

ЛАБОРАТОРНА РОБОТА № 12. ВЗАЄМОДІЯ З КОРИСТУВАЧЕМ ШЛЯХОМ МЕХАНІЗМУ ВВЕДЕННЯ/ВИВЕДЕННЯ.

ЛАБОРАТОРНА РОБОТА № 14. ВЗАЄМОДІЯ З ФАЙЛАМИ.

1. ВИМОГИ

1.1. Розробник

- Бельчинська Катерина Юріївна;
- студентка групи КІТ-320;
- 2021.

1.2. Індивідуальне завдання (лабораторна робота №12)

Програму, яка була розроблена у лабораторній роботі з показниками змінити так, щоб:

- початкові дані вводилися з клавіатури;
- видача результуючих даних провадилася у консоль;
- при старті програми виводилась інформація об авторі, номері лабораторної роботи;
- при запиті даних, користувач отримав повідомлення, що від нього очікують.

Продемонструвати взаємодію з користувачем шляхом використання функцій:

- *printf()* та *scanf()*;
- *gets()*, *getc()* та *puts()*, *putc()*;
- *write()*, *read()*

1.3. Індивідуальне завдання (лабораторна робота №14)

Програму, яка була розроблена в лабораторній роботі зі строками, змінити так, щоб:

- початкові дані вводилися з файлу;
- видача результуючих даних провадилася не тільки у консоль, але й у файл;
- ім'я вхідного та результуючого файлу повинно бути отримано від користувача;
- при запиті даних, користувач отримав повідомлення, що від нього очікують.

2. ОПИС ПРОГРАМИ (ЛАБОРАТОРНА РОБОТА №12)

2.1. Функціональне призначення

Показчики 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'ять 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. 5.

