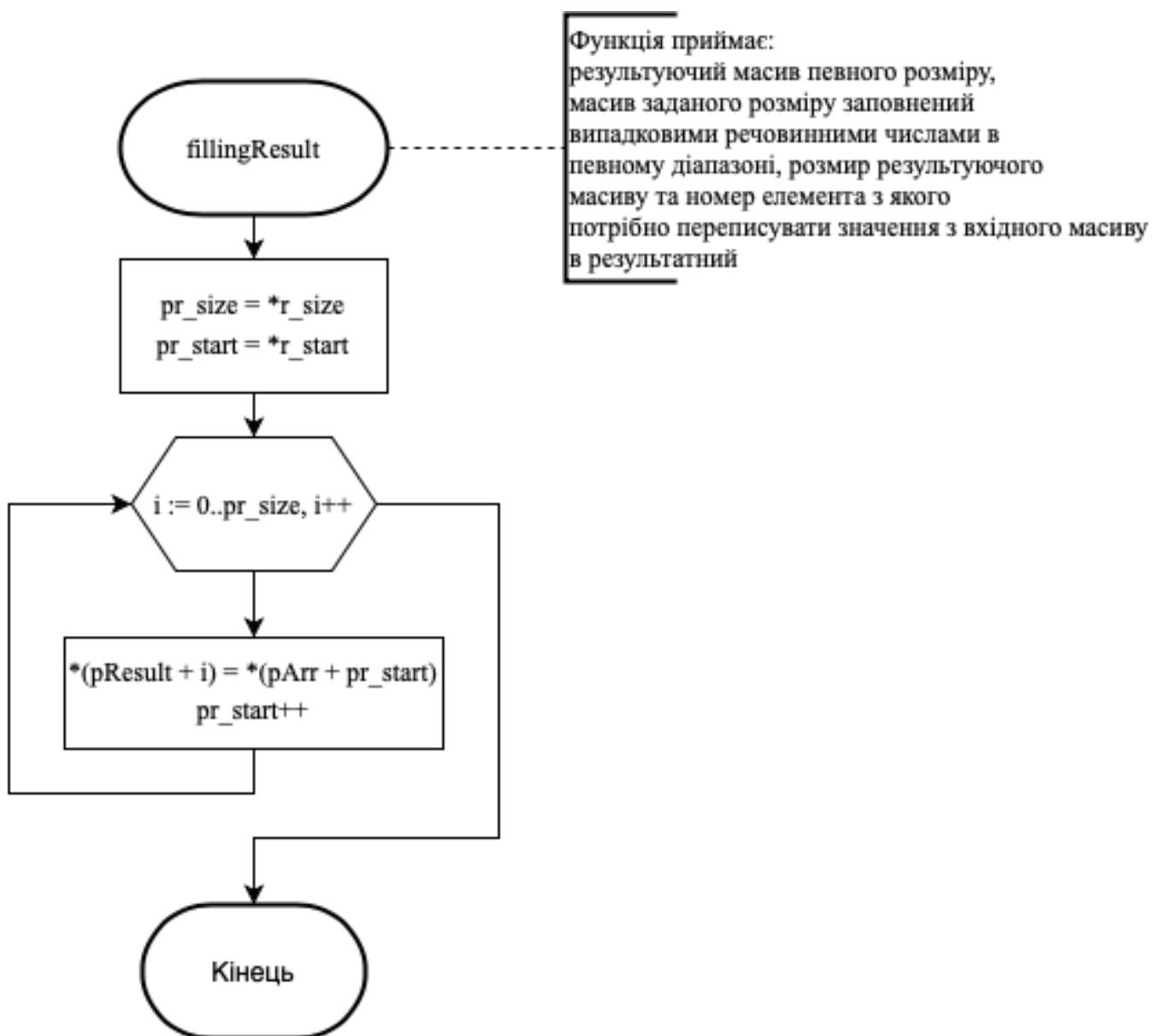
