Міністерство освіти і науки України

Національний технічний університет

«Харківський політехнічний інститут»

Кафедра «Обчислювальна техніка та програмування»

**ЗВІТ**

Про виконання лабораторної роботи № 7

«Фунцкії»

Керівник: викладач

Бульба С. С.

Виконавець: студент гр. КІТ-120в

Стегній Б. В.

Харків 2020

**Лабораторна робота № 7. Функції**

**1 Вимоги**

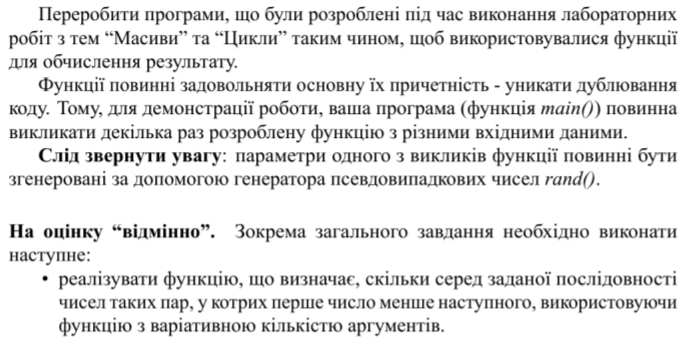
* 1. **Розробник**

• Стегній Богдан Володимирович;

• Студент групи КІТ-120в;

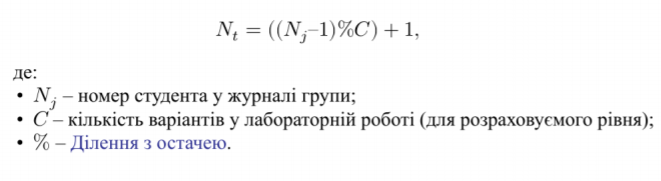
• 22-лист-2020.

* 1. **Загальне завдання**



**1.3 Індивідуальне завдання**

**За даною формулою обчислив варіант для написання звіту**



1. **Виконання роботи**
   1. Створив директорію lab07 у репозиторії Programing-repo.



Рисунок 1 – Створення директорії

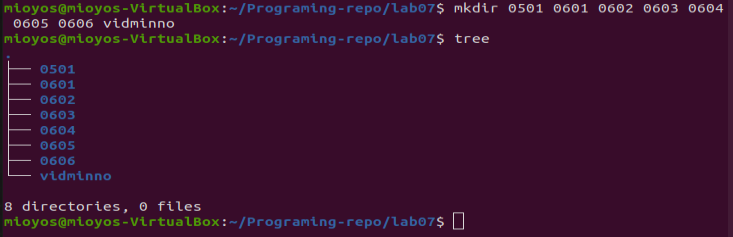


Рисунок 2 – створення директорій

* 1. Заніс потрібні файлі до директорій для того, щоб можна було працювати.

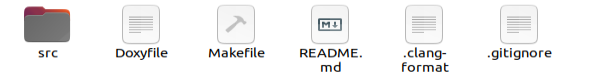


Рисунок 3 – потрібні файли для роботи

* 1. Спочатку перероби код під функції для першої задачі з лабораторної роботи №5 «Масиви».