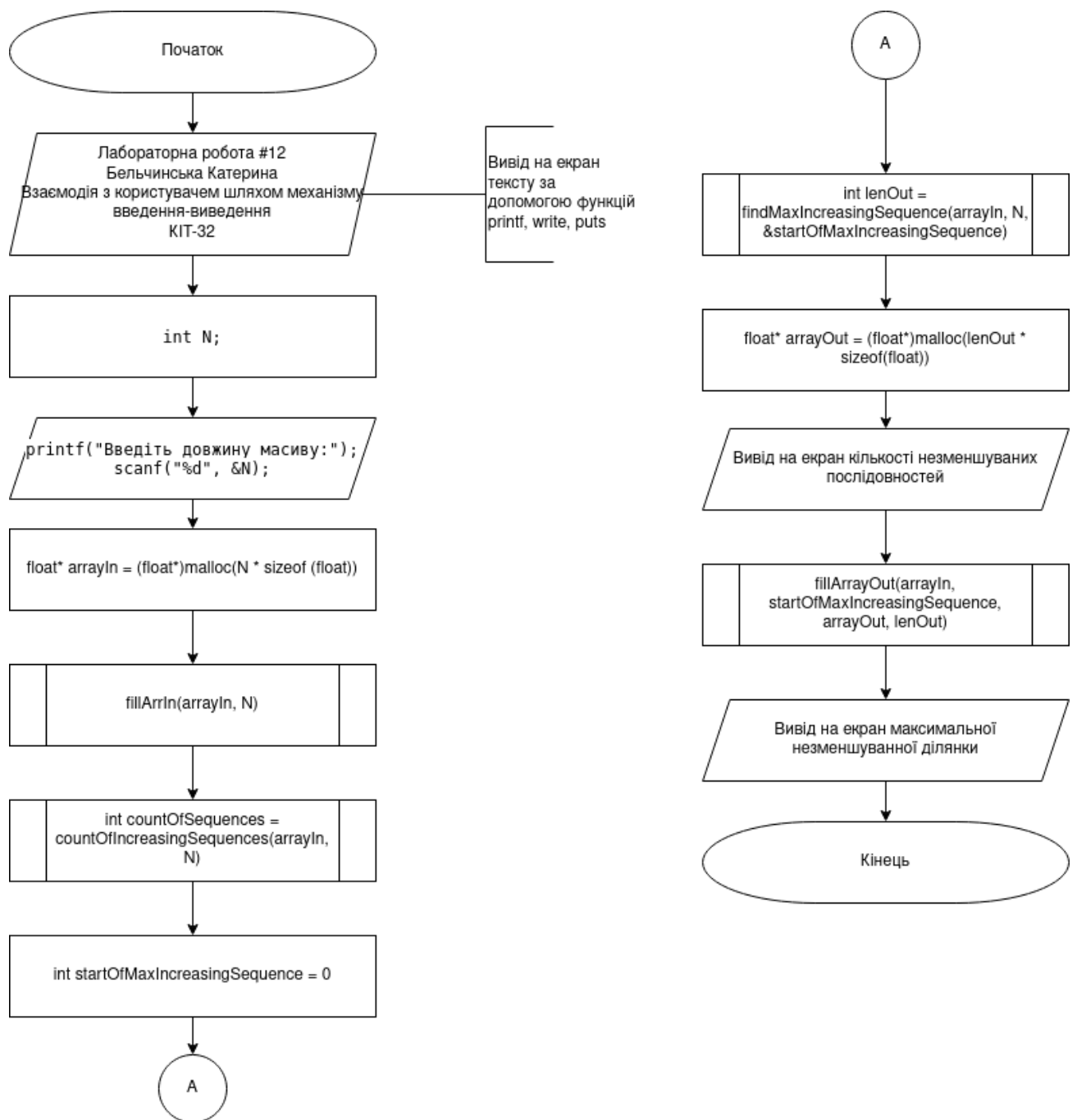


Рис.1. Схема алгоритму функції main

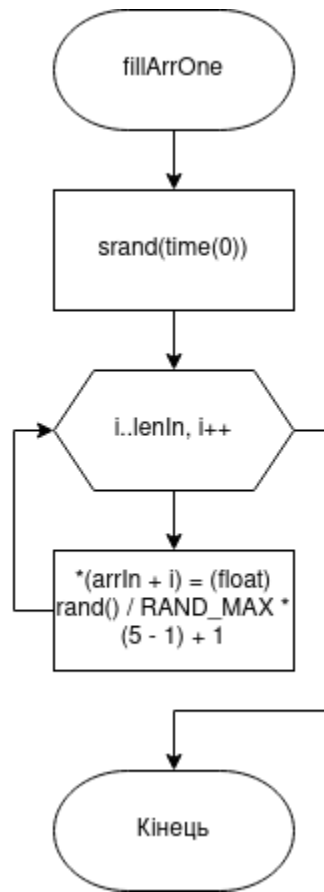


Рис.2. Схема алгоритму функції `fillArrOne`

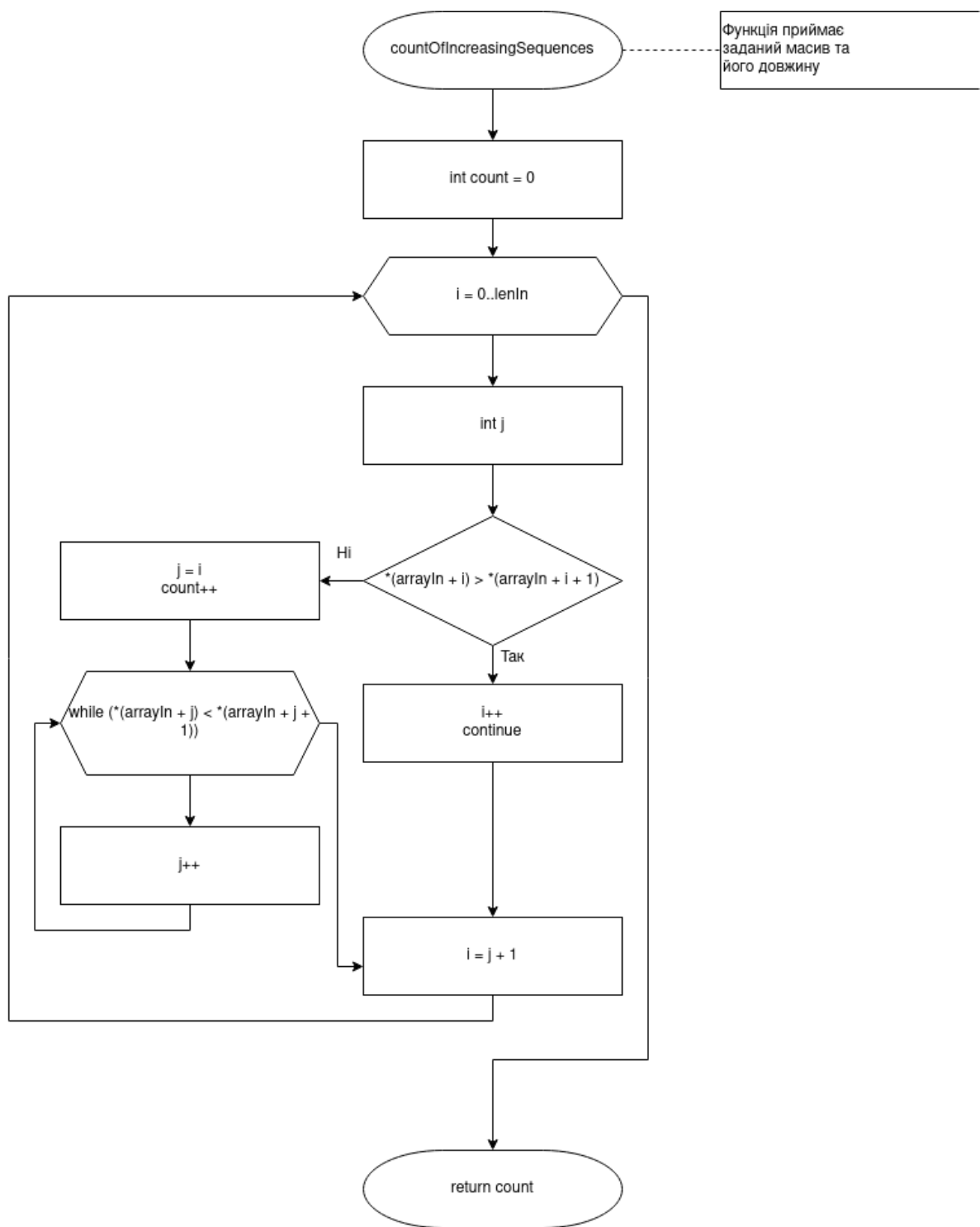


Рис.3. Схема алгоритму функції countOfIncreasingSequences

