

Лабораторна робота №11 Вступ до показчиків

1 Вимоги

1.1 Розробник

- Носов Микола
- студент групи кіт 1206
- 23.12.2020

1.2 Мета:

Навчитися працювати з показчиками, виділенням пам'яті

1.4 Індивідуальне завдання:

1. Дано масив з N цілих чисел. Визначити, чи є в масиві елементи, що повторюються; якщо такі є, то створити масив, в якому вказати, скільки разів які елементи повторюються. Таким чином, в результуючому масиві кожен непарний елемент - число, що повторюються; кожен парний елемент - кількість повторювань.
2. Дано масив з N цілих чисел. Знайти безперервну послідовність позитивних чисел у вхідному масиві, сума елементів якої максимальна, та переписати їх у вихідний масив.
3. Дано масив з N речовинних чисел. Підрахувати кількість ділянок, які утворюють безперервні послідовності чисел з не-зменшуваними значеннями. Максимальну ділянку переписати у інший масив.
4. Дано масив масивів з $N * N$ цілих чисел. Елементи головної діагоналі записати в одновимірний масив, отриманий масив упорядкувати за зростанням.

2 Опис програми

2.1 Функціональне призначення

Програма призначена для:

- знаходження числа повторювань елементів масиву;
- знаходження безперервної додатньої послідовності, сума елементів якої найбільша ;
- знаходження максимальної ділянки чисел, с не-зменшуваними значеннями;
- упорядкування за зростанням елементів головної діагоналі матриці;

2.2 Логічна структура проекту

Структура проекту (див рис. 1)

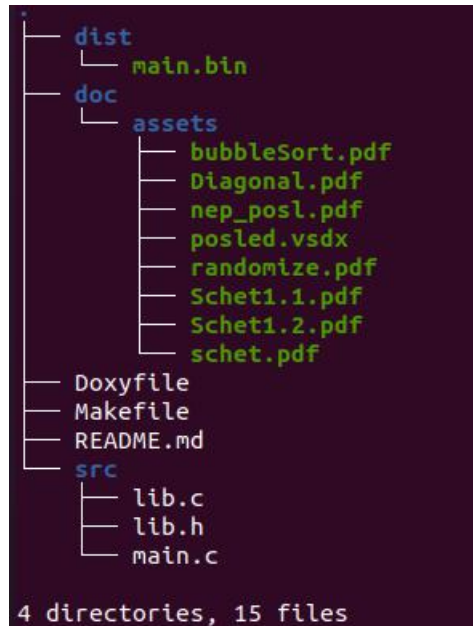


Рисунок 1 - робота утиліти tree

1. **Задача 1** - знаходження числа повторювань елементів масиву

Призначення :

- знайти число повторювань елементів масиву

Аргументи :

- масив чисел

Опис роботи:

- масив сортується за збільшенням

- доки перший елемент дорівнює наступному, числа повторюються

Схеми алгоритмів(див рис 2, 3, 4):