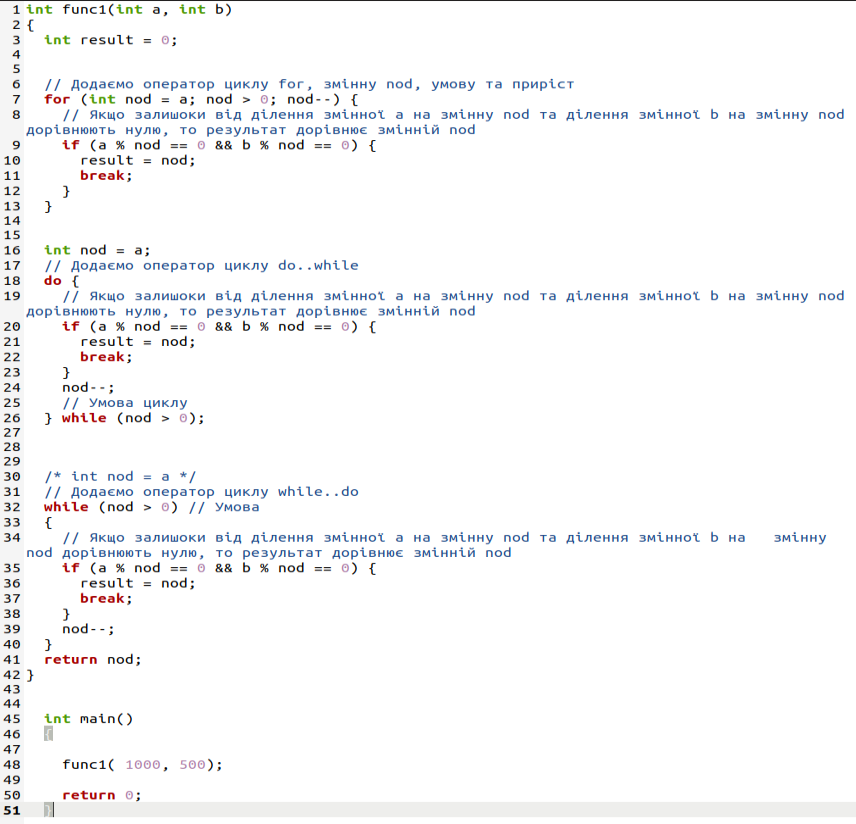
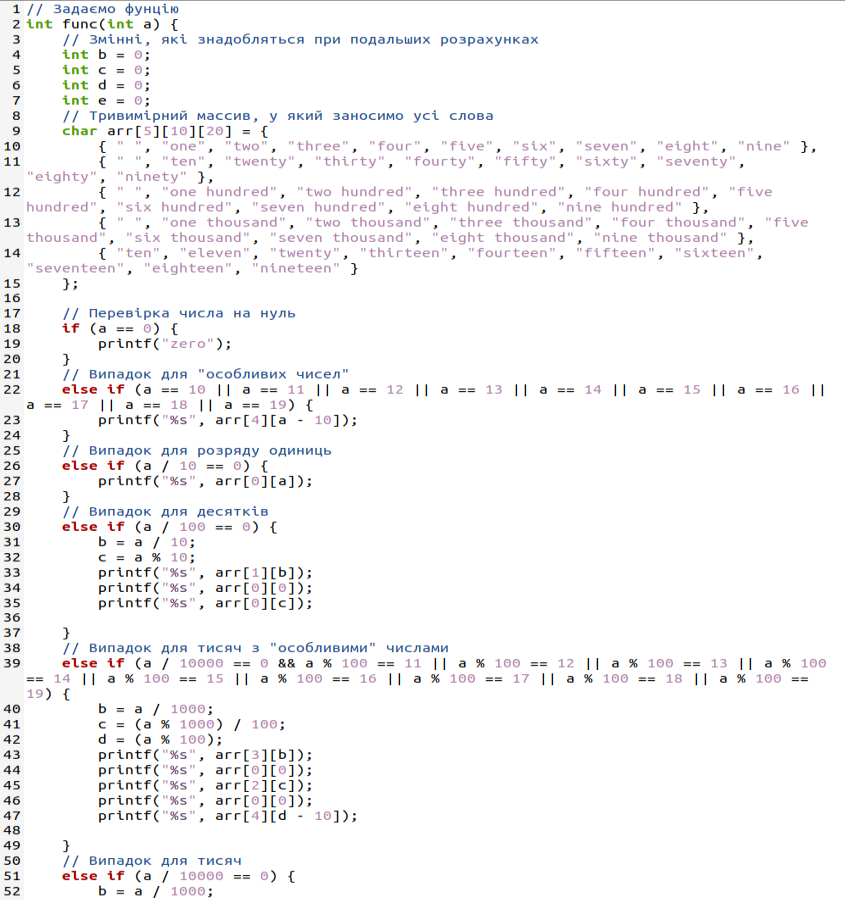


Рисунок 4 – код першого завдання

* 1. Після цього написав код з використанням функції для третьої задачі у шостій лабораторній роботі «Масиви».