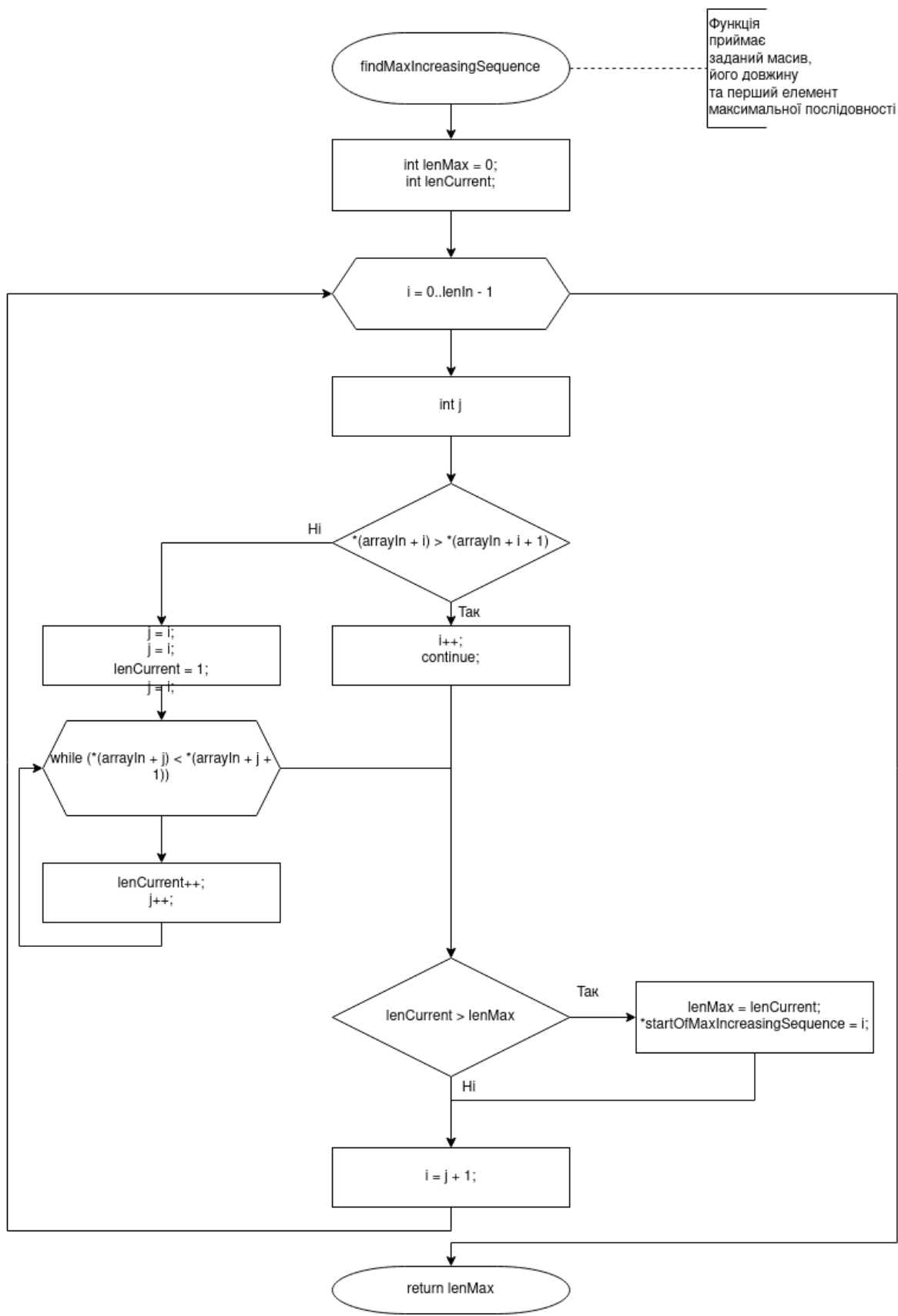


Рис.4. Схема алгоритму функції findMaxIncreasingSequence

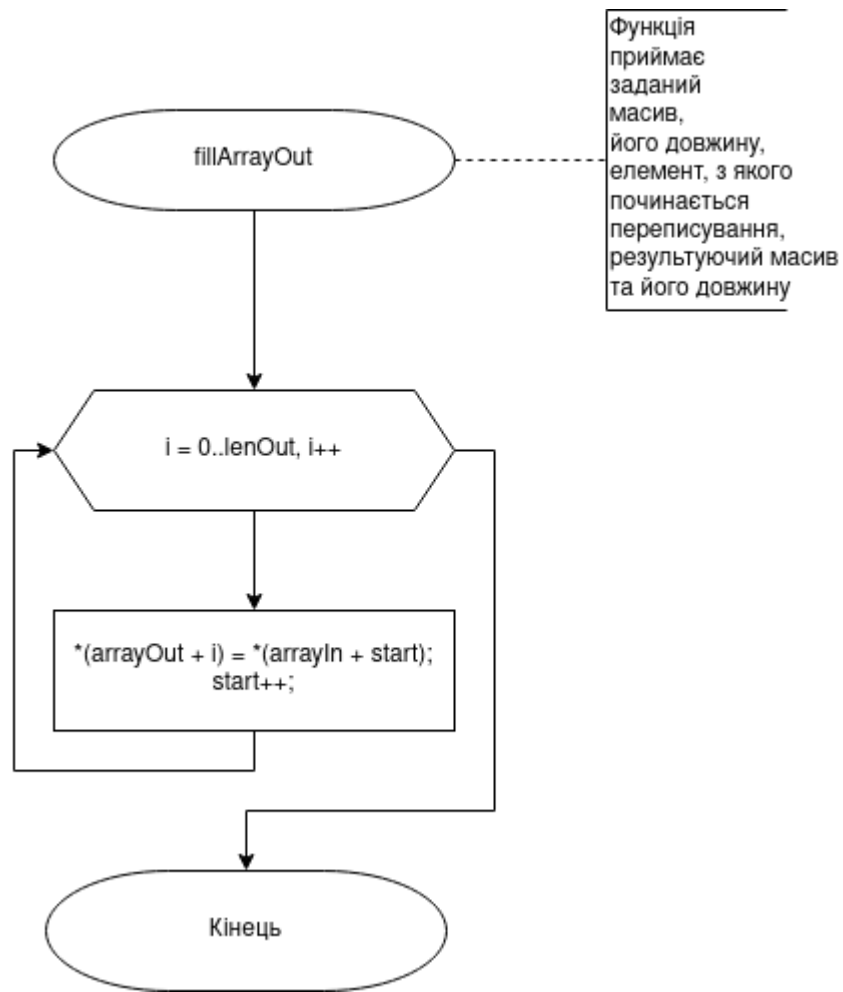
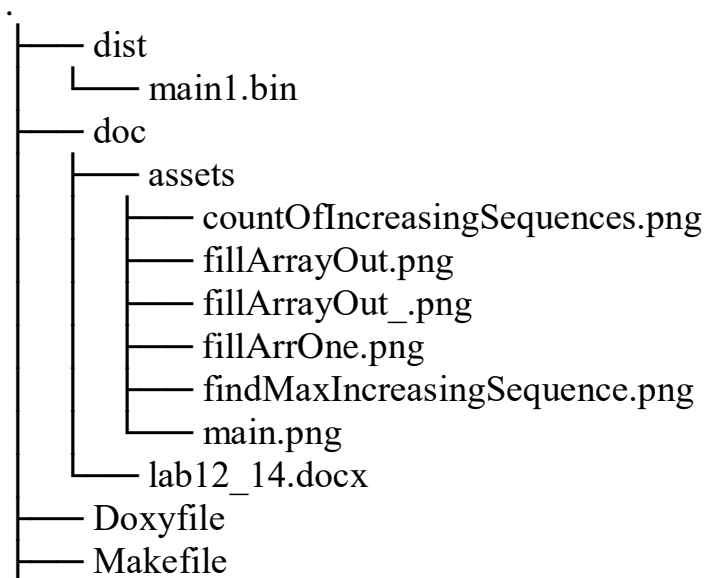
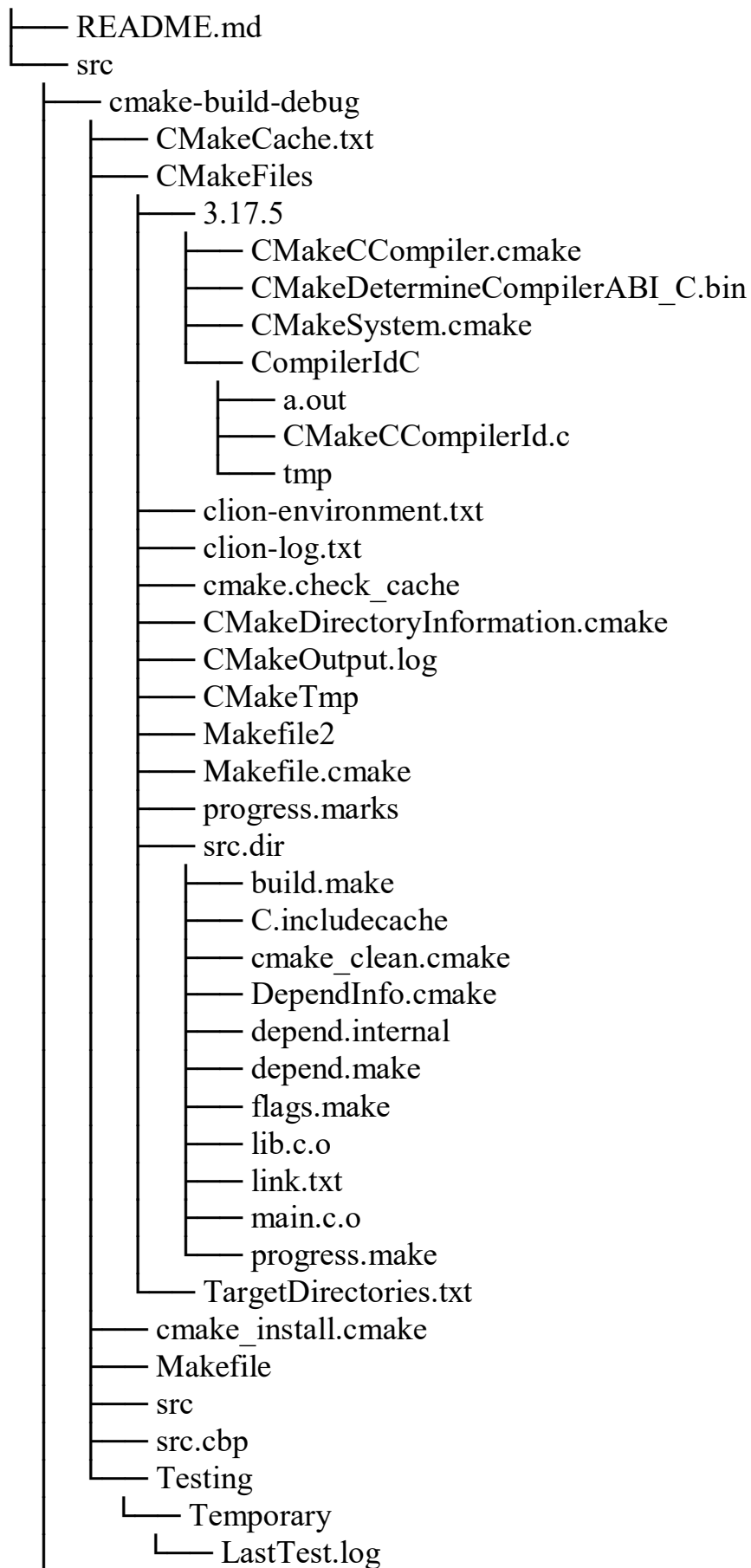
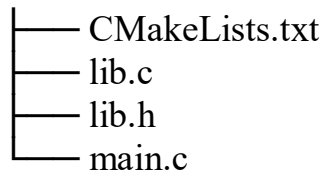


