

А л г о р и т м и т а с т р у к т у р и д а н и х .

О с н о в и а л г о р и т м і з а ц і ї

Додаток 1

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра автоматизованих систем обробки інформації
і управління

Звіт

з лабораторної роботи №7
з дисципліни «Алгоритми та структури даних-1.
Основи алгоритмізації»
«Дослідження лінійного пошуку в послідовностях»
Варіант 8

Виконав
студент

ІП-13, Гончаров Євген Олександрович

(шифр, прізвище, ім'я, по батькові)

Перевірів

Наталія Вечерковська Сергіївна

(прізвище, ім'я, по батькові)

Алгоритми та структури даних.

Основи алгоритмізації

Лабораторна робота 7

Дослідження лінійного пошуку в послідовностях

Мета – дослідити методи послідовного пошуку у впорядкованих і неупорядкованих послідовностях та набутти практичних навичок їх використання під час складання програмних специфікацій.

Індивідуальне завдання

Варіант 8

Завдання

| 8 | $45 + 2 * i$ | $61 - 2 * i$ | Середнє арифметичне елементів, коди яких більше 55

1. Постановка задачі

Спочатку за допомогою ф-ції `getMass1` та `getMass2` згенеруємо 2 масиви. Потім за допомогою ф-ції `getElementsFromMassAbove55` знайдемо всі елементи обох масивів що більші за 55, та середнє арифметичне між їх елементами за допомогою ф-ї `getAverageElementsFromMasses`, запишемо результат у 3-й масив та виведемо його.

2. Побудова математичної моделі.

Складемо таблицю імен змінних.

Змінна	Тип	Ім'я	Призначення
Змінна <code>mass1</code>	Символьний[10]	<code>mass1</code>	Вхідні данні
Змінна <code>mass2</code>	Символьний[10]	<code>mass2</code>	Вхідні данні
Змінна <code>seed</code>	Цілочисельний	<code>seed</code>	Вхідні данні
Змінна <code>getAverageFromMasses</code>	Функція(символьний[])	<code>getAverageFromMasses(mass[])</code>	Обчислення
Змінна <code>getElementsFromMassAbove55</code>	Функція(символьний[])	<code>getElementsFromMassAbove55(mass[])</code>	Обчислення
Змінна <code>getMass1</code>	Функція(цілочисельний)	<code>getMass1(seed)</code>	Обчислення
Змінна <code>getMass2</code>	Функція(цілочисельний)	<code>getMass2(seed)</code>	Обчислення
Змінна <code>result</code>	Цілочисельний	<code>result</code>	Кінцеві данні
Змінна <code>getEqualFromMasses</code>	Символьний[]	<code>getEqualFromMasses(mass1, mass2)</code>	Обчислення

Опишемо знаходження довжини масиву через стандартний метод `length`

Опис підпрограм

`getMass1(int seed)` – генерація першого масиву

`getMass2(int seed)` – генерація другого масиву

`getAverageFromMasses (char[] mass)`– повертає середнє значення масиву

А л г о р и т м и т а с т р у к т у р и д а н и х .

О с н о в и а л г о р и т м і з а ц і ї

getElementsFromMassAbove55 – приймає масив та повертає лише ті елементи, коди яких більше за 55

Р о з в' я з а н н я

Програмні специфікації запишемо у псевдокодi та графічній формi у вигляді блок-схеми.

Крок 1. Визначемо основні дії.

Крок 2. Введемо змінну seed

Крок 3. Обрахуємо значення масиву mass1 та mass2

Крок 4. Знайдемо однакові елементи та запишемо їх

Крок 5. Знайдемо середнє арифметичне всіх елементів що більші за 55, результат запишемо

Крок 6. Виведемо результат

П с е в д о к о д

Основна програма:

Крок 1.

Початок

Визначемо основні дії.

Введемо змінну seed

Обрахуємо значення масиву mass1 та mass2

Знайдемо однакові елементи та запишемо їх

Знайдемо середнє арифметичне всіх елементів що більші за 55, результат запишемо

Виведемо результат

кінець

Крок 2.

Початок

Визначемо основні дії.

Введемо змінну seed

Обрахуємо значення масиву mass1 та mass2

Знайдемо однакові елементи та запишемо їх

Знайдемо середнє арифметичне всіх елементів що більші за 55, результат запишемо

Виведемо результат

кінець

Алгоритми та структури даних.