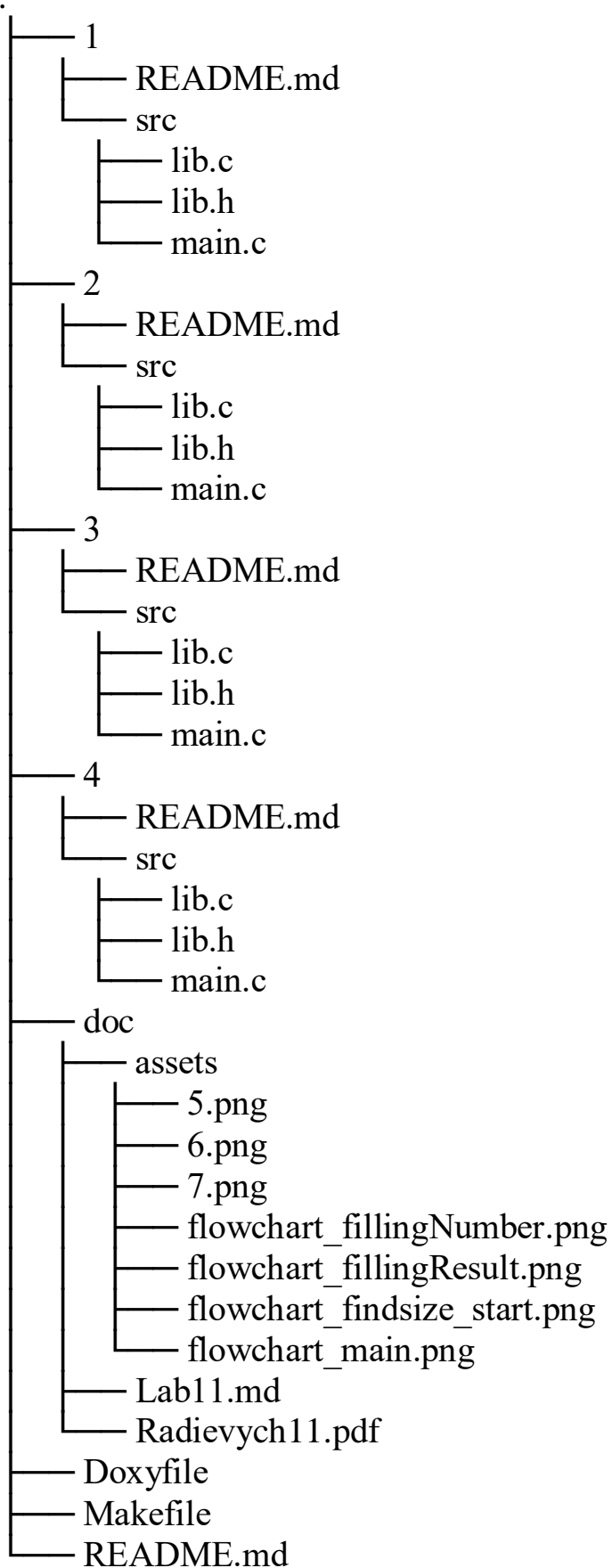