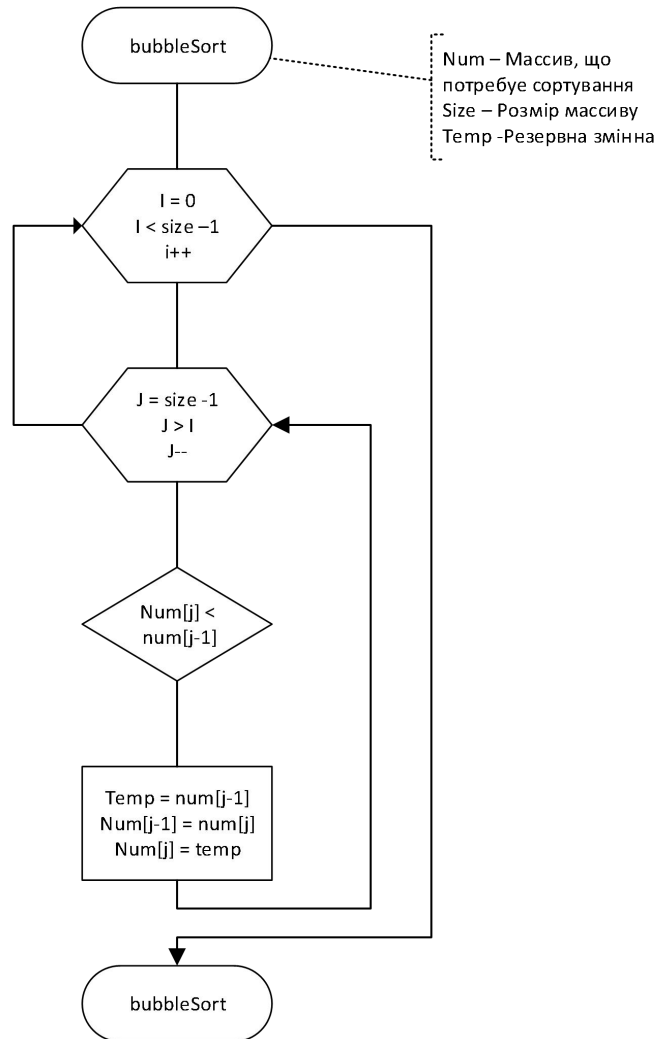


Рисунок 2 - бульбашкове сортування

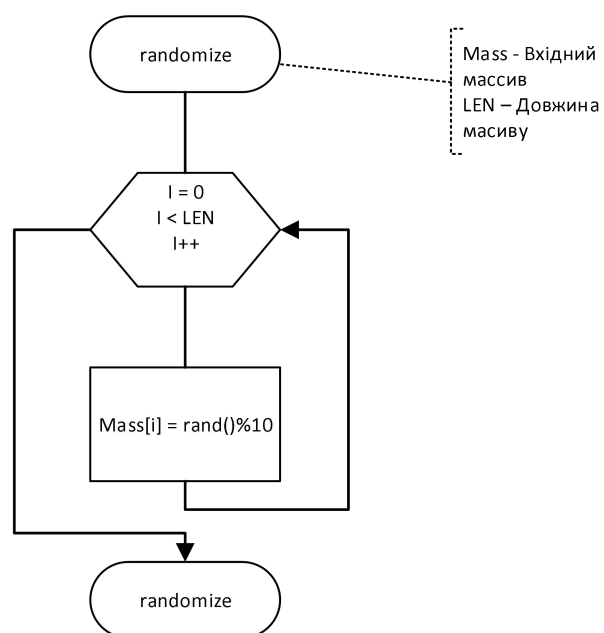


Рисунок 3 - заповнення масива випадковими числами

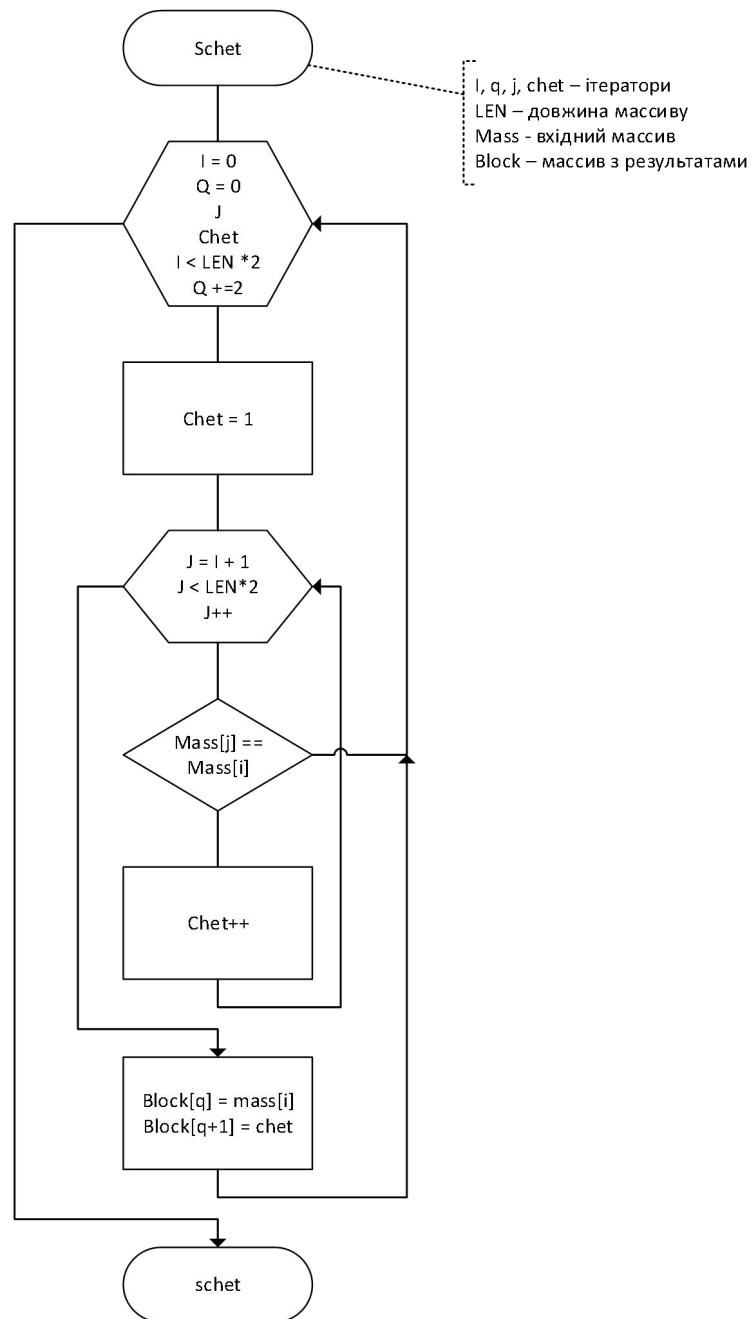


Рисунок 4 - основний алгоритм задачі

Задача 2

Призначення :

- знаходження безперервної додатньої послідовності, сума елементів якої найбільша

Аргументи :

- числова послідовність

Опис роботи:

- визначення послідовностей
- заповнення 2 - вимірного масиву елементами всіх послідовностей
- підрахунок суми всіх послідовностей та порівняння

Блок схеми задачі :

