

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 №8
з дисципліни «Алгоритми та структури даних-1.
Основи алгоритмізації»
«Дослідження алгоритмів пошуку та сортування»
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 8

Дослідження алгоритмів пошуку та сортування

Мета – дослідити алгоритми пошуку та сортування, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

Індивідуальне завдання

Варіант 8

Завдання

№ варіанта	Розмірність	Тип даних	Обчислення значень елементів одновимірного масиву
8	4 x 6	Цілий	Із максимальних значень елементів стовпців двовимірного масиву. Відсортувати обміном за спаданням.

1. Постановка задачі

Введемо висоту та ширину двовимірного масиву, за допомогою спеціальної ф-ї заповнимо цей масив випадковими числами, наступним кроком за допомогою спеціальної ф-ї згенеруємо масив який складається із найбільших елементів стовбців першого масиву та за допомогою спеціальної ф-ї відсортуємо його у порядку спадання.

2. Побудова математичної моделі.

Складемо таблицю імен змінних.

Змінна	Тип	Ім'я	Призначення
Змінна height	Цілочисельний	height	Вхідні данні
Змінна weight	Цілочисельний	weight	Вхідні данні
Змінна array	Цілочисельний[][]	array[][]	Проміжні данні
Змінна result	Цілочисельний[]	result[]	Проміжні та кінцеві данні
Змінна massBubbleSort	Функція(цілочисельний[])	massBubbleSort (array[])	Обчислення
Змінна generateDoubleMass	Функція(цілочисельний, цілочисельний)	generateDoubleMass (height, weight)	Обчислення
Змінна getMaxFromColumns	Функція(цілочисельний[][])	getMaxFromColumns (array)	Обчислення

Опишемо генерацію випадкового цілого числа через базову функцію Math.random()

Опишемо отримання довжини масиву стандартною функцією Array.length

Опис підпрограм

massBubbleSort (array[]) – бульбашкове сортування масиву(у порядку спадання)

generateDoubleMass (height, weight) – генерація двовимірного 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

getMaxFromColumns (array) – генерація масиву із найбільшх елементів рядка двовірного масиву

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 змінні height та weight

Крок 3. Запишемо в змінну array результат виконання generateDoubleMass(height, weight)

Крок 4. Запишемо у змінну result результат виконання getMaxFromColumns(array)

Крок 5. Відсортуємо result за допомогою massBubbleSort(result)

Крок 6. Виведемо result

П с е в д о к о д

Основна програма:

Початок

Визначемо основні дії

Введення height

Введення weight

array = generateDoubleMass(height, weight)

result = getMaxFromColumns(array)

result = massBubbleSort(result)

Виведення result

Кінець

Підпрограми:

```
generateDoubleMass(height, weight)
    цілочисельний array[][]
    повторити для i від 0 до height:
        повторити для j від 0 до weidht:
            array[i][j] = Math.random()*99 + 1
        все повторити
    все повторити
    повернути array
кінець generateDoubleMass
```

```
getMaxFromColumns(array[][])
    цілочисельний result[array[0].length]
    повторити для i від 0 до array[0].length:
        result[i] = array[0][i]
    все повторити

    повторити для j від 0 до array[0].length:
        повторити для i від 0 до array.length:
            якщо array[i][j] > result[j] то:
                result[j] = array[i][j]
            все якщо
        все повторити
    все повторити
    повернути result
кінець getMaxFromColumns
```

Алгоритми та структури даних.