Рис.5. Схема алгоритму функції fillArrayOut

2.3. Структура проекту







2.4. Генерування Doxygen-документації

Лабораторна робота №12 0.1

Введення/виведення даних

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

Лабораторна робота №12 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 незміншуваними значеннями. Максимальну ділянку переписати в інший масив.

Автор
Belchynska K.

Дата
23-feb-2021

Версія
1.0

Створено системою [doxygen](#) 1.8.17

Рис.6 Титульна сторінка Doxygen.

Функції

Файл main.c

Файл з викликом усіх функцій використаних для заповнення результуючого масиву і підрахування кількості ділянок [Детальніше...](#)

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

Діаграма включених заголовочних файлів для main.c:

```

graph TD
    A[src/main.c] --> B[lib.h]
  
```

Функції

```
int main ()
```

Головна функція. [Детальніше...](#)

Детальний опис

Файл з викликом усіх функцій використаних для заповнення результуючого масиву і підрахування кількості ділянок

Автор
Belchynska K.

Дата
23-feb-2021

Версія

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 <unistd.h>
#include "lib.h"
```

Діаграма включених заголовочних файлів для lib.c:

```

graph TD
    src["src/lib.c"] --> unistd["unistd.h"]
    src --> lib["lib.h"]
    src --> stdlib["stdlib.h"]
    src --> time["time.h"]
    src --> stdio["stdio.h"]
    src --> syscall["syscall.h"]
    src --> string["string.h"]
  
```

Функції

void fillArrOne (float *arrIn, int lenIn)

int countOfIncreasingSequences (float *arrIn, int lenIn)

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

void fillArrayOut (float *arrayIn, int start, float *arrayOut1, int lenOut)

void printArrayOut (float *arrayIn, int start, float *arrayOut, int arrayLength)

Рис. 8. Коментарі до функцій файлу lib.c

2.5.Перевірка на утечки пам'яті за допомогою Valgrind:

```

==7495== HEAP SUMMARY:
==7495==    in use at exit: 0 bytes in 0 blocks
==7495== total heap usage: 4 allocs, 4 frees, 2,076 bytes allocated
==7495== All heap blocks were freed -- no leaks are possible
==7495== For lists of detected and suppressed errors, rerun with: -s
==7495== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
  
```

Рис. 9. Перевірка на утечки пам'яті

3. ОПИС ПРОГРАМИ (ЛАБОРАТОРНА РОБОТА №14)

3.1.Функціональне призначення

Програму доцільно використовувати для розрахування частоти появи у даному тексті конкретного символу.

3.2.Опис логічної структури

Функція 'main' виділяє пам'ять для заданого і результуючого масиву, викликає усі функції для обчислення частоти. Схема алгоритму функції наведена на рис. 10.

Функція listdir обчислює та виводить структуру файлів та каталогів та її розмір. Схема алгоритму функції наведена на рис. 11.

Функція readFromFile зчитує початкову інформацію з файлу. Схема алгоритму функції наведена на рис. 12.

Функція writeToFile записує результат обчислення у файл та виводить в консоль. Схема алгоритму функції наведена на рис. 13.

Функція `countTextLength` обчислює довжину заданого масиву. Схема алгоритму функції наведена на рис. 14.

Функція `checker` перевіряє кожен елемент на повтори. Схема алгоритму функції наведена на рис. 15.

Функція `countOfUniqueElements` обчислює кількість унікальних елементів. Схема алгоритму функції наведена на рис. 16.