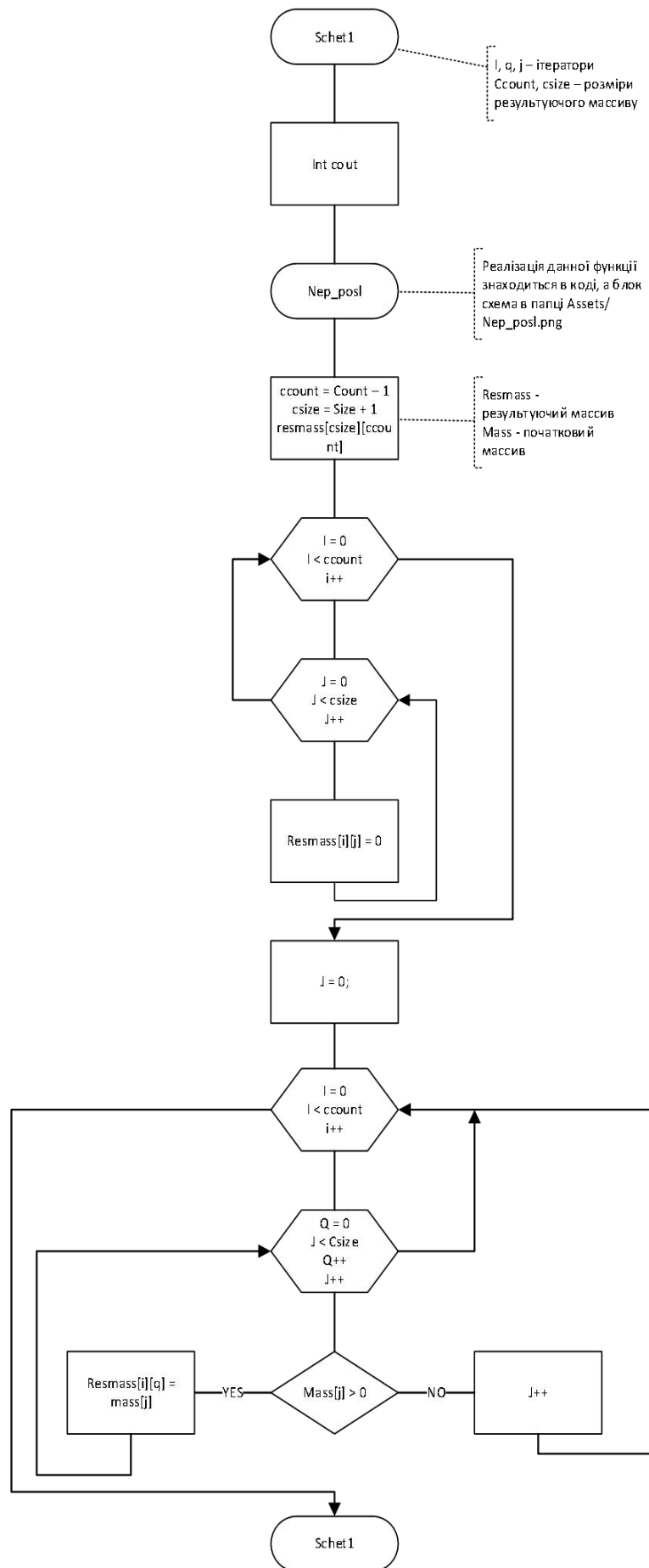


Рисунок 5 - основний алгоритм задачі

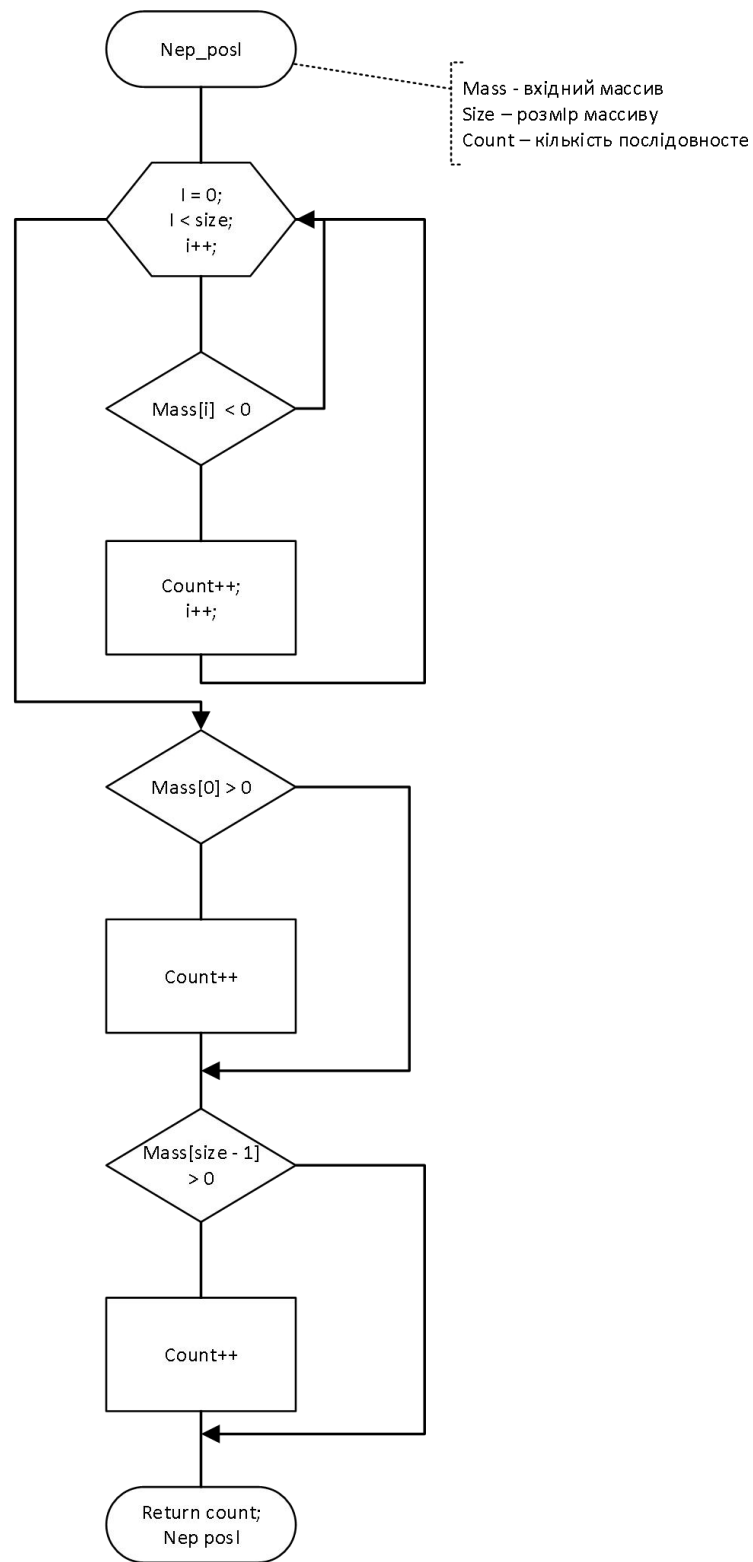