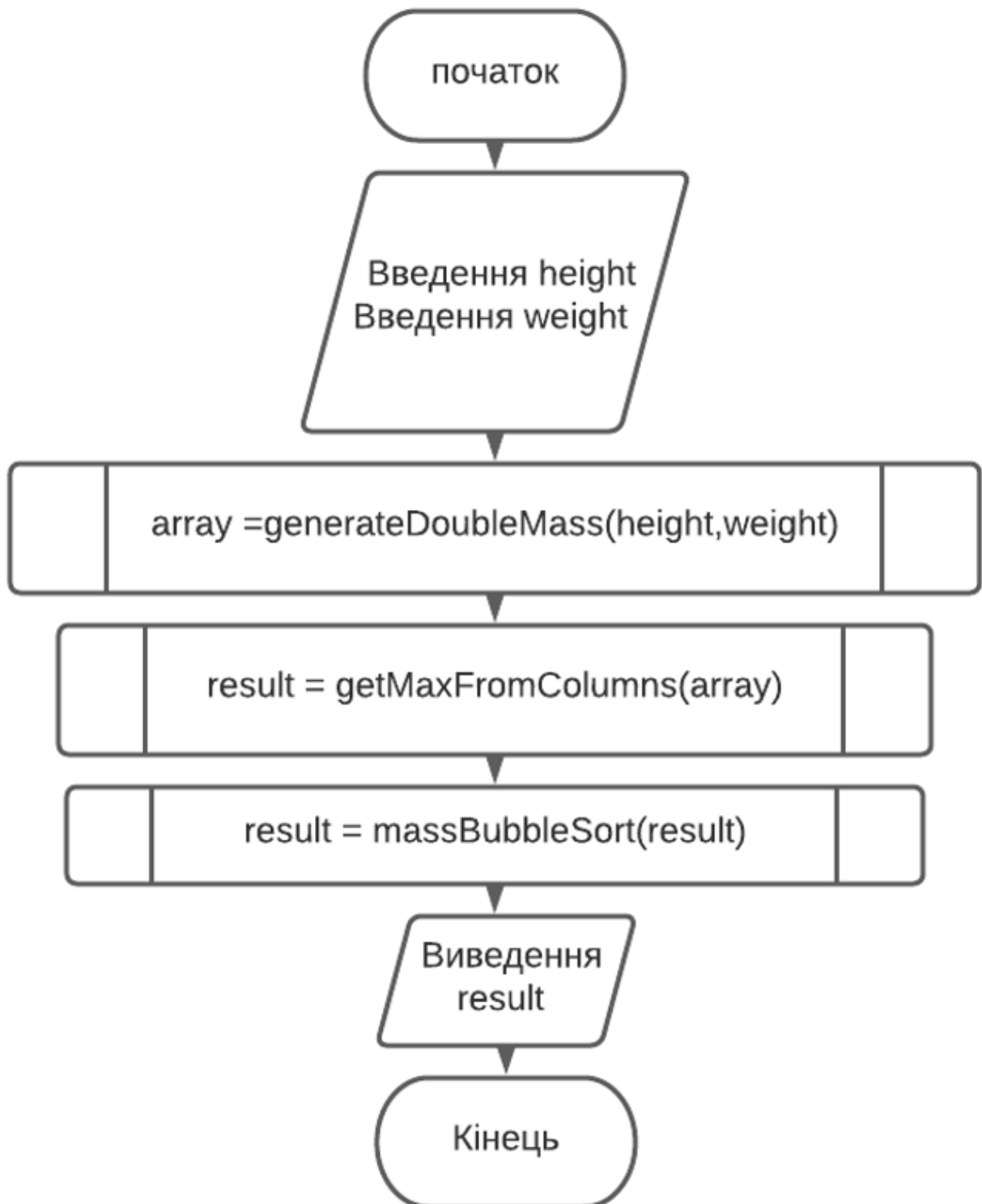
Основи алгоритмізації

```
massBubbleSort(array[])
  цілочисельний counter = 1
  поки counter != 0 повторити:
    counter = 0
    для i від 0 до array.length-1:
      якщо array[i] < array[i+1] то:
        array[i] += array[i+1]
        array[i+1] = array[i] - array[i+1]
        array[i] = array[i] - array[i+1]
        counter++
    все якщо
  все повторити
  повернути array
кінець massBubbleSort
```