Рисунок 4 – схема алгоритму функції fillingResult

## Структура проекту



## 3 Варіанти використання

Цю програму можна використовувати за для знаходження максимальної кількості послідовних не-зменшуваних чисел в заданому масиві та знаходження кількості таких послідовностей.

Результат роботи з doxygen продемонстровано на рисунку 5, рисунку 6 та рисунку 7.

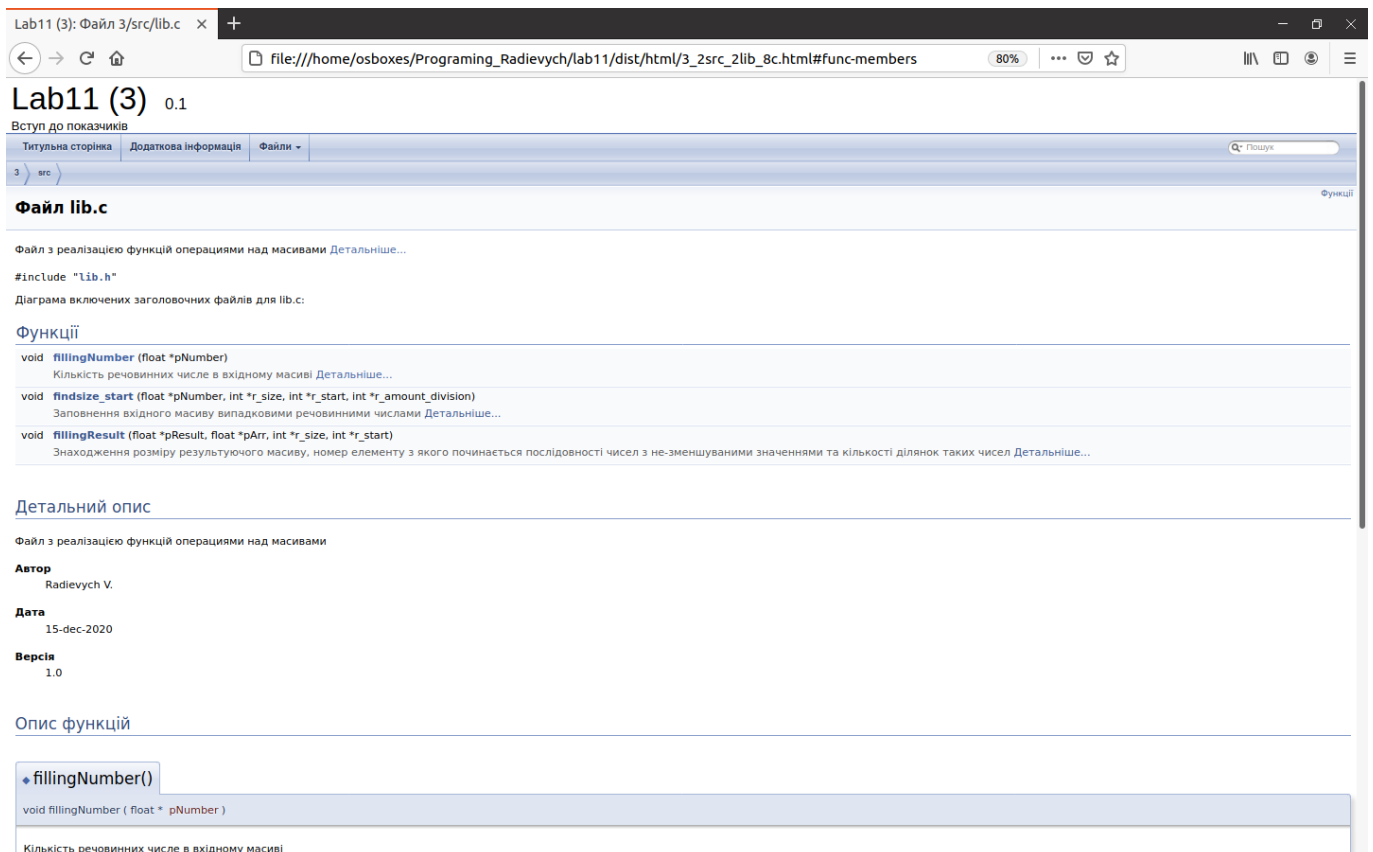


Рисунок 5 – робота з doxygen



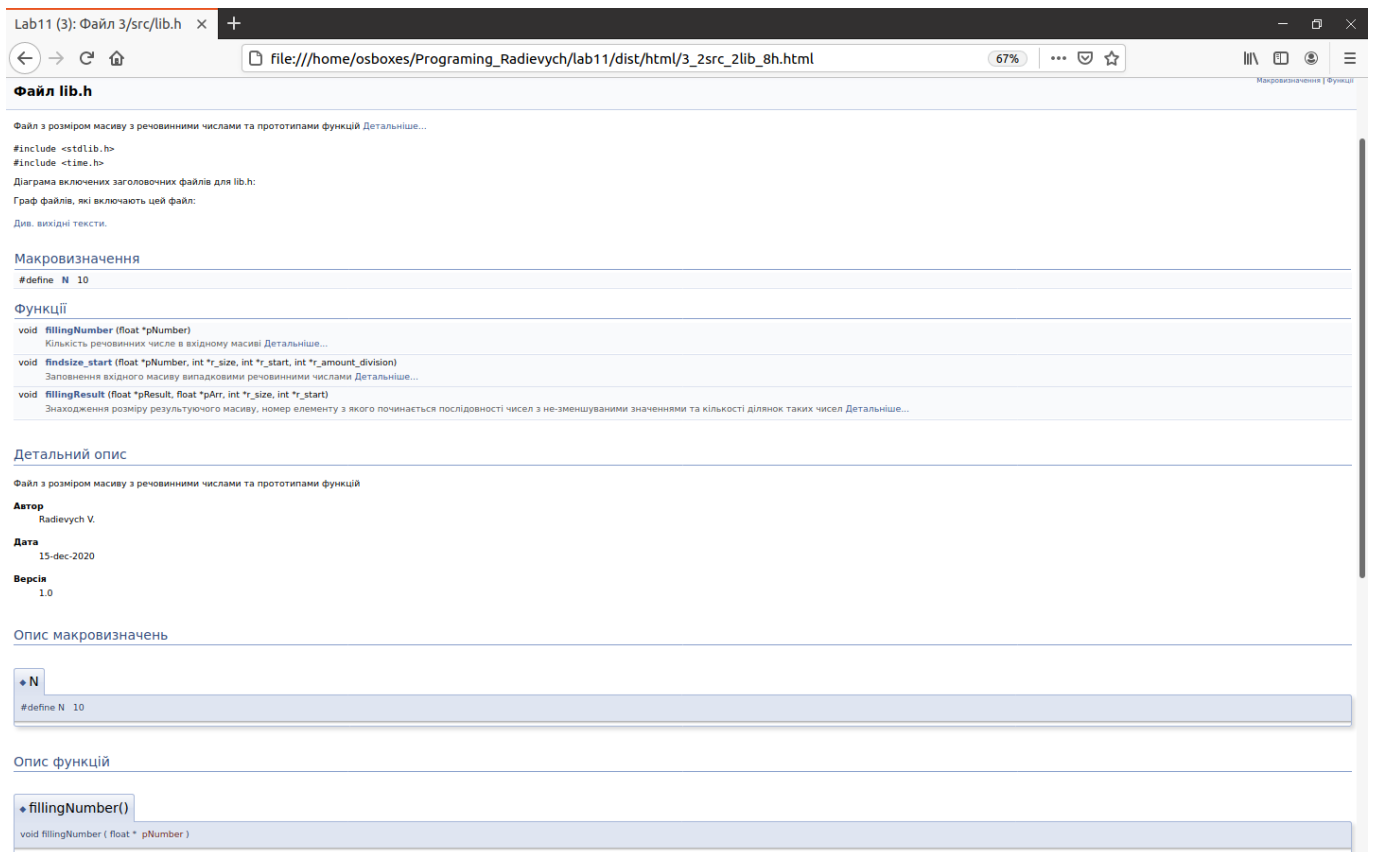


Рисунок 6 – робота з doxygen

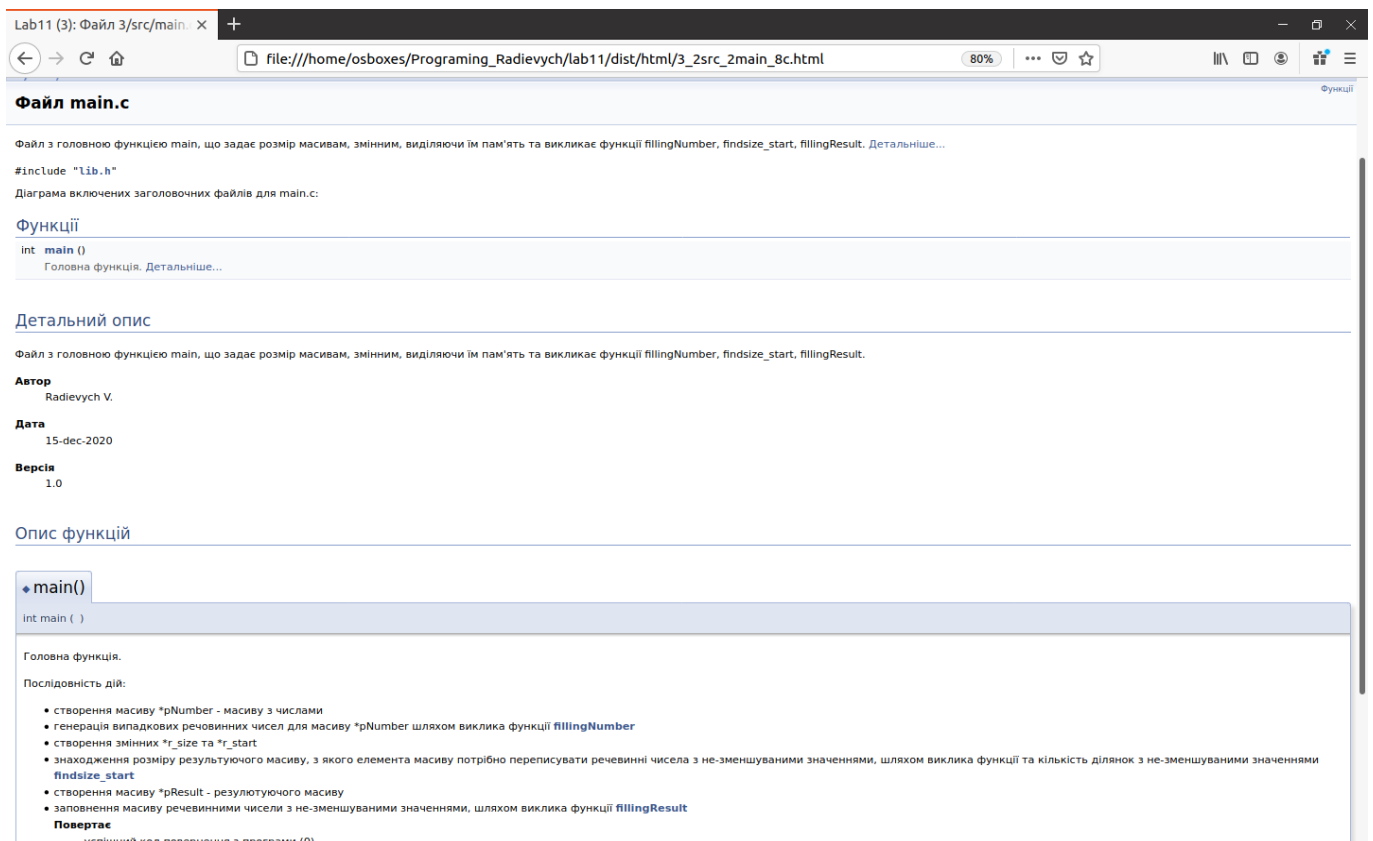


Рисунок 7 – робота з doxygen

## **Висновок**

При виконанні даної лабораторної роботи я закріпив набуті мною навички, створення програми, використовуючи показчики.

Посилання на GitHub, де знаходяться усі програми:

[https://github.com/KotKHPI/Programming\\_Radievych](https://github.com/KotKHPI/Programming_Radievych)