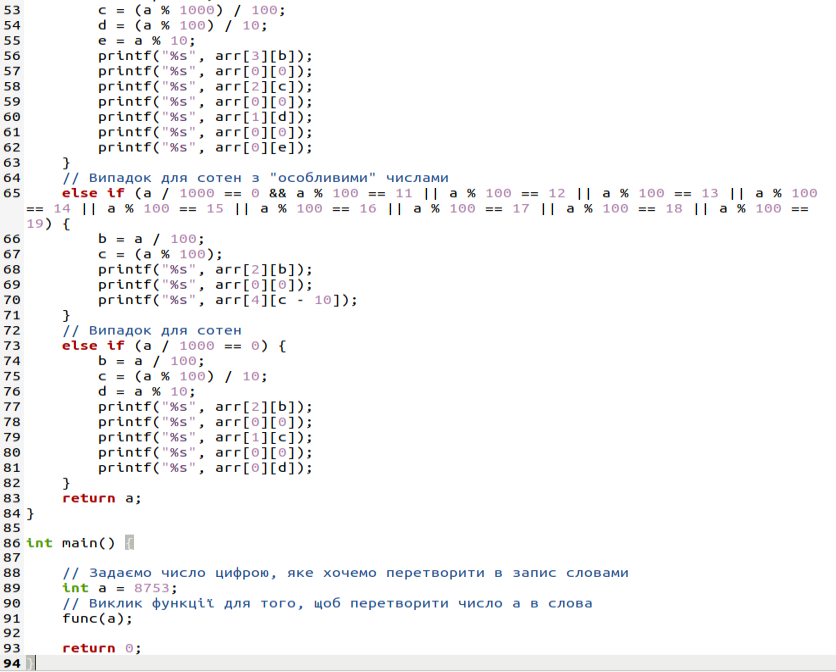


Рисунок 5 – код другої задачі

* 1. Реалізував функцію, яка визначає, скільки серед заданої послідовності чисел таких пар, у котрих перше число менше наступного, використовуючи функцію з варіативною кількістю аргументів.

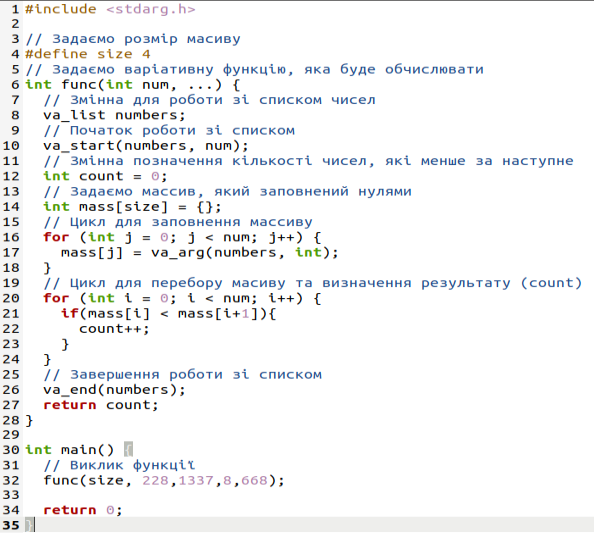


Рисунок 6 – варіативна функція

**2.6** За допомогою команди make clean prep compile скомпілював проекти.



Рисунок 7 – компіляція першої задачі



Рисунок 8 – компіляція другої задачі



Рисунок 9 – компіляція третьої

**2.7** Усі зміни були додані, зафіксовані та завантажені до онлайн-репозиторію Programing-repo за допомогою команд git add, git commit, git push.



