

ЛАБОРАТОРНА РОБОТА №11.

ВСТУП ДО ПОКАЖЧИКІВ.

1. Вимоги

1.1. Розробник

- Бельчинська Катерина Юріївна;
- студентка групи КІТ-320;
- 10-dec-2020.

1.2. Загальне завдання

Реалізувати програму з використанням покажчиків.

1.3. 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.

2. Опис програми

2.1. Функціональне призначення

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

2.2. Опис логічної структури

Функція 'main' виділяє пам'ть для заданого і результуючого масиву, викликає усі функції для заповнення заданого 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.

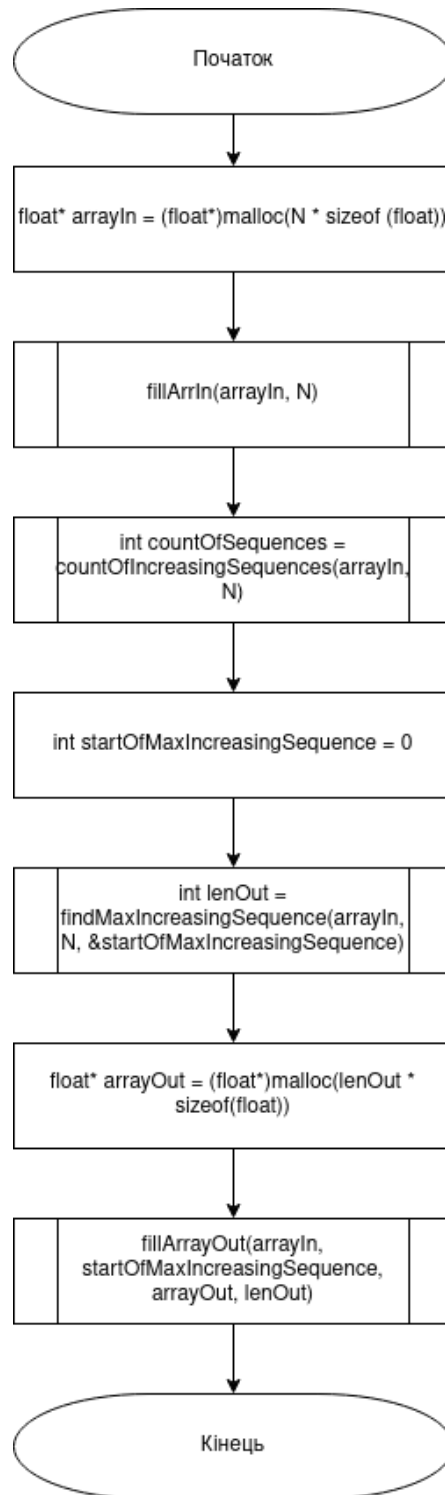


Рис. 1. Схема алгоритму функції main

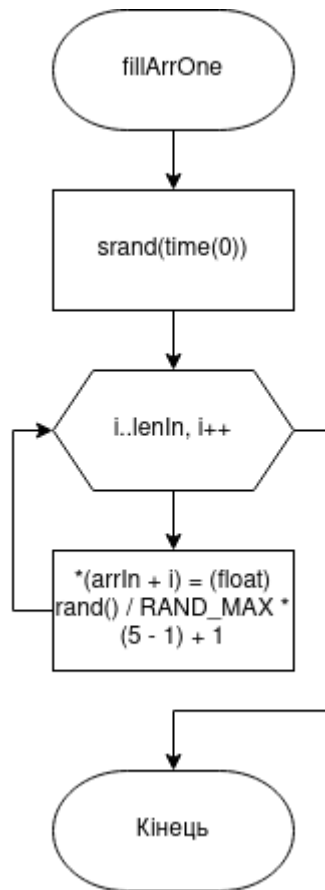


Рис. 2. Схема алгоритму функції `fillArrOne`

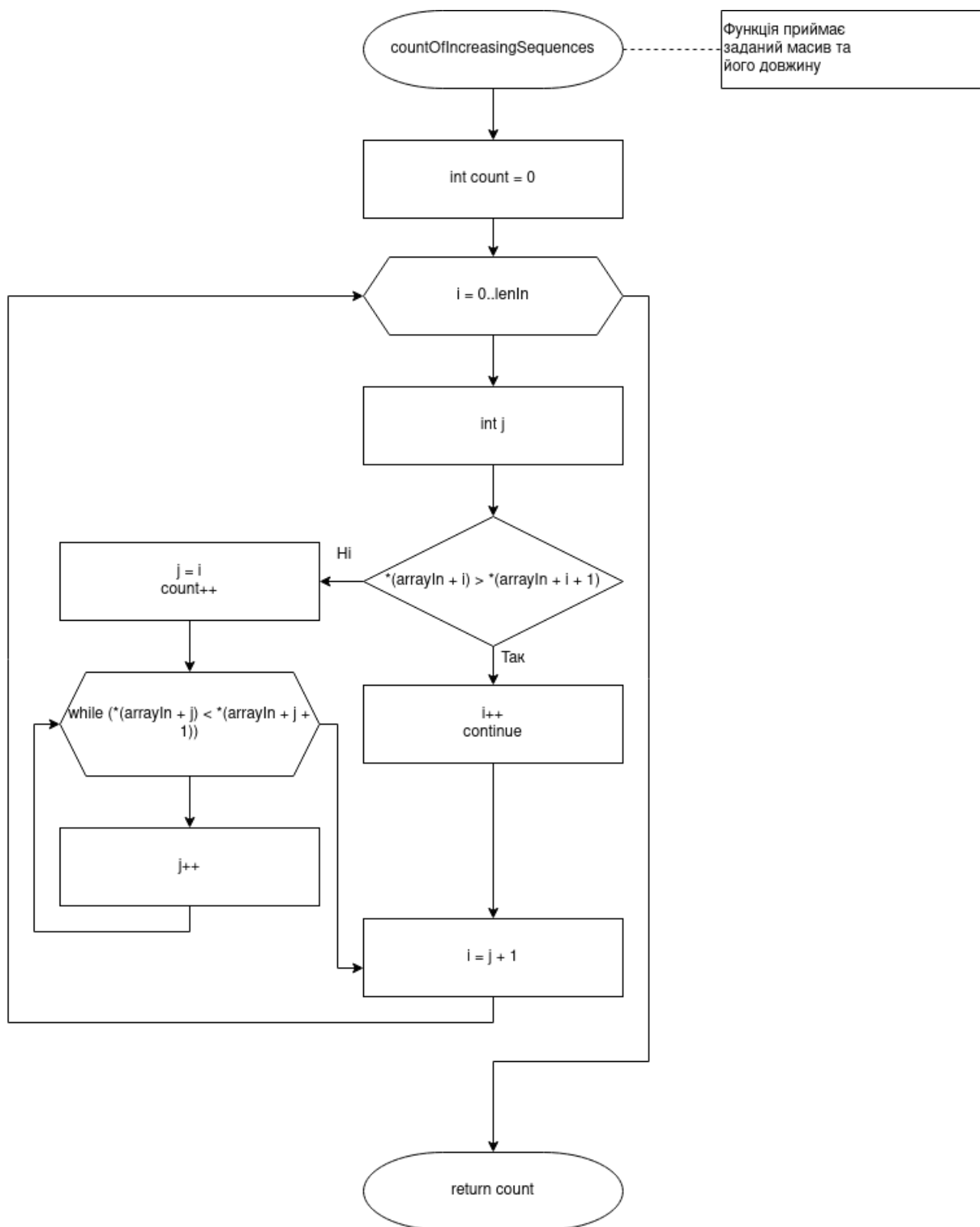