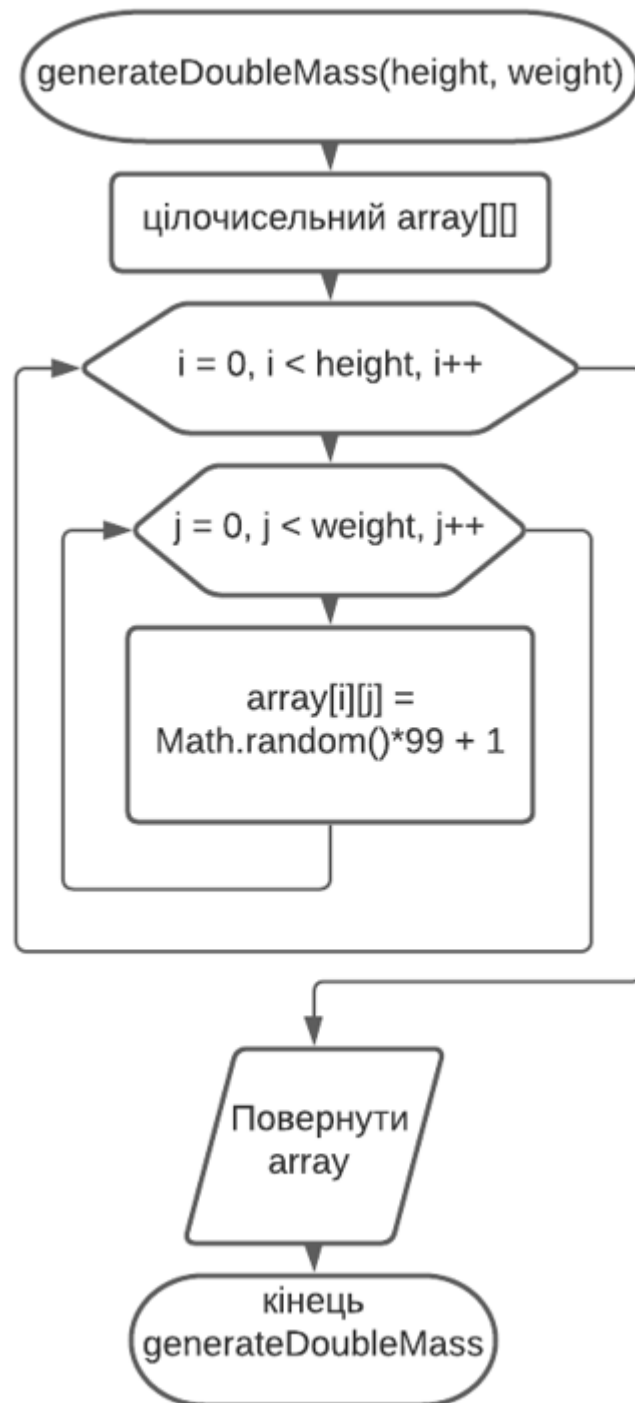
Основи алгоритмізації

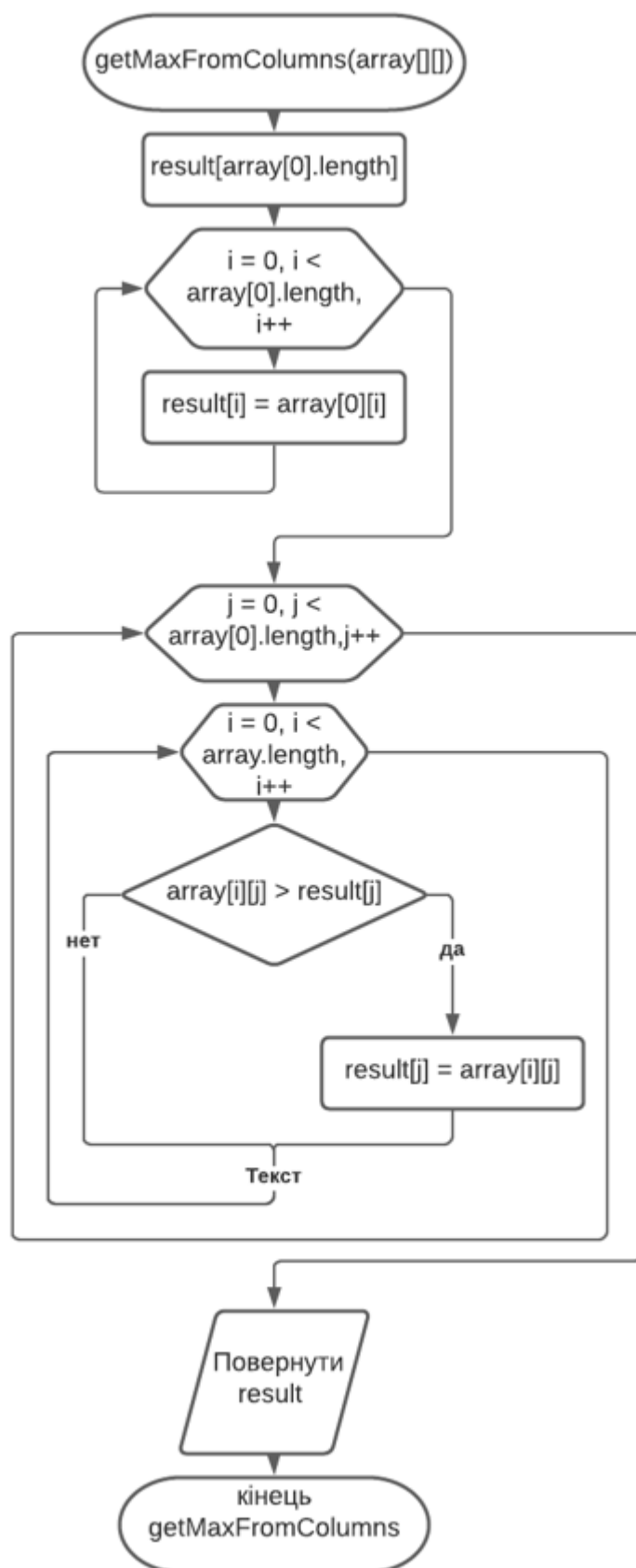
Блок-схема