Рисунок 10 – git add



Рисунок 11 – git commit



Рисунок 12 – git push

**2.8** Зробив блок-схему для першої програми.

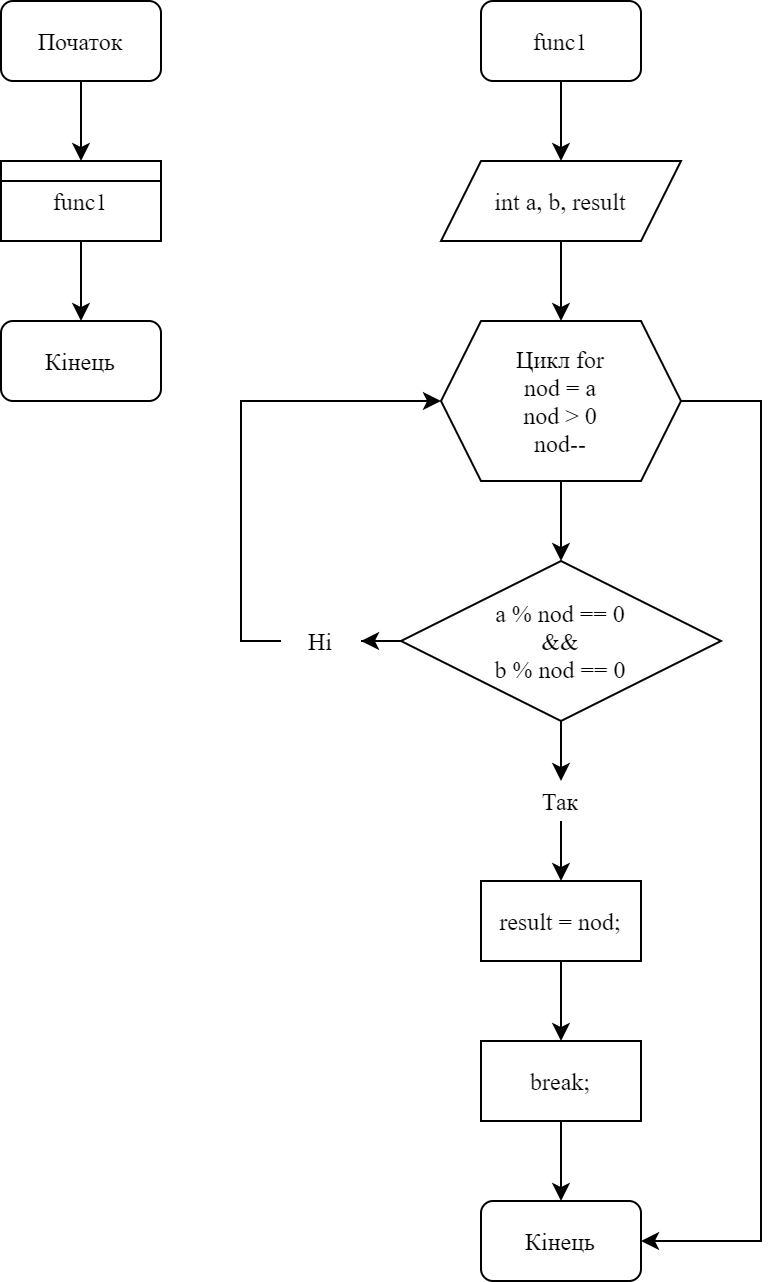


Рисунок 13 – блок-схема першої програми

**2.9** Зробив блок-схему для другої програми.

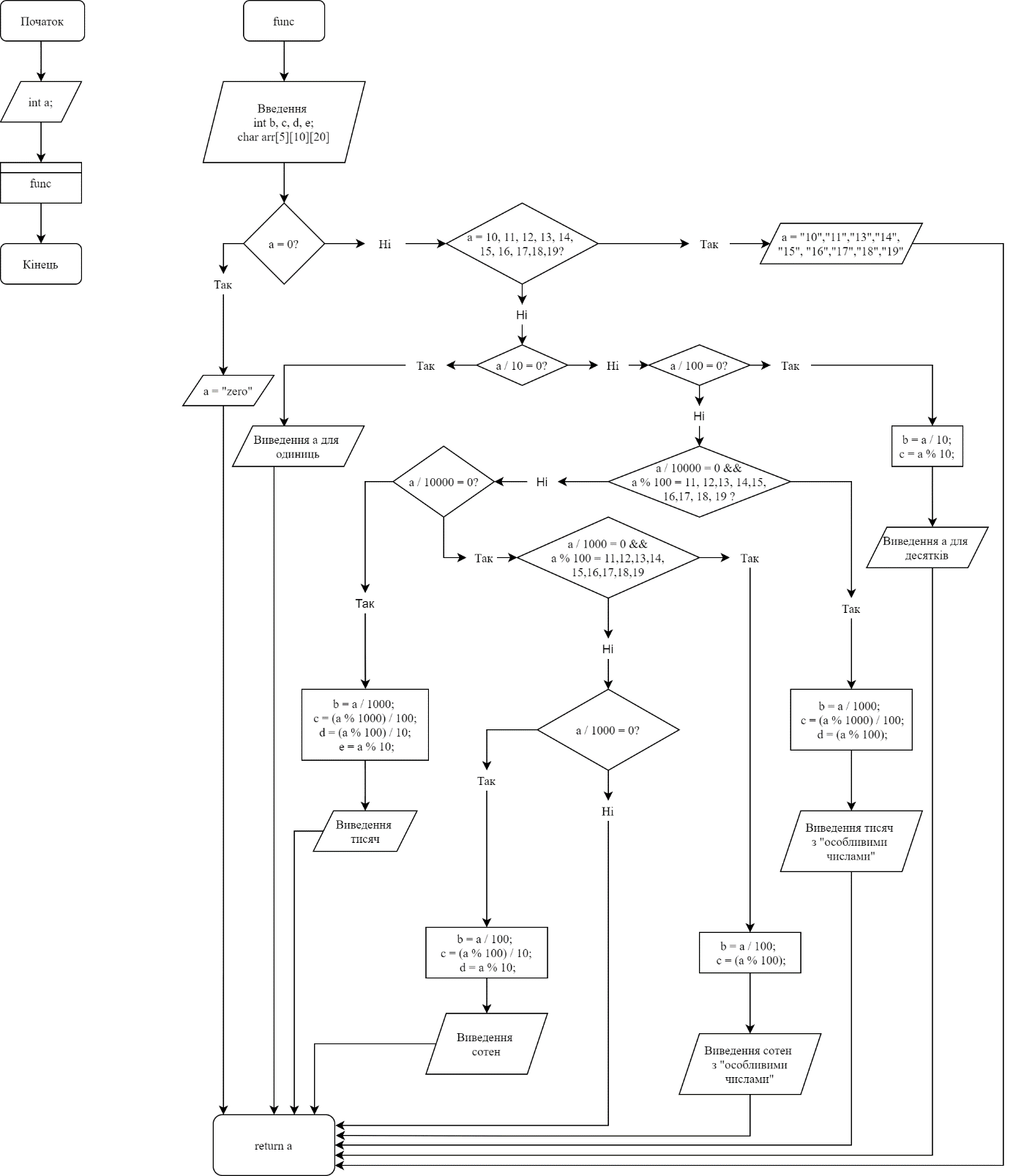
****

Рисунок 14 – блок-схема другої програми

**2.10** Зробив блок-схему для третьої програми.

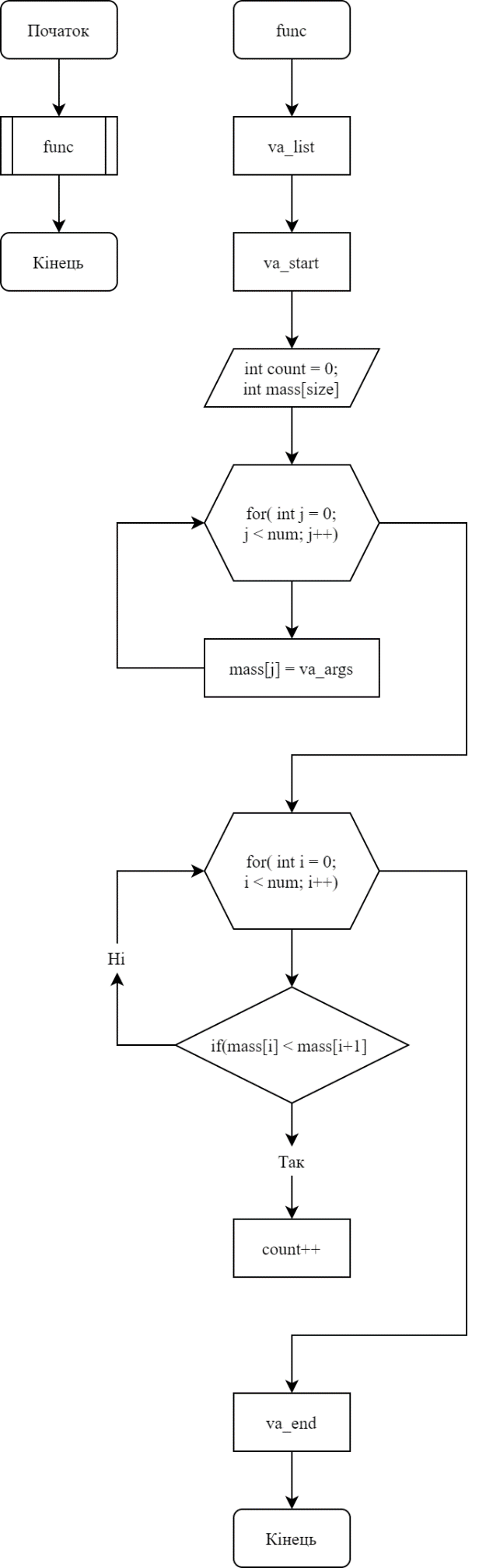


Рисунок 15 – блок-схема третьої програми

**Висновки**

При виконанні лабораторної роботи № 7 «Функції», переробив 7 програм таким чином, щоб у їх виконанні були задіяні функції. Були розвинуті нові навички у роботі з середовищем Linux.

**Контрольні питання**

1. Коли аргументи передаються в функцію **за значенням**, то відбувається створення копій аргументів, передача їх у функцію і присвоєння параметрам. Викликана функція працює з копією аргументів, тому жодні зміни значень параметрів не відіб’ються на зміні аргументів.

2. У тілі функції може бути декілька операторів return, але може не бути й жодного. В останньому випадку повернення у викликаючу програму відбувається після виконання останнього оператора тіла функції.