Основи алгоритмізації

Крок 3.

Початок

Визначемо основні дії.

Seed = 0

Обрахуємо значення масиву mass1 та mass2

Знайдемо однакові елементи та запишемо їх

Знайдемо середнє арифметичне всіх елементів що більші за 55, результат запишемо

Виведемо результат

Кінець

Крок 4.

Початок

Визначемо основні дії.

Seed = 0

mass1 = getMass1(seed)

mass2 = getMass2(seed)

Знайдемо однакові елементи та запишемо їх

Знайдемо середнє арифметичне всіх елементів що більші за 55, результат запишемо

Виведемо результат

Кінець

Крок 5.

Початок

Визначемо основні дії.

Seed = 0

mass1 = getMass1(seed)

mass2 = getMass2(seed)

equalElements = getEqualFromMasses(mass1, mass2)

Знайдемо середнє арифметичне всіх елементів що більші за 55, результат запишемо

Виведемо результат

кінець

Алгоритми та структури даних.

Основи алгоритмізації

Крок 6.

Початок

Визначемо основні дії.

Seed = 0

mass1 = getMass1(seed)

mass2 = getMass2(seed)

solution = getAverageFromMass(equalElements)

Вивести solution

кінець

Алгоритми та структури даних.

Основи алгоритмізації

Підпрограми:

getMass1(seed)

символьний mass1[10]

повторити для i від 0 до 9:

 mass1[i] = символний(2*(seed+i) + 45)

все повторити

повернути mass1

кінець getMass1

getMass2(seed)

символьний mass2[10]

повторити для i від 0 до 9:

 mass2[i] = символний(61 - 2*(seed+i))

все повторити

повернути mass2

кінець getMass2

getElementsFromMassAbove55(mass[])

цілочисельний count = 0

повторити для i від 1 до mass.length-1:

 якщо mass[i] > 55 то:

 count++

 інакше:

 mass[i] = 0

 все інакше

все повторити

символьний result[count]

повторити для i від 1 до mass.length-1:

 якщо mass[i] != 0 то:

 result[result.length-count] = mass[i]

 count--

 все якщо

все повторити

повернути result

кінець getElementsFromMassAbove55

Алгоритми та структури даних.

Основи алгоритмізації

getAverageFromMasses(mass)

символьний massAbove55[] =

getElementsFromMassAbove55(mass);

цілочисельний result = 0;

якщо massAbove55.length == 0 то:

повернути result

інакше:

повторити для i від 0 до massAbove55.length-1:

result += massAbove55[i]

все повторити

все інакше

повернути result[]/massAbove55.length

кінець **getAverageElementsFromMasses**

getEqualFromMasses(mass1, mass2)

цілочисельний count = 0

повторити для i від 0 до mass1.length :

повторити для j від 0 до mass2.length :

якщо mass1[i] == mass2[j] то:

count++

все якщо

все повторити

все повторити

цілочисельний equalElements[count]

повторити для i від 0 до mass1.length :

повторити для j від 0 до mass2.length :

якщо mass1[i] == mass2[j] то:

equalElements[count-1] = mass1[i]

count--

все якщо

все повторити

все повторити

повернути equalElements

кінець **getEqualFromMasses**

Алгоритми та структури даних.