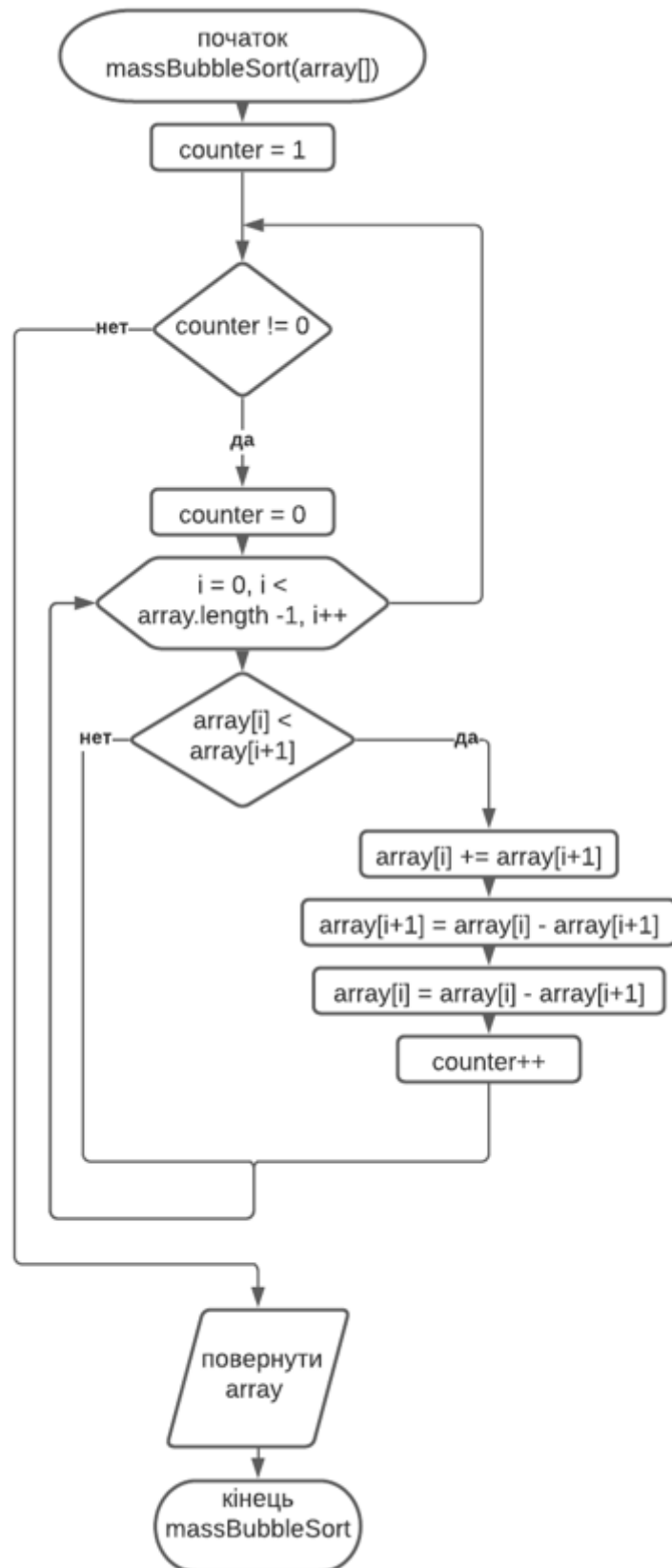
Основна програма:



Підпрограми:







Алгоритми та структури даних.