3. Прототип функції — декларація [функції](https://uk.wikipedia.org/wiki/%D0%9F%D1%96%D0%B4%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D0%B0" \o "Підпрограма) в деяких [мовах програмування](https://uk.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B2%D0%B0_%D0%BF%D1%80%D0%BE%D0%B3%D1%80%D0%B0%D0%BC%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F" \o "Мова програмування), при якій вказується тільки її назва, [тип](https://uk.wikipedia.org/wiki/%D0%A2%D0%B8%D0%BF_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85) та кількість і типи її [аргументів](https://uk.wikipedia.org/wiki/%D0%90%D1%80%D0%B3%D1%83%D0%BC%D0%B5%D0%BD%D1%82" \o "Аргумент). Таке оголошення необхідне і достатнє для [компілятора](https://uk.wikipedia.org/wiki/%D0%9A%D0%BE%D0%BC%D0%BF%D1%96%D0%BB%D1%8F%D1%82%D0%BE%D1%80" \o "Компілятор) при створенні правильного об'єктного коду.

4. void як ім'я типу значення, що повертається функцією: вказує на те, що функція не повертає значення, а виклик такої функції є void-виразом. Тіло такої функції не повинно містити операторів return з виразами.

5. Наприклад:

int funcname()

{

}

6. Напириклад

void funcname()

{

}

7. Параметр за замовчуванням - це параметр функції, який має певне значення. Якщо користувач не передає в функцію значення для параметра, то використовується значення за замовчуванням. Якщо ж користувач передає значення, то це значення використовується замість значення за замовчуванням.