Функція `getsymbols` переписує унікальні елементи в масив. Схема алгоритму функції наведена на рис. 17.

Функція `getSymbolsCounts` отримує кількість повторів кожного елемента. Схема алгоритму функції наведена на рис. 18.

Функція `fillZeros` ініціалізує результуючий масив. Схема алгоритму функції наведена на рис. 19.

Функція `getSymbolsFrequencies` вираховує та записує в масив частоту появи кожного елемента. Схема алгоритму функції наведена на рис. 20.

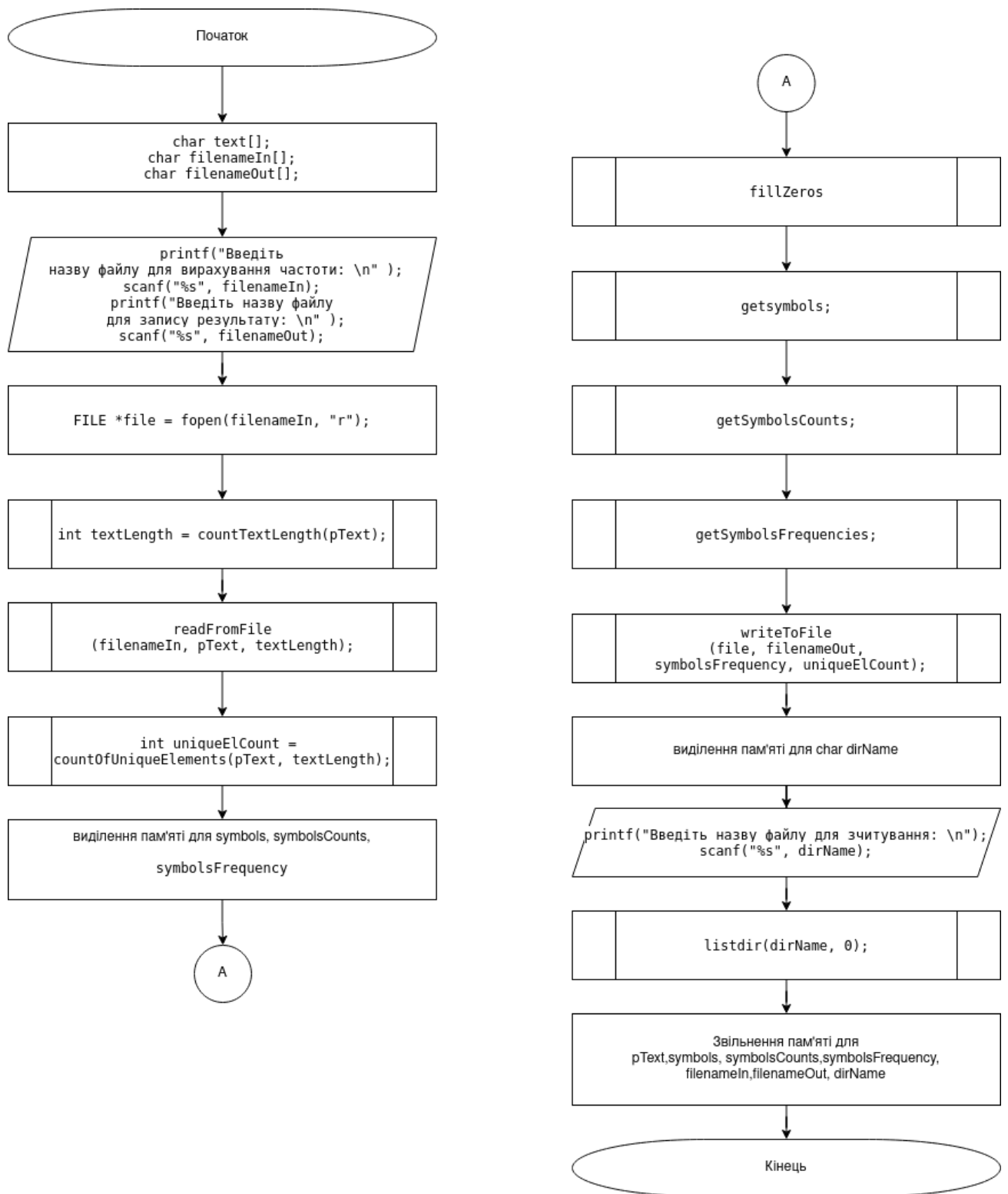


Рис.10. Схема алгоритму функції main

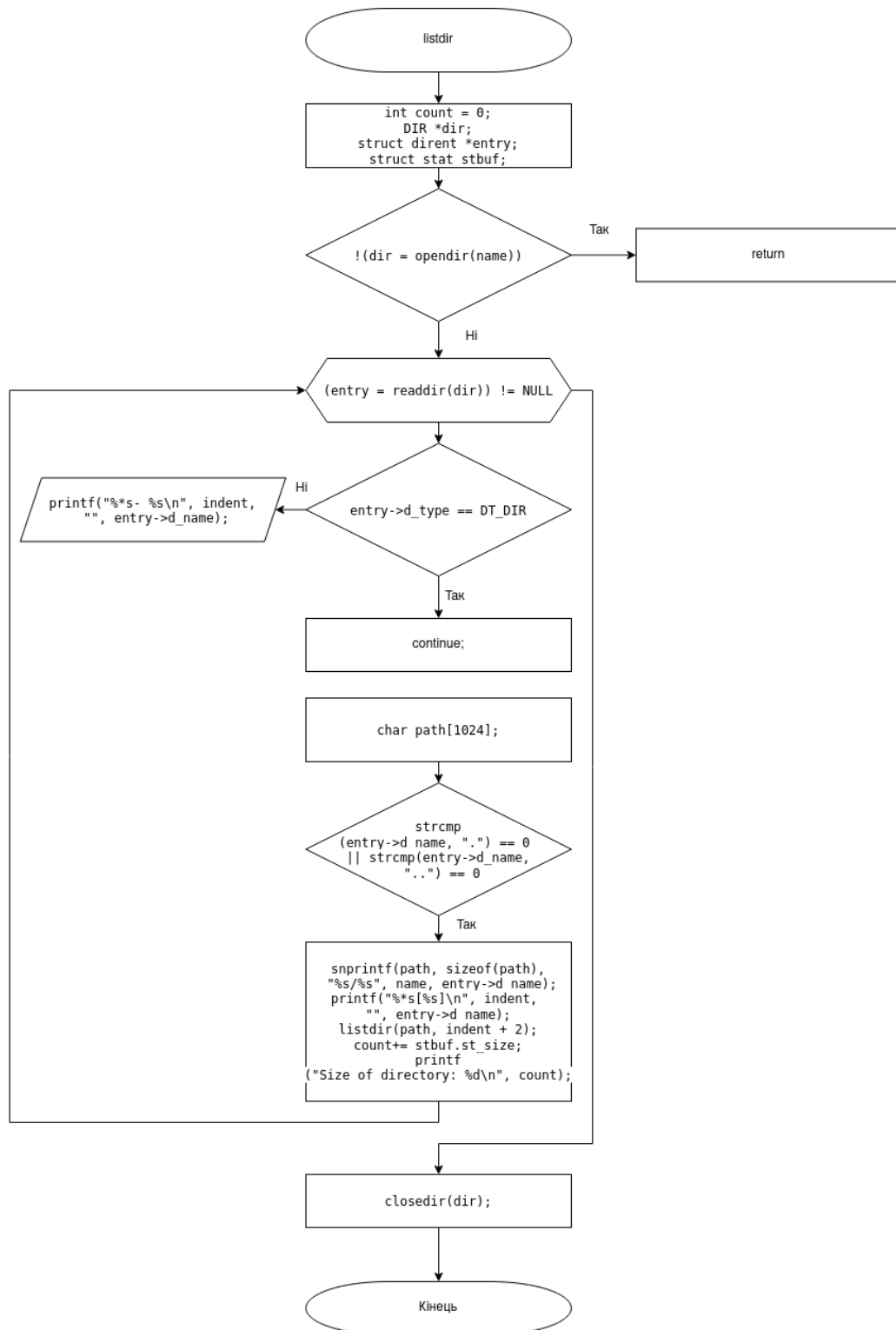


Рис.11. Схема алгоритму функції listdir

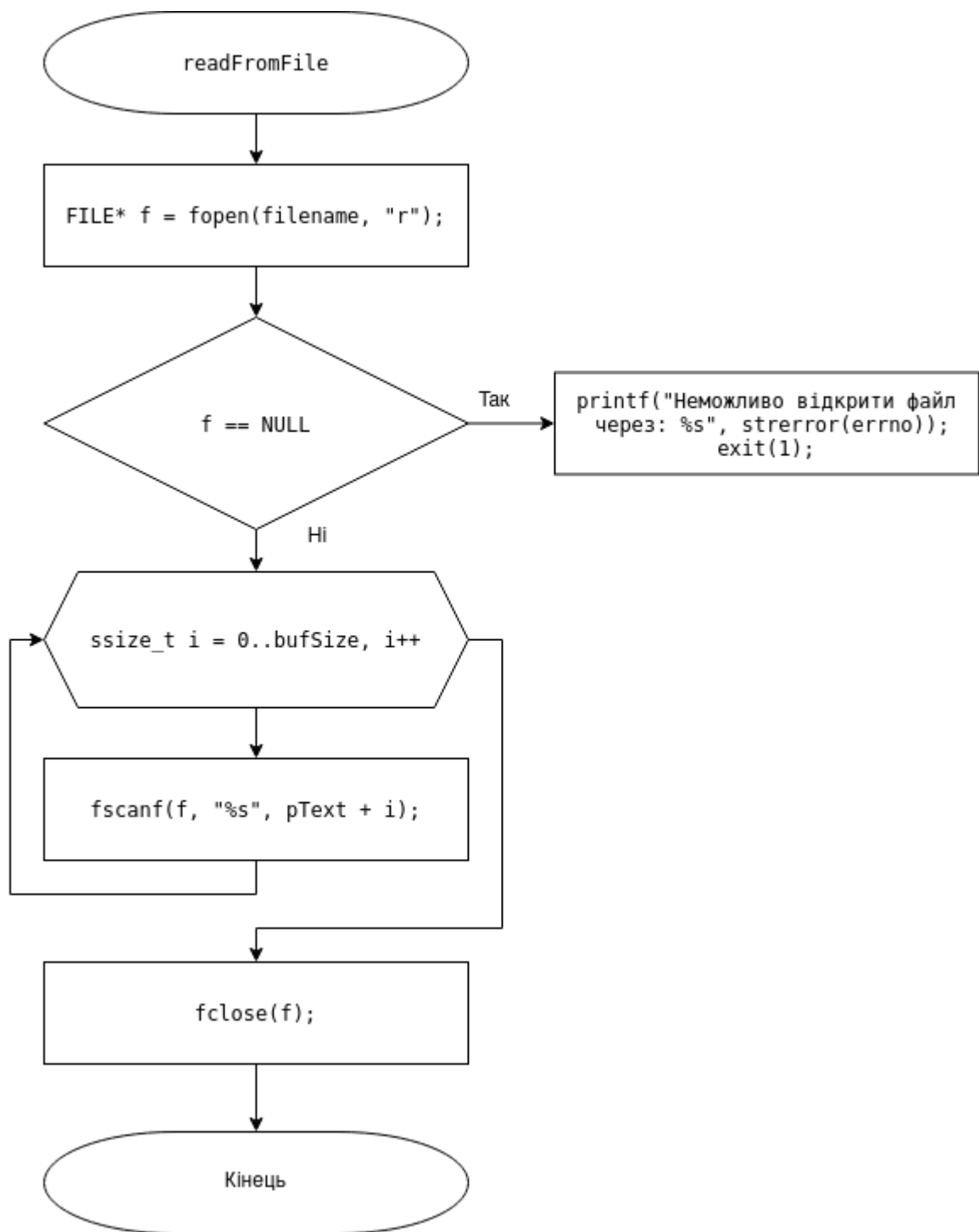


Рис.12. Схема алгоритму функції `readFromFile`

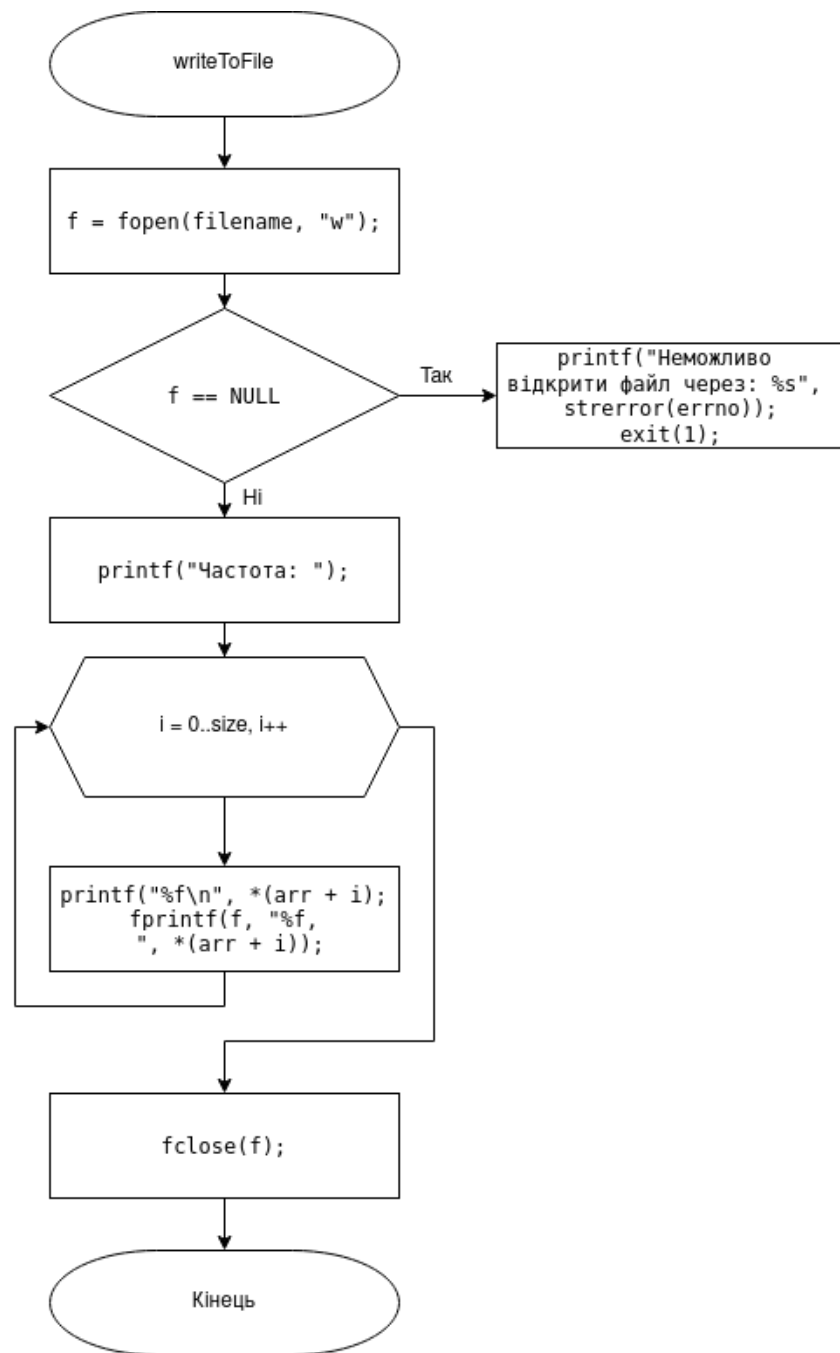


Рис.13. Схема алгоритму функції `writeToFile`

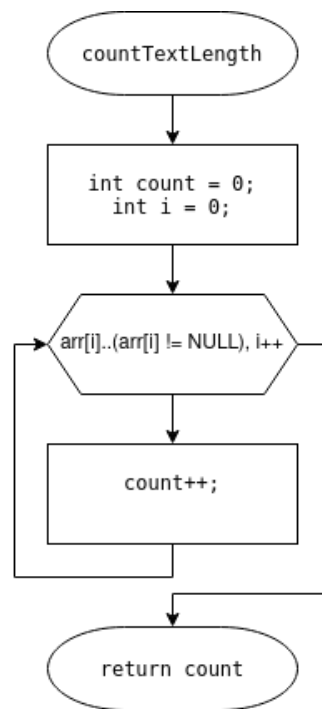


Рис.14. Схема алгоритму функції `CountTextLength`.

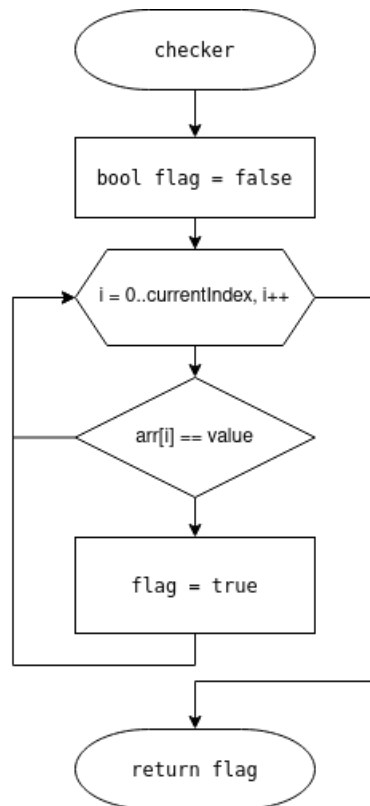


Рис.15. Схема алгоритму функції `checker`

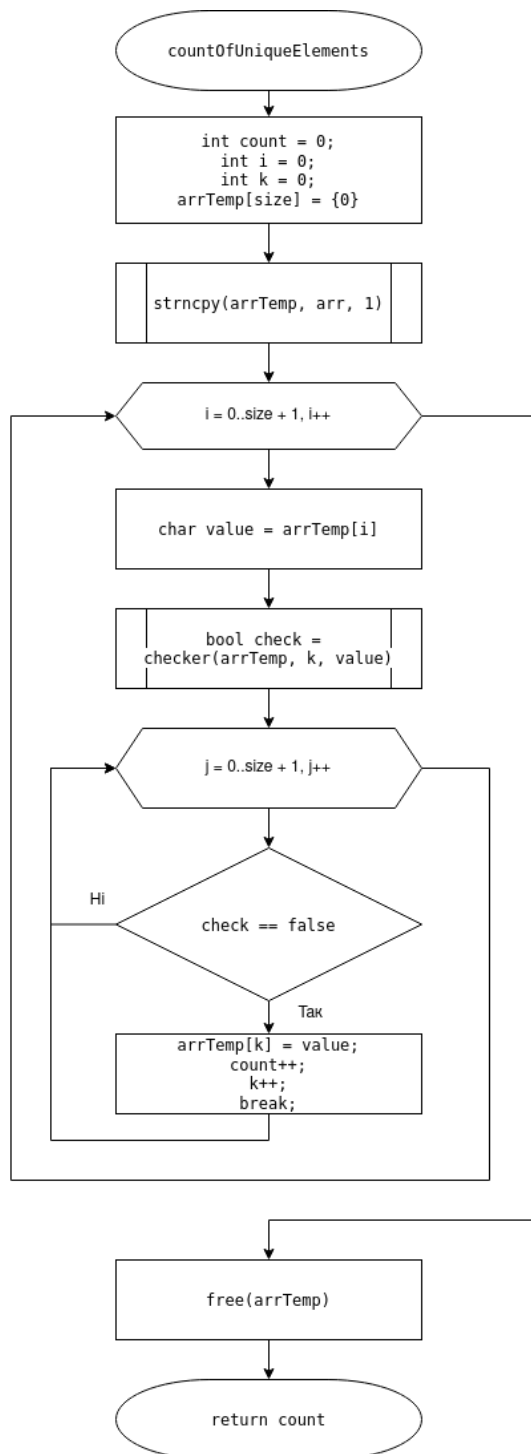


Рис.16. Схема алгоритму функції CountOfUniqueElements.

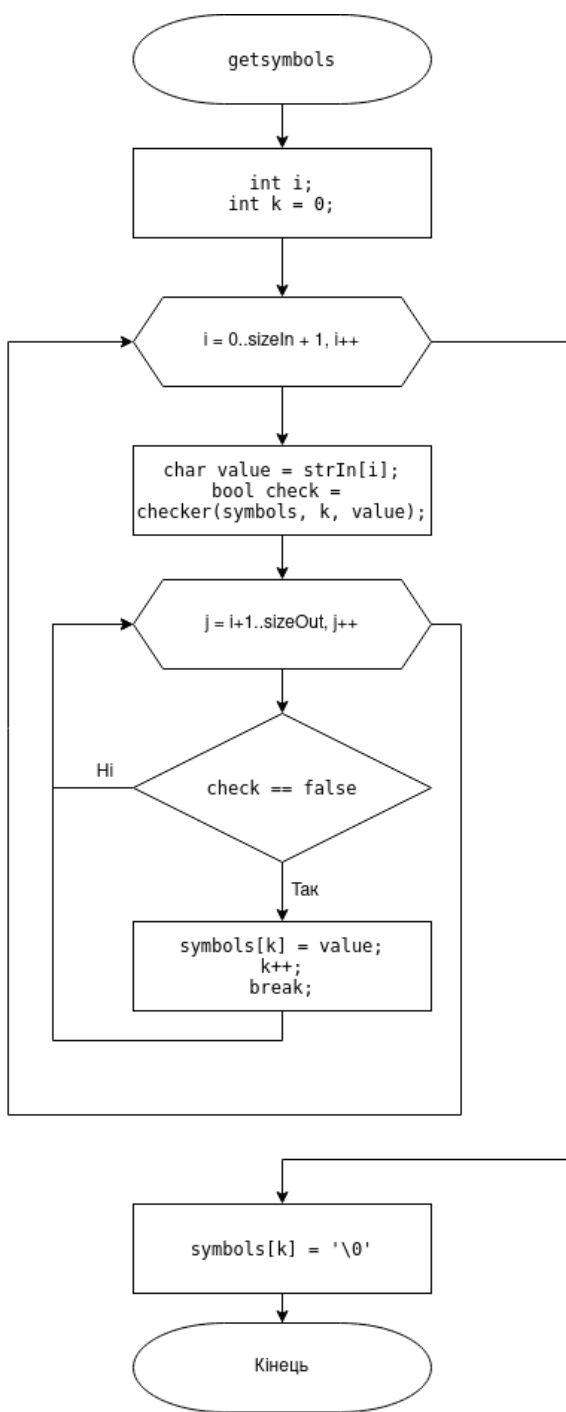


Рис.17. Схема алгоритму функції `getsymbols`.

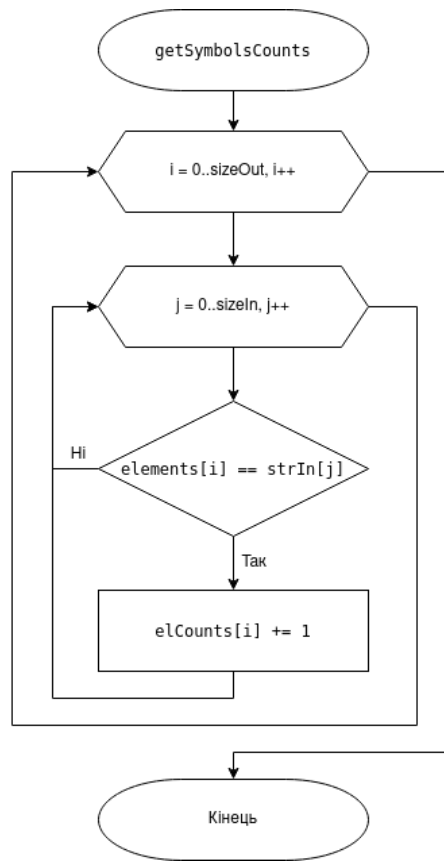


Рис.18. Схема алгоритму функції getSymbolsCounts.

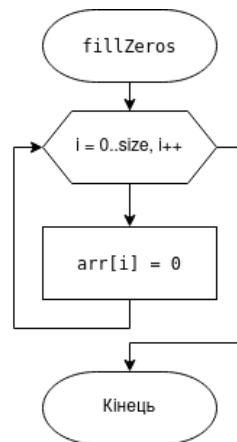


Рис.19. Схема алгоритму функції fillZeros.

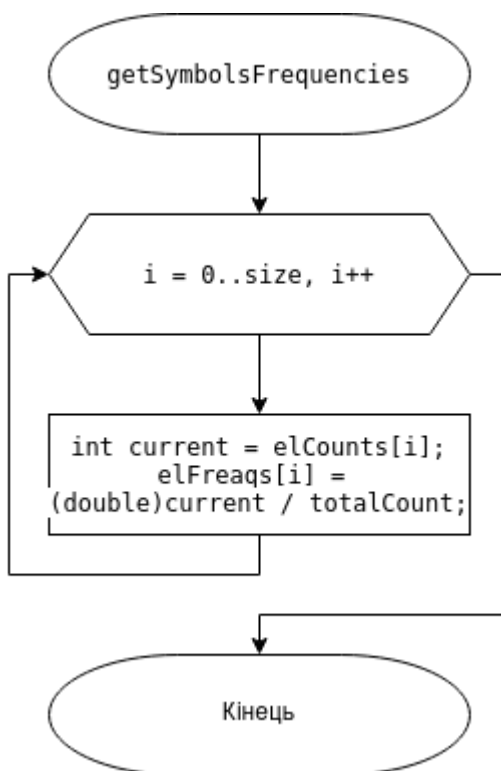