Рис. 3. Схема алгоритму функції countOfIncreasingSequences

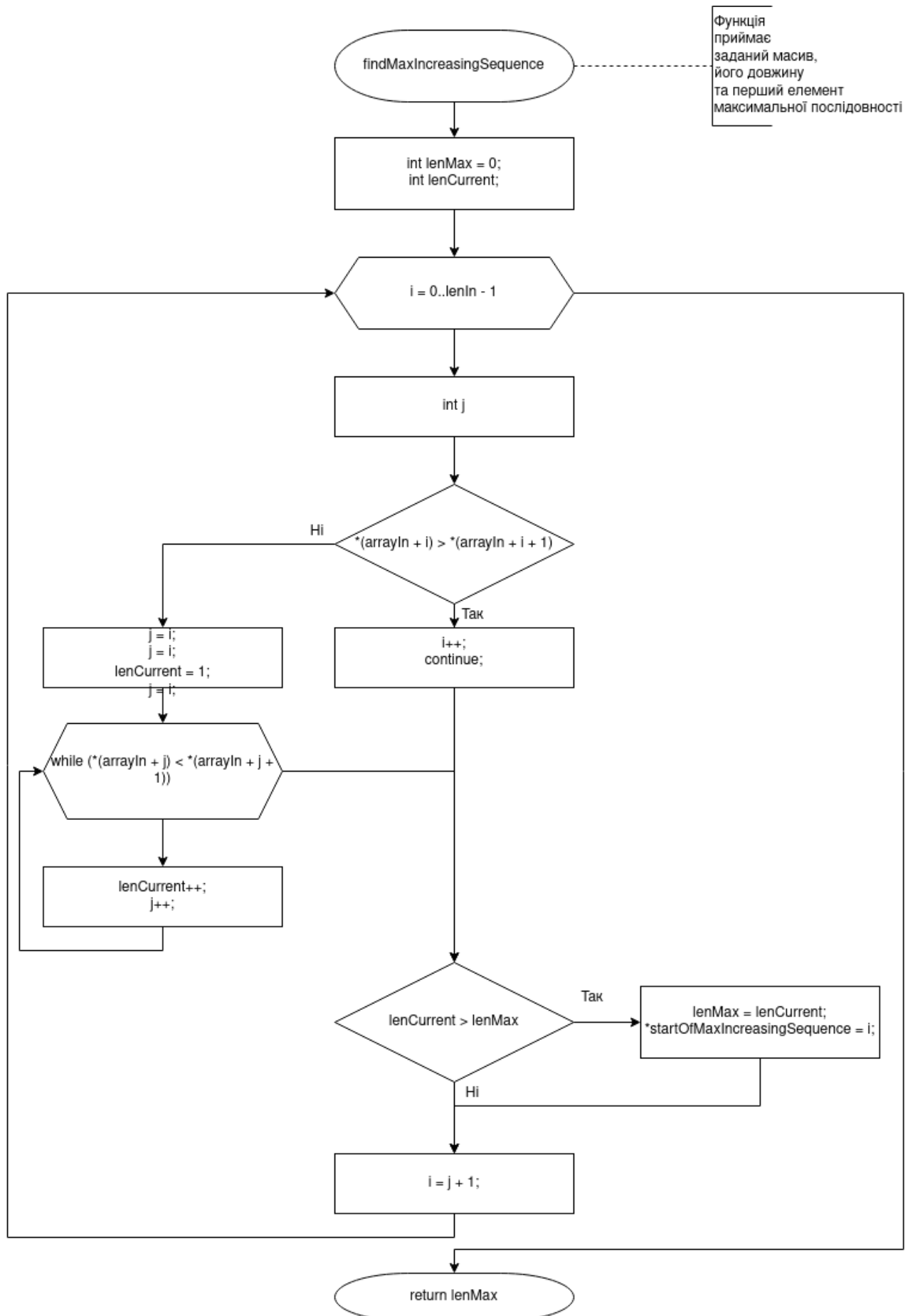


Рис. 4. Схема алгоритму функції `findMaxIncreasingSequence`

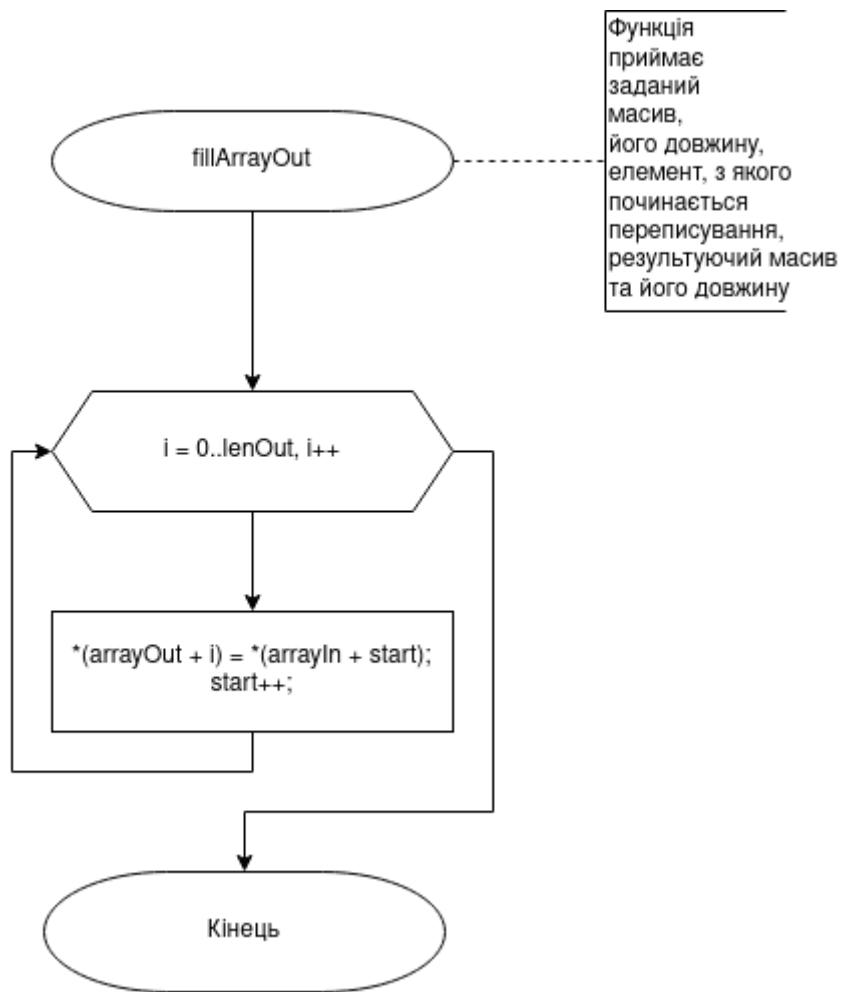
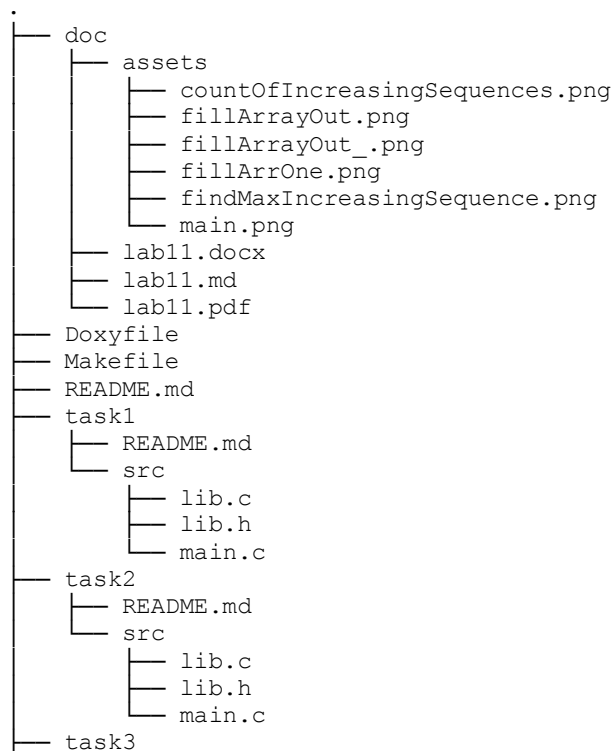
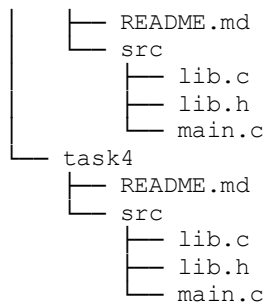