Основи алгоритмізації

Блок-схема

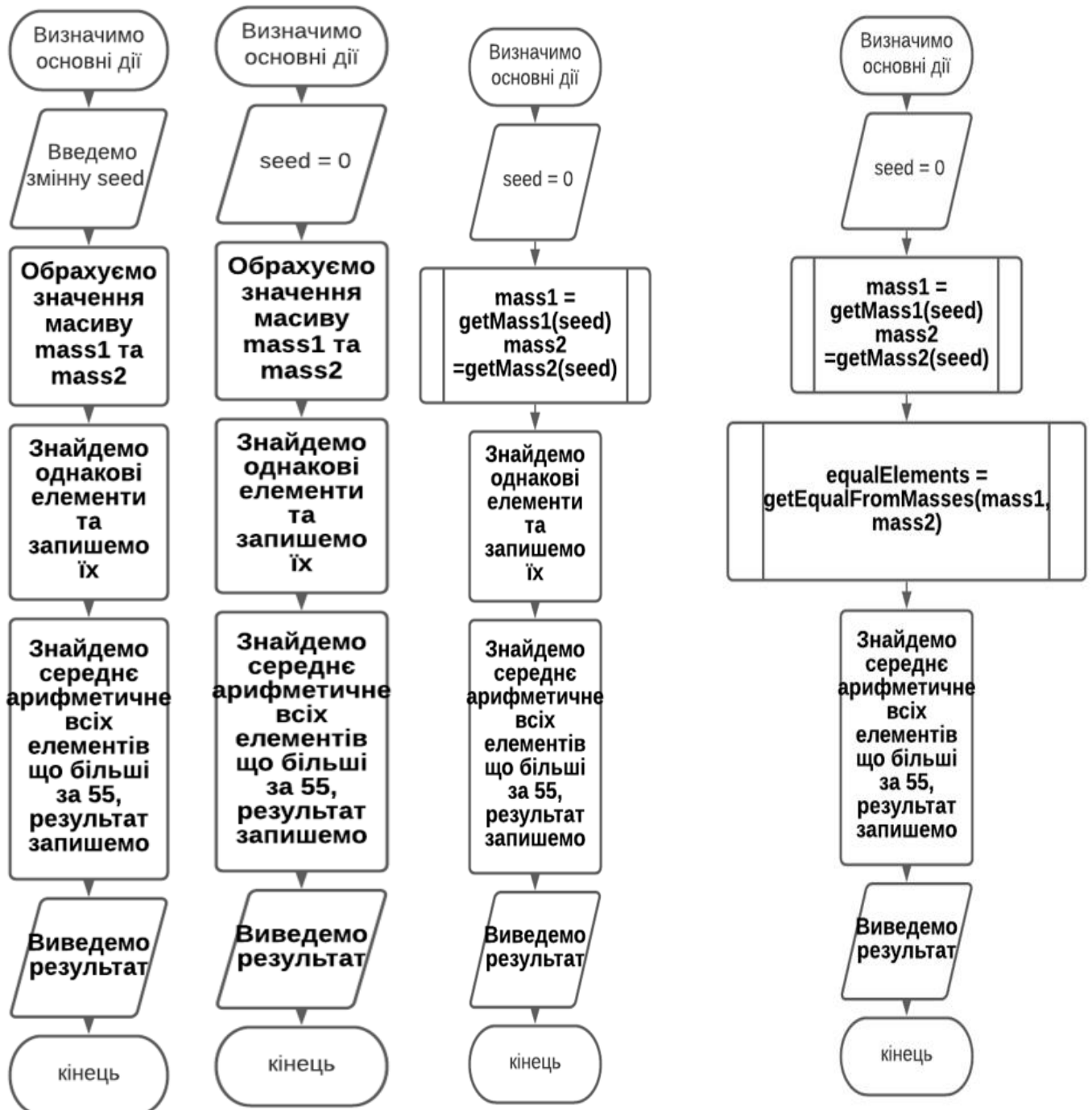
Основна програма:

Крок 1, 2

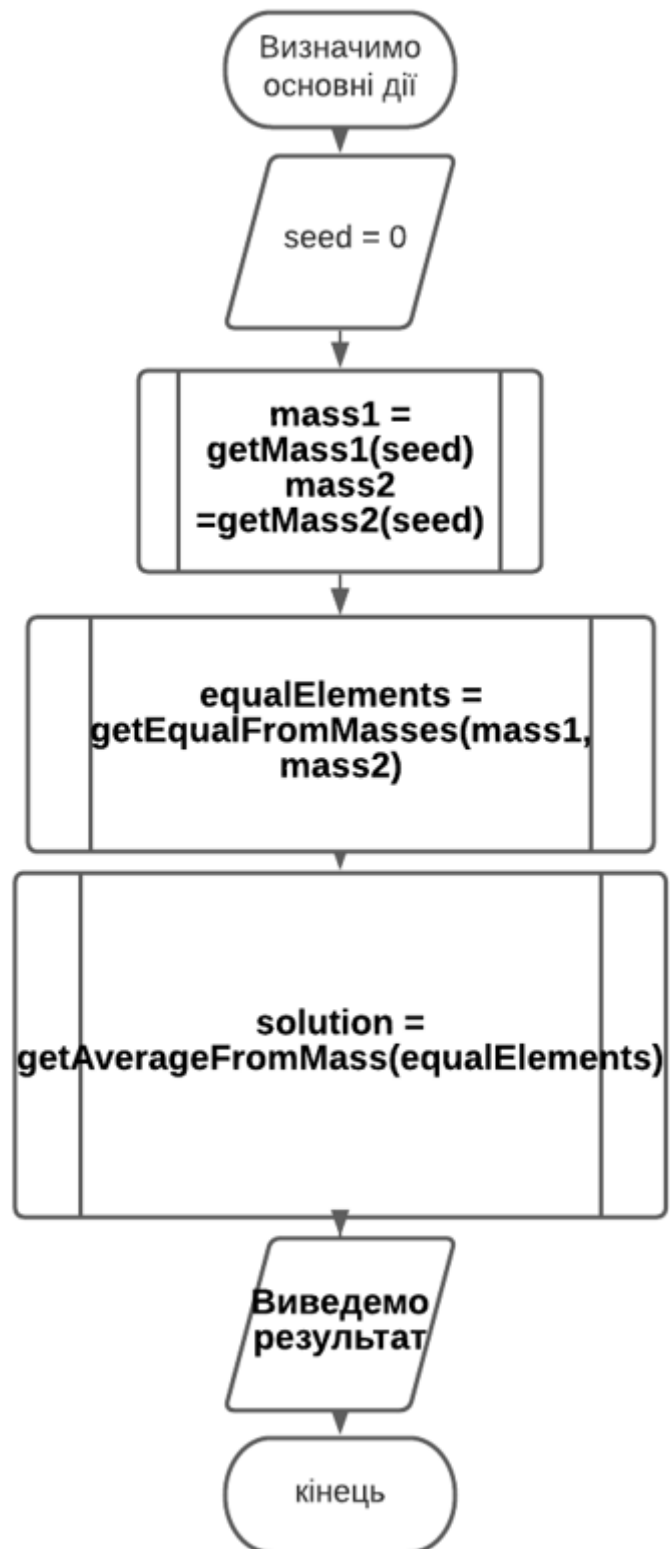
Крок 3

Крок 4

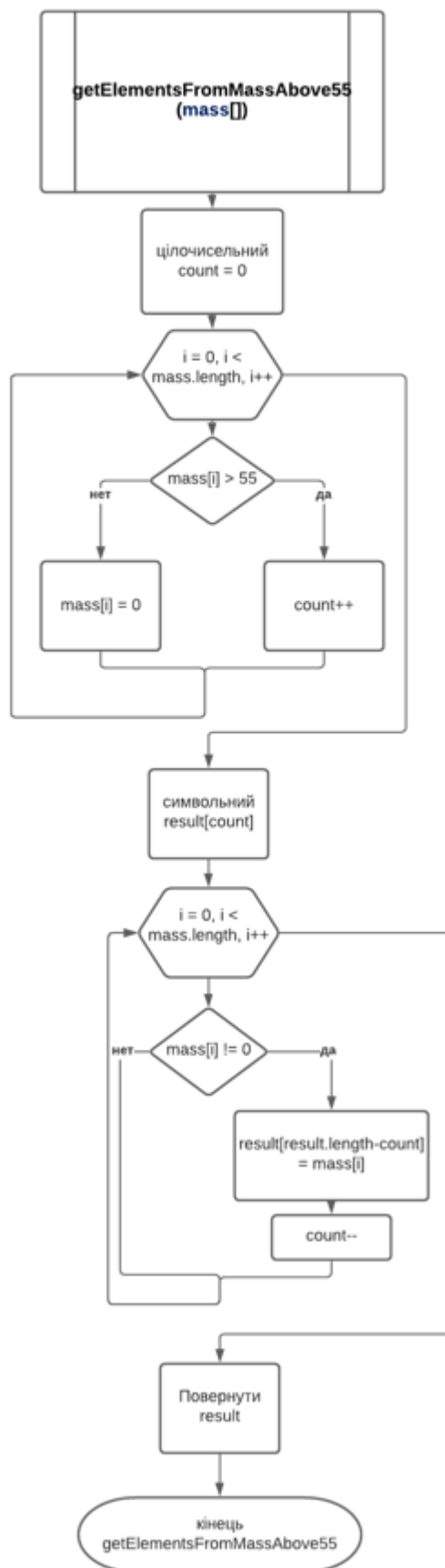