Рисунок 6 - підрахунок кількості послідовностей

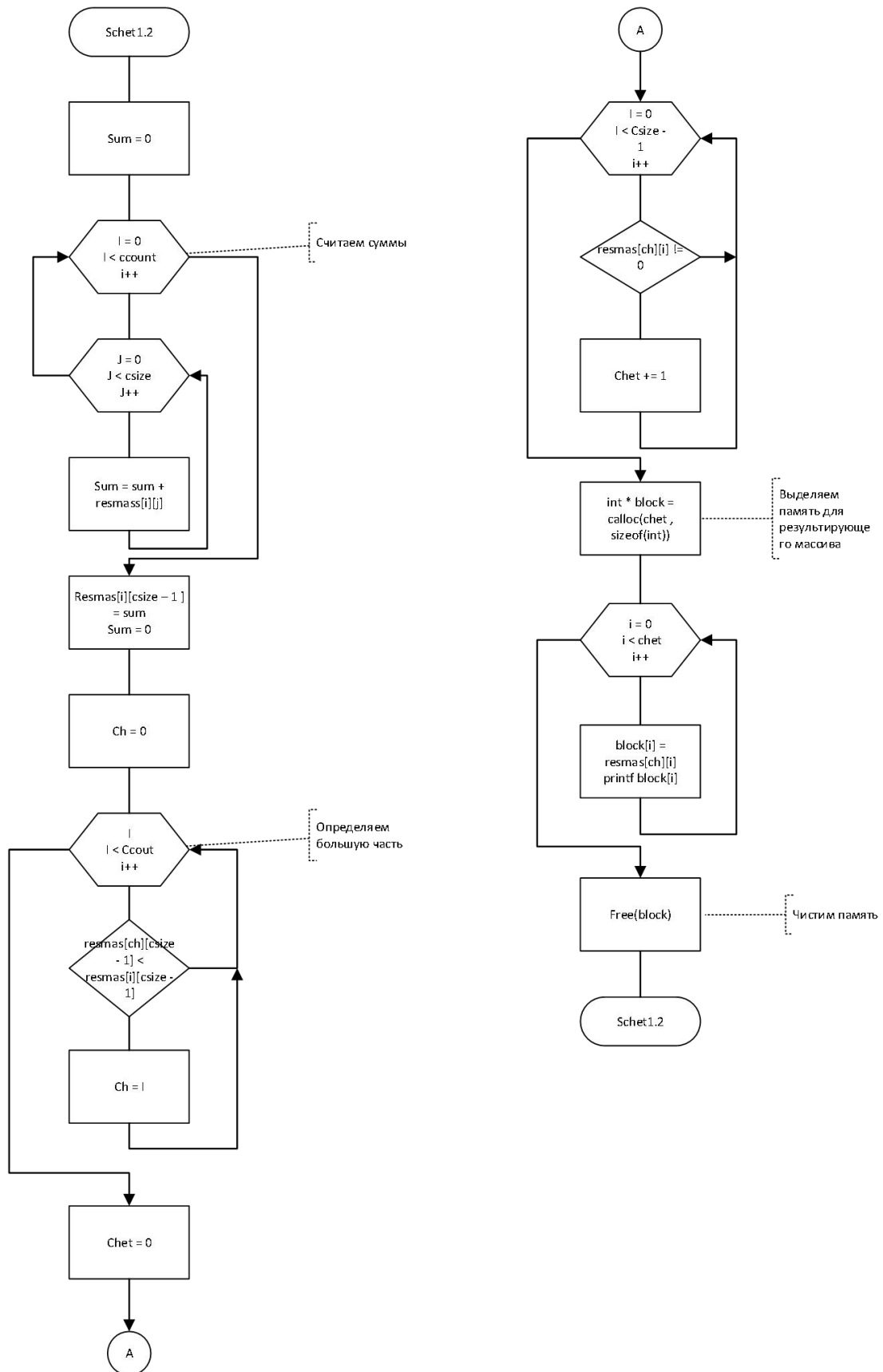


Рисунок 7 - продолжения основного цикла задачи 2

Задача 4

Призначення :