Рис. 5. Схема алгоритму функції fillArrayOut

2.3. Структура проекту





2.4. Генерування Doxygen-документації

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

Лабораторна робота №11 0.1

Вступ до показчиків

Титульна сторінка	Додаткова інформація	Файли ▾	Пошук
-------------------	----------------------	---------	-------

Лабораторна робота №11 Документація

Загальне завдання

Створити програму з використанням показчиків;

Індивідуальне завдання

Підрахувати кількість ділянок заданого масиву з N речовинних чисел, які утворюють безперервні послідовності чисел з незменшуваними значеннями. Максимальну ділянку переписати в інший масив.

Автор

Belchynska K.

Дата

09-dec-2020

Версія

1.0

Створено системою [doxygen](#) 1.8.17

Рис.6 Титульна сторінка Doxygen.

Опис функцій

main()
int main ()

Головна функція.

Послідовність дій:

- виділення пам'яті для заданого масиву речовинних чисел `arrayIn`
- виклик функції для заповнення заданого масиву псевдовипадковими числами `fillArrayIn`
- виклик функції, що оприділяє кількість незменшуваних послідовностей `countOfIncreasingSequences`
- виклик функції, що визначає максимальну довжину ділянки `findMaxIncreasingSequence`
- виділення пам'ті під результуючий масив `arrayOut`
- виклик функції, що заповнює результуючий масив `fillArrayOut`

Повертає
успішний код повернення з програми (0)

Граф всіх викликів цієї функції:

```
graph LR
    main[main] --> countOfIncreasingSequences[countOfIncreasingSequences]
    main --> fillArrayOut[fillArrayOut]
    main --> fillArrOne[fillArrOne]
```

Рис.7 Коментарі до функції main.

Файл lib.c

Функції

Файл з реалізацією функцій для обчислення кількості незменшуваних ділянок, визначення максимальної та переписування її у інший масив Детальніше...

```
#include "lib.h"
```

Діаграма включених заголовочних файлів для lib.c:

```

graph TD
    task3srclibc[task3/src/lib.c] --> libh[lib.h]
    task3srclibc --> stdlibh[stdlib.h]
    task3srclibc --> timeh[time.h]
  
```

Функції

void fillArrOne (float *arrin, int lenIn)

int countOfIncreasingSequences (float *arrayIn, int lenIn)

int findMaxIncreasingSequence (float *arrayIn, int lenIn, int *startOfMaxIncreasingSequence)

void fillArrayOut (float *arrayIn, int start, float *arrayOut, int lenOut)

Детальний опис

Файл з реалізацією функцій для обчислення кількості незменшуваних ділянок, визначення максимальної та переписування її у інший масив

Автор

Belchynska K.

Дата

Рис. 8. Коментарі до функцій файлу lib.c

3. Варіанти використання

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

Variable	Value	Type
Local Variables		
arrayIn	0x555555592a0	float *
*arrayIn	2.71968746	float
countOfSequences	3	int
startOfMaxIncreasingSequence	6	int
lenOut	3	int
arrayOut	0x555555592d0	float *
*arrayOut	2.32697964	float
Function Arguments		

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.