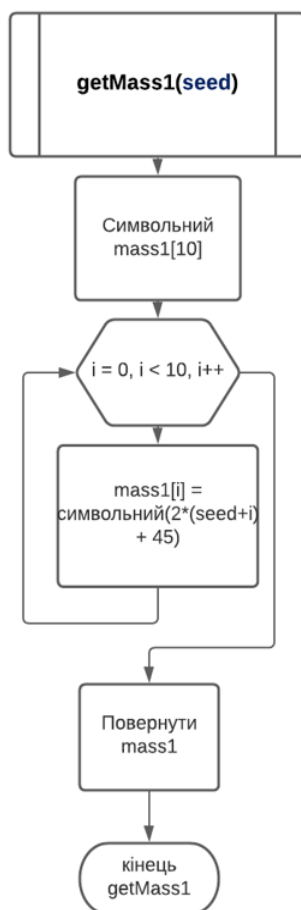
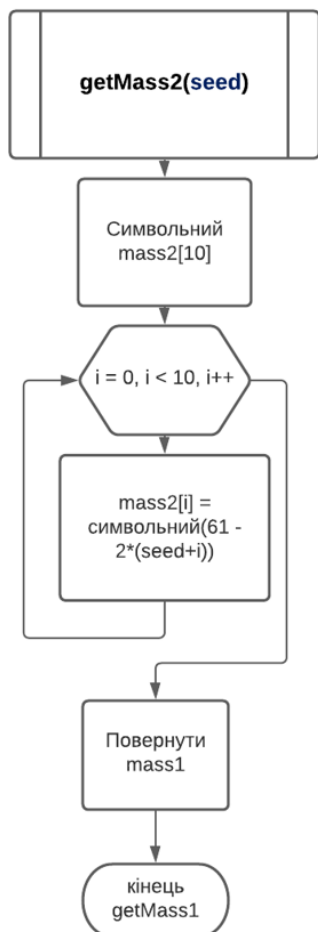
Крок 5