8. Параметрів у функції може бути скільки завгодно чи не бути взагалі. Круглі дужки в будь-якому випадку вказуються. Якщо параметрів кілька, то їх між собою необхідно розділити за допомогою коми. Вони дозволяють більш зручно отримати передані аргументи функції при її виклику.

9. Випадкові числа в мові програмування С можуть бути згенеровані функцією rand () зі стандартної бібліотеки С. Функція rand() генерує числа в діапазоні від 0 до RAND\_MAX. RAND\_MAX - це константа, визначена в бібліотеці <cstdlib>.

10. Потрібно використовувати фунцію time()

11. Для використання бібліотечних функцій потрібно зазначити їх в самому початку роботи (наприклад, #include <funcname>). У бібліотчечних функціях вже прописан алгоритм дій, який не потрібно прописувати самому. Також бібліотечні функції не «захламляють» код. Функціїї, написані користувачем не портібно ніяк підключати. Але на відміну від бібліотечних функцій, користувач подвивен приписувати алгоритм дій сам.

12. Варіативні функції - це функції з перемінним числом аргументов.

13. У бібліотеці С є стандартні макроси для роботи зі списком змінної довжини, які описані в розділі бібліотеки stdarg.h. Таких макросів є чотири: va\_lіst, va\_start, va\_arg, va\_end.