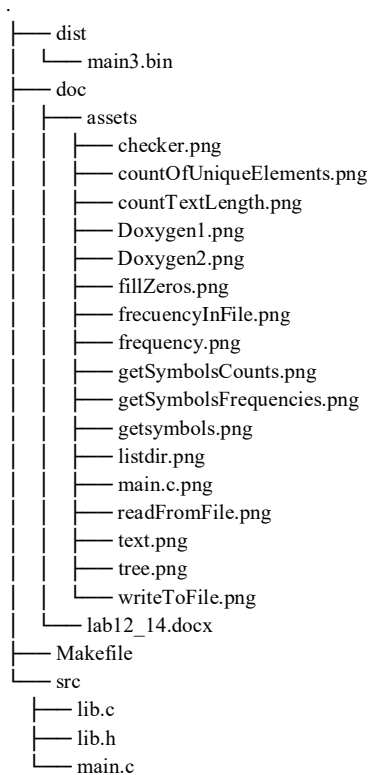


Рис.20. Схема алгоритму функції `getSymbolFrequencies`.

3.3. Структура проекту



3.4. Генерування Doxygen-документації

Лабораторна робота №13 0.1

Строки

Титульна сторінкаДодаткова інформаціяФайли ▾

Пошук

Лабораторна робота №13 Документація

Визначити, скільки у тексті слів (без використання ітерації по кожному символу у циклу). Видати всі слова за алфавітом.

Індивідуальне завдання

Автор
Belchynska K.

Дата
19-dec-2020

Версія
1.0

Генерування документів 1 з 17

Рис. 16. Титульна сторінка Doxygen

doc	
task1	
src	
lib.c	Файл з реалізацією функцій
lib.h	Файл з прототипами функцій
main.c	Головний файл з викликом функцій, виділенням пам'яті для динамічних масивів та заданням вхідного тексту
task2	
src	
lib.c	Файл з реалізацією функцій
lib.h	Файл з прототипами функцій
main.c	Головний файл з викликом функцій, виділенням пам'яті для динамічних масивів та заданням вхідного тексту
task3	
src	
lib.c	
lib.h	Файл з прототипами функцій
main.c	Головний файл з викликом функцій, виділенням пам'яті для динамічних масивів та заданням вхідного тексту
task4	
src	
lib.c	Файл з реалізацією функцій
lib.h	Файл з прототипами функцій
main.c	Головний файл з викликом функцій, виділенням пам'яті для динамічних масивів та заданням вхідного тексту

Рис. 21. Структура файлів в Doxygen.

3.5. Перевірка на утечки пам'яті за допомогою Valgrind:

Рис. 22.
Перевірка
на утечки
пам'яті