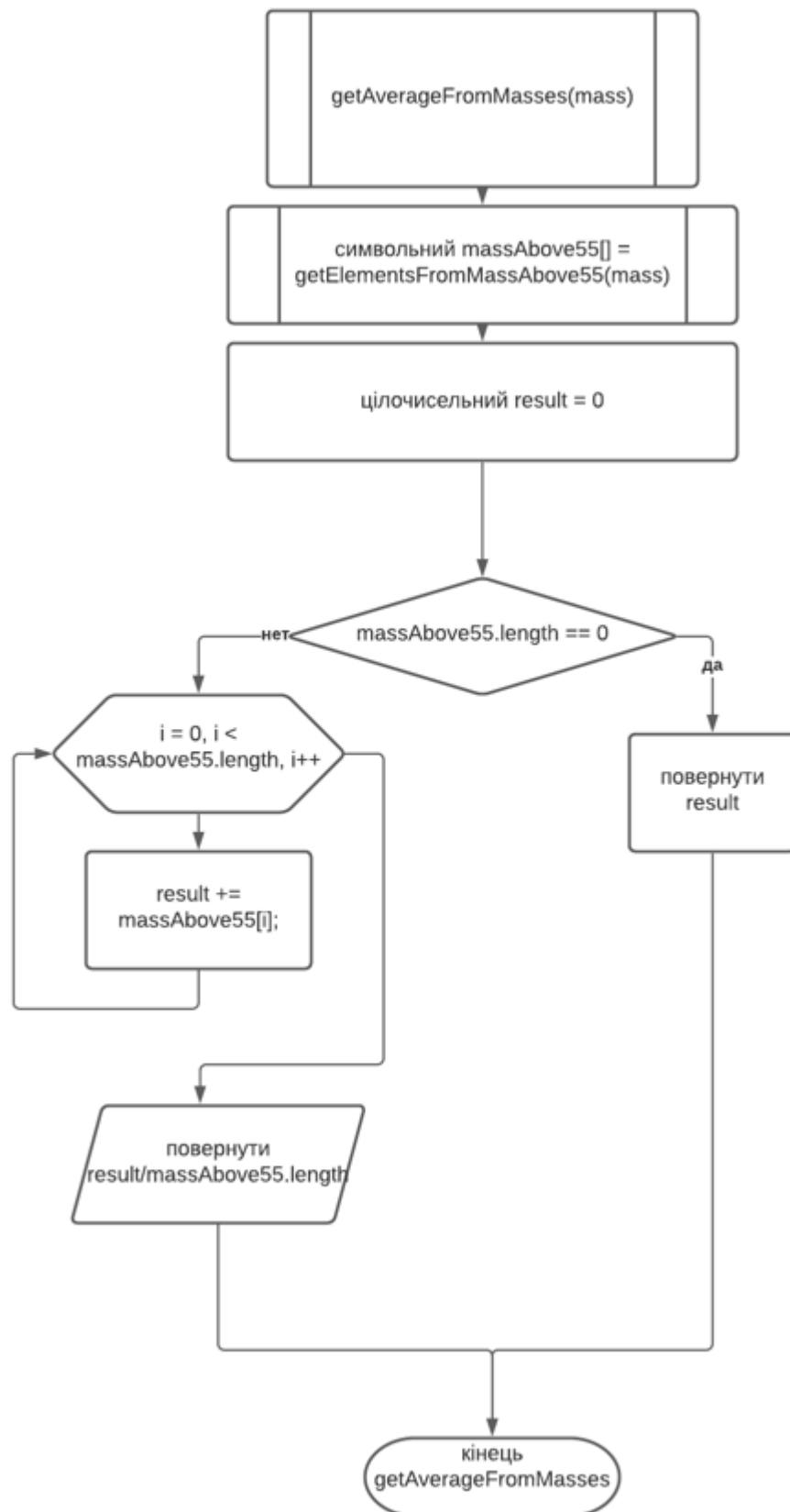


К р о к 6

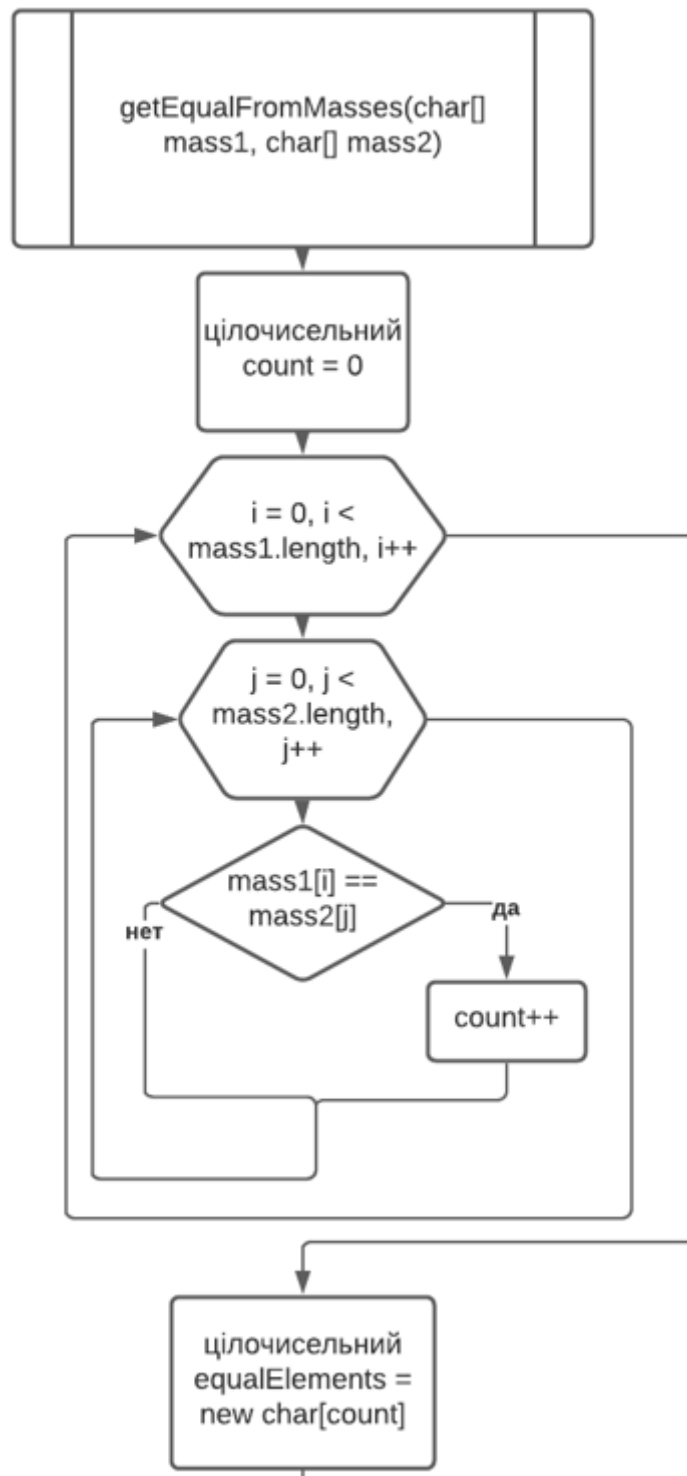