- упорядкування за зростанням елементів головної діагоналі матриці

Аргументи :

- 2 - вимірний масив

Опис роботи:

- елементи головної діагоналі переписуються у звичайний масив
- елементи сортуються за зростанням

Схема алгоритму (див рис 8 , 2) :

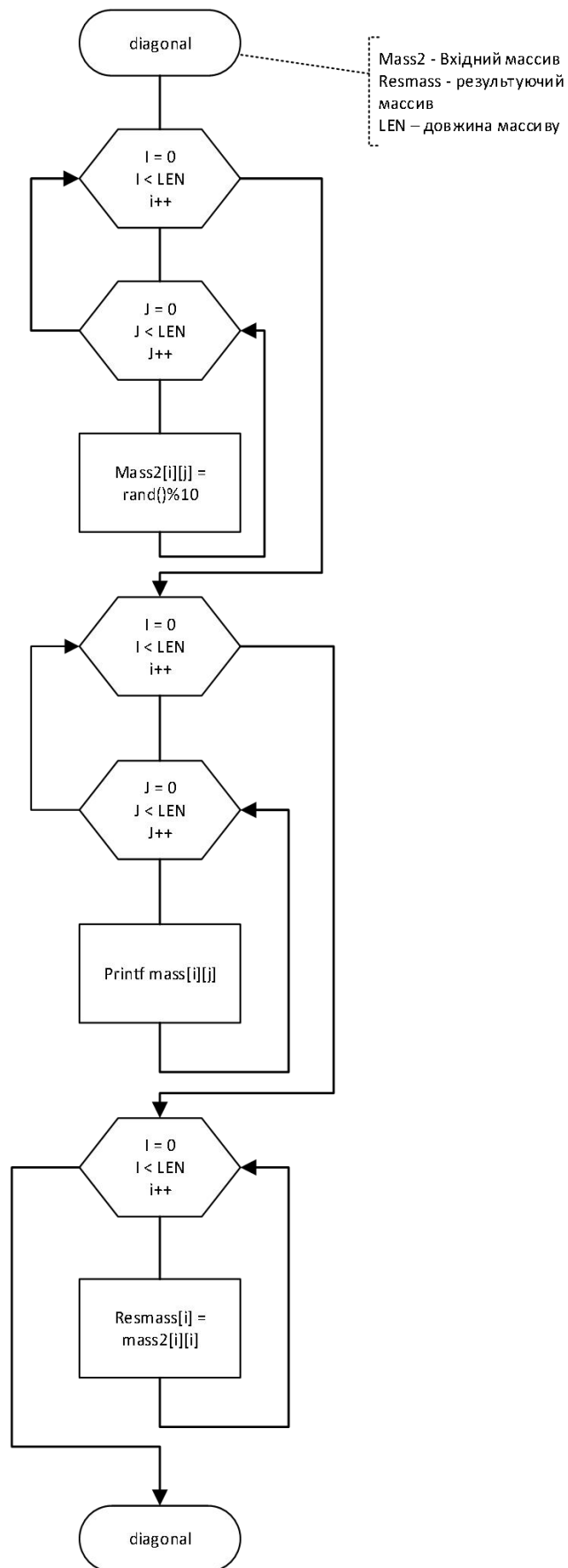


Рисунок 8 - основний алгоритм задачі

2.3 Важливі фрагменти програми

`int * block = calloc(b, sizeof(int));` - виділення пам'яті

`void schet (int block[], int mass[], int *chetmas);` - передача змінної та масиву в функцію

3 Варіанти використання

Для обчислення результатів ми використовуємо `gdb` - дебагер, влаштований в утиліту `gcc`.

Щоб подивитися на результат, ми повинні:

- зібрати проект за допомогою `make`
- запустити його бінарний файл у відлагоднику;
- вказати ім'я функції, як точку зупину щоб подивитись на значення змінних;
- запускаємо програму та вводимо: `"info locals"`;

Висновок :

В умовах даної лабораторної роботи ми навчилися працювати з динамічним відділенням пам'яті та вказівниками.