Основи алгоритмізації

Тестування

```
Введіть к-ть рядків: 4
Введіть к-ть стовбців: 6
Згенерований масив:
| 84 60 70 71 47 36 |
| 66 47 62 66 32 57 |
| 70 70 26 36 56 28 |
| 20 45 16 34 92 86 |

Масив найбільших елементів зі стовбців
[ 84 70 70 71 92 86 ]
Відсортований масив найбільших елементів зі стовбців
[ 92 86 84 71 70 70 ]

Process finished with exit code 0
```

К о д

```
1  import java.util.Scanner;
2
3  public class LabaASD_8 {
4
5      public static void main(String[] args) {
6          Scanner in = new Scanner(System.in);
7
8          System.out.print("Введіть к-ть рядків: ");
9          int height = in.nextInt();
10         System.out.print("Введіть к-ть стовбців: ");
11         int weight = in.nextInt();
12
13         int[][] array = generateDoubleMass(height, weight);
14
15         int[] result = getMaxFromColumns(array);
16         System.out.println("Згенерований масив:");
17         printDoubleMass(array);
18         System.out.println();
19         System.out.println("Масив найбільших елементів зі стовбців");
20         printSoloMass(result);
21         System.out.println("Відсортований масив найбільших елементів зі стовбців");
22         printSoloMass(massBubbleSort(result));
23
24     }
```

Алгоритми та структури даних.

Основи алгоритмізації

```
26
27 public static int[] massBubbleSort(int[] array){
28     int counter = 1;
29     while(counter != 0){
30         counter = 0;
31         for(int i = 0; i < array.length - 1; i++){
32             if(array[i] < array[i+1]){
33                 array[i] += array[i+1];
34                 array[i+1] = array[i] - array[i+1];
35                 array[i] = array[i] - array[i+1];
36                 counter++;
37             }
38         }
39     }
40     return array;
41 }
42
```

```
42
43 @ public static int[][] generateDoubleMass(int height, int weight){
44     int[][] array = new int[height][weight];
45
46     for(int i = 0; i < height; i++){
47         for(int j = 0; j < weight; j++){
48             array[i][j] = (int)(Math.random()*(99)) + 1;
49         }
50     }
51     return array;
52 }
53
```

Алгоритми та структури даних.

Основи алгоритмізації

```
54
55 @ public static int[] getMaxFromColumns(int[][] array){
56     int[] result = new int[array[0].length];
57     for(int i = 0; i < array[0].length; i++){
58         result[i] = array[0][i];
59     }
60
61     for(int j = 0; j < array[0].length; j++){
62         for(int i = 0; i < array.length; i++){
63             if(array[i][j] > result[j]){
64                 result[j] = array[i][j];
65             }
66         }
67     }
68     return result;
69 }
```

```
70
71 @ public static void printDoubleMass(int[][] array){
72     for(int i = 0; i < array.length; i++){
73         System.out.print("| ");
74         for(int j = 0; j < array[i].length; j++){
75             if(array[i][j] < 10){
76                 System.out.print(array[i][j] + " ");
77             }else {
78                 System.out.print(array[i][j] + " ");
79             }
80         }
81         System.out.println("|");
82     }
83 }
84
```

Алгоритми та структури даних.

Основи алгоритмізації

```
85 @ public static void printSoloMass(int[] array){  
86     System.out.print("[ ");  
87     for(int i = 0; i < array.length; i++){  
88         System.out.print(array[i] + " ");  
89     }  
90     System.out.println("]");  
91 }
```

Висновки

Ми дослідили алгоритми пошуку та сортування, та набути практичних навичок їх використання під час складання програмних специфікацій.

В результаті виконання лабораторної роботи ми отримали алгоритм для знаходження відсортованого масиву найбільших елементів зі стовбців, декомпозиували задачу на 6 кроків: визначили основні дії, ввели данні для генерації масиву, згенерували масив, знайшли всі найбільші елементи стовбців і записали їх у інший масив та відсортували його у порядку спадання, результат вивели.