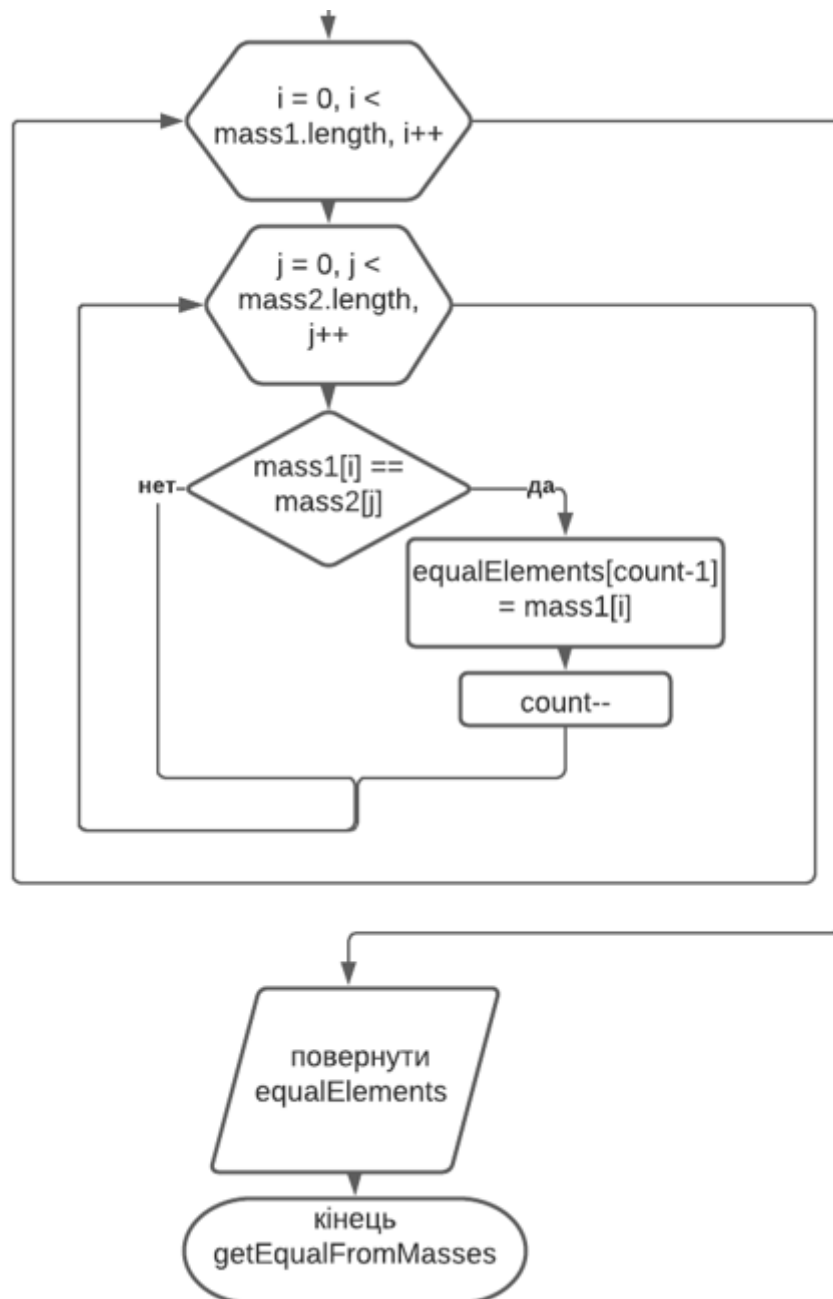
Підпрограми:





Основи алгоритмізації





Алгоритми та структури даних.