```
kate@kate-HP-ProBook-440-G3:~/Programming-Belchynska/lab13$ valgrind --leak-check=yes dist/main3.bin
==10881== Memcheck, a memory error detector
==10881== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==10881== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==10881== Command: dist/main3.bin
==10881==
==10881== Invalid free() / delete / delete[] / realloc()
==10881==    at 0x483CA3F: free (in /usr/lib/x86_64-linux-gnu/valgrind/vgpreload_memcheck-amd64-linux.so)
==10881==    by 0x109DFC: main (main.c:64)
==10881== Address 0x1ffefffec is on thread 1's stack
==10881==    in frame #1, created by main (main.c:43)
==10881==
==10881== HEAP SUMMARY:
==10881==    in use at exit: 0 bytes in 0 blocks
==10881==    total heap usage: 4 allocs, 5 frees, 77 bytes allocated
==10881==
==10881== All heap blocks were freed -- no leaks are possible
==10881==
```

4. ВАРІАНТИ ВИКОРИСТАННЯ (ЛАБОРАТОРНА РОБОТА №12)

Програма виводить на екран послідовність речовинних чисел та довжину
найбільшої послідовності:

```
Результуючий масив:

Результуючий масив:
122.000000
3.211100
3.222200

Process finished with exit code 0
|
```

Рис. 23. Вивід результату

5. ВАРІАНТИ ВИКОРИСТАННЯ (ЛАБОРАТОРНА РОБОТА №14)

Програма зчитує інформацію з файлу та виводить результат у файл
та консоль.

```
1 sfhosspp'"jpo|
```

Рис.24. Вхідний текст у файлі

```

Введіть назву файлу для вирахування частоти:
/home/robert/Programing-Beitchurka/0401/Text.txt
Введіть назву файлу для запису результату:
/home/robert/Programing-Beitchurka/0401/Text.txt
Filesize: 13
Частота: 0.230769
0.076923
0.076923
0.153846
0.153846
0.076923
0.076923
0.076923
0.076923
Введіть назву файлу для зчитування:

```

Рис. 25. Вивід результату обчислення у консолью

```

1 0.230769, 0.076923, 0.076923, 0.153846, 0.153846, 0.076923, 0.076923, 0.076923, 0.076923,

```

Рис.26. Вивід результату у файл

```

Size of directory: 1994085
[src.dir]
- cmake_clean.cmake
- build.make
- lib.c.o
- depend.internal
- progress.make
- depend.make
- flags.make
- C.includecache
- main.c.o
- link.txt
- DependInfo.cmake
Size of directory: 3988170
- progress.marks
- CMakeDirectoryInformation.cmake
- Makefile2
- clion-environment.txt
[3.17.5]
- CMakeSystem.cmake
[CompilerIdC]
[tmp]
Size of directory: 0

```

Рис.27. Вивід структури файлів як утиліта tree та вивід розміру файлів

ВИСНОВКИ

Лабораторна робота №12: в ході даної лабораторної роботи були використані функції для неформатованого низькорівневого та високорівневого вводу/виводу та форматованого вводу/виводу.

Лабораторна робота №14: в ході даної лабораторної роботи було досліджено роботу з файлами, викидання початкових даних з файлу та вивід результату у файл.