Основи алгоритмізації

Тестування

Введіть насіння для створення масивів

1

Перший масив

47 49 51 53 55 57 59 61 63 65

Другий масив

59 57 55 53 51 49 47 45 43 41

Рівні елементи масивів чий код більше за 55

59 57 0 0 0 0 0

Середнє значення спільних елементів чий код більше за 55

58

К о д

```
1  import java.util.Scanner;
2
3  public class Laba7ASD {
4      public static void main(String[] args) {
5          System.out.println("Введіть насіння для створення масивів");
6          Scanner in = new Scanner(System.in);
7          int seed = in.nextInt();
8          char[] mass1 = getMass1(seed);
9          char[] mass2 = getMass2(seed);
10         char[] equalElements = getEqualFromMasses(mass1, mass2);
11         int solution = getAverageFromMass(equalElements);
12
13         printTests(mass1, mass2, solution, equalElements);
14     }
```

Алгоритми та структури даних.

Основи алгоритмізації

```
17     public static int getAverageFromMass(char[] mass){
18         char[] massAbove55 = getElementsFromMassAbove55(mass);
19         int result = 0;
20         if(massAbove55.length == 0){
21             return result;
22         }
23         for(int i = 0; i < massAbove55.length; i++){
24             result += massAbove55[i];
25         }
26         return result/massAbove55.length;
27     }
```

```
40     @ public static char[] getElementsFromMassAbove55(char[] mass){
41         int count = 0;
42         for(int i = 0; i < mass.length; i++){
43             if(mass[i] > 55){ count++; }
44             else{ mass[i] = 0; }
45         }
46         char[] result = new char[count];
47         for(int i = 0; i < mass.length; i++){
48             if (mass[i] != 0){
49                 result[result.length - count] = mass[i];
50                 count--;
51             }
52         }
53         return result;
54     }
```

Алгоритми та структури даних.

Основи алгоритмізації

```
55  @   public static char[] getMass1(int seed){
56      char[] mass1 = new char[10];
57      for(int i = 0; i < 10; i++){
58          mass1[i] = (char)(2 * (seed + i) + 45);
59      }
60      return mass1;
61  }
62  @   public static char[] getMass2(int seed){
63      char[] mass2 = new char[10];
64      for(int i = 0; i < 10; i++){
65          mass2[i] = (char)(61 - (2 * (seed + i)));
66      }
67      return mass2;
68  }
```

```
57  @   public static char[] getEqualFromMasses(char[] mass1, char[] mass2){
58      int count = 0;
59      for(int i = 0; i < mass1.length; i++){
60          for(int j = 0; j < mass2.length; j++){
61              if(mass1[i] == mass2[j]){count++;}
62          }
63      }
64
65      char[] equalElements = new char[count];
66
67      for(int i = 0; i < mass1.length; i++){
68          for(int j = 0; j < mass2.length; j++){
69              if((int)mass1[i] == (int)mass2[j]){
70                  equalElements[count-1] = mass1[i];
71                  count--;
72              }
73          }
74      }
75      return equalElements;
76  }
```


Алгоритми та структури даних.

Основи алгоритмізації

```
77 public static void printTests(char[] mass1, char[] mass2, int solution, char[] equalElements){
78     System.out.println("Перший масив");
79     for(int i = 0; i < 10; i++){
80         System.out.print((int)mass1[i] + " ");
81     }
82     System.out.println();
83     System.out.println("Другий масив");
84     for(int i = 0; i < 10; i++){
85         System.out.print((int)mass2[i] + " ");
86     }
87     System.out.println();
88     System.out.println("Рівні елементи масивів чий код більше за 55");
89     for(int i = 0; i < equalElements.length; i++){
90         System.out.print((int)equalElements[i] + " ");
91     }
92     System.out.println();
93     System.out.println("Середнє значення спільних елементів чий код більше за 55");
94     System.out.println(solution);
95 }
```

Висновки

Ми дослідили методи послідовного пошуку у впорядкованих і неупорядкованих послідовностях та набути практичних навичок їх використання під час складання програмних специфікацій.

В результаті виконання лабораторної роботи ми отримали алгоритм для знаходження середнього арифметичного елементів із двох списків які більші по коду за 55, декомпозували задачу на 6 кроків: визначили основні дії, ввели наслідки для генерації масивів, згенерували масиви, знайшли всі елементи масивів і обчислили їх середнє арифметичне та вивели результат.