

Міністерство освіти і науки України
Національний технічний університет України «Київський політехнічний
інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

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

Варіант __16__

Виконав студент __ІП-15, Куманецька І. В.____
Перевірив _Вечерковська А. С._____

Київ 2021

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 16

Завдання

Розробити алгоритм та написати програму, яка складається з наступних дій:

1. Опису змінної індексованого типу (двовимірний масив) згідно з варіантом (табл. 1).
2. Ініціювання змінної, що описана в п.1 даного завдання.
3. Створення нової змінної індексованого типу (одновимірний масив) та її ініціювання значеннями, що обчислюються згідно з варіантом (табл. 1).

16	5 x 7	Цілий	Із середнього арифметичного від'ємних значень елементів стовпців двовимірного масиву. Відсортувати обміном за спаданням.
----	-------	-------	--

Постановка задачі

Створити двовимірний масив з довільними елементами. Для кожного з семи стовпців вирахувати середнє арифметичне від'ємних значень, створити з них новий одновимірний масив. У другому масиві відсортувати елементи обміном за спаданням.

Побудова математичної моделі

Змінна	Тип	Ім'я	Призначення
Перший масив	Ціле число	arr	Вхідні дані
Константа, кількість рядків у матриці	Ціле число	m	Вхідні дані
Константа, кількість стовпців у матриці	Ціле число	n	Вхідні дані
Другий масив	Ціле число	means	Кінцеві дані
Початковий двовимірний масив для get_means	Ціле число	matrix	Проміжні дані
Кількість рядків у matrix	Ціле число	lines	Проміжні дані
Кількість стовпців у matrix	Ціле число	columns	Проміжні дані
Середнє арифметичне від'ємних чисел стовпця	Дійсне число	alg_mean	Проміжні дані
Кількість від'ємних елементів стовпця	Ціле число	amount	Проміжні дані
Кінцевий масив, який повертає get_means	Ціле число	negative	Проміжні дані
Масив, елементи якого відсортовує sort_array	Дійсне число	data	Проміжні дані
Кількість елементів data	Ціле число	number	Проміжні дані
Ітератор циклу	Ціле число	i	Проміжні дані
Ітератор циклу	Ціле число	j	Проміжні дані
Стан масиву	Булевий	changed	Проміжні дані

Створюємо початковий масив. У функції `get_means(matrix, lines, columns)` перебираються значення усіх стовпців матриці, розраховується середнє арифметичне від'ємних елементів кожного стовпця. Результатом роботи функції

є масив з цих середніх значень. Функція `sort_array(data, number)` сортує масив методом обміну та повертає його.

Через `a += b` позначається операція `a = a + b`, через `a /= b` позначається операція `a /= b`.

Розв'язання

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

Крок 1. Визначення основних дій.

Крок 2. Деталізація генерації другого масиву за допомогою підпрограми.

Крок 3. Сортування елементів другого масиву.

Псевдокод

початок

`m := 5`

`n := 7`

введення `arr`

`means := sort_array(get_means(arr, m, n), n)`

кінець

Підпрограми

`get_means(matrix, lines, columns)`

початок

повторити для `i` від 1 до `columns`

`alg_mean := 0`

amount := 0

повторити для j від 1 до lines

якщо matrix[j][i] < 0

то alg_mean += matrix[j][i]

amount += 1

все якщо

все повторити

якщо amount > 0

то alg_mean /= amount

все якщо

negative[i] := alg_mean

все повторити

повернути negative

кінець

sort_array(data, number)

початок

повторити

changed := False

повторити для i від 1 до number-1

якщо data[i] < data[i+1]

то data[i] := data[i] + data[i+1]

$\text{data}[i+1] := \text{data}[i] - \text{data}[i+1]$

$\text{data}[i] := \text{data}[i] - \text{data}[i+1]$

$\text{changed} := \text{True}$

все якщо

все повторити

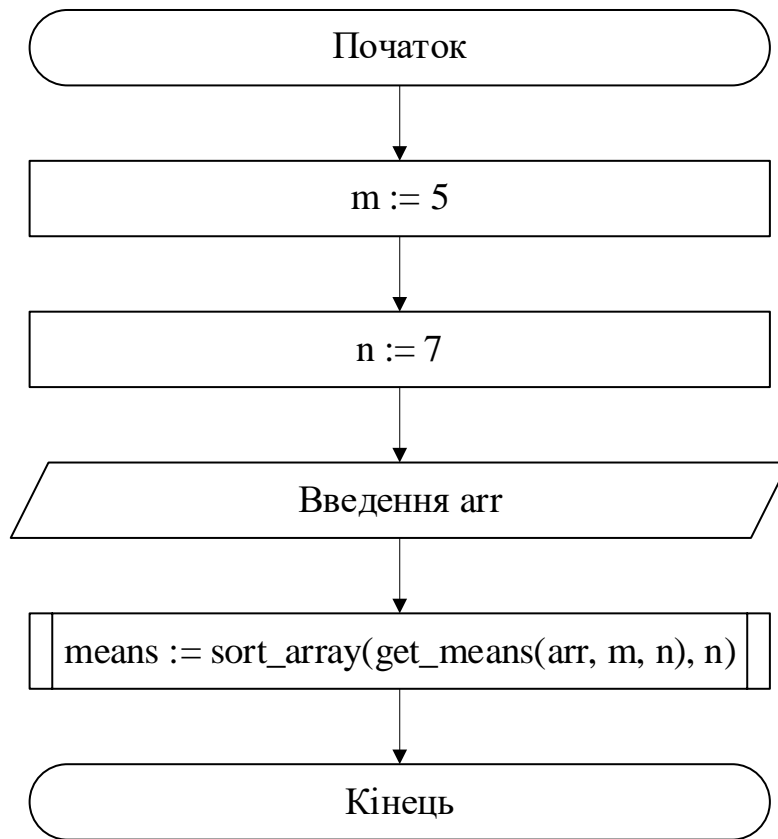
поки changed

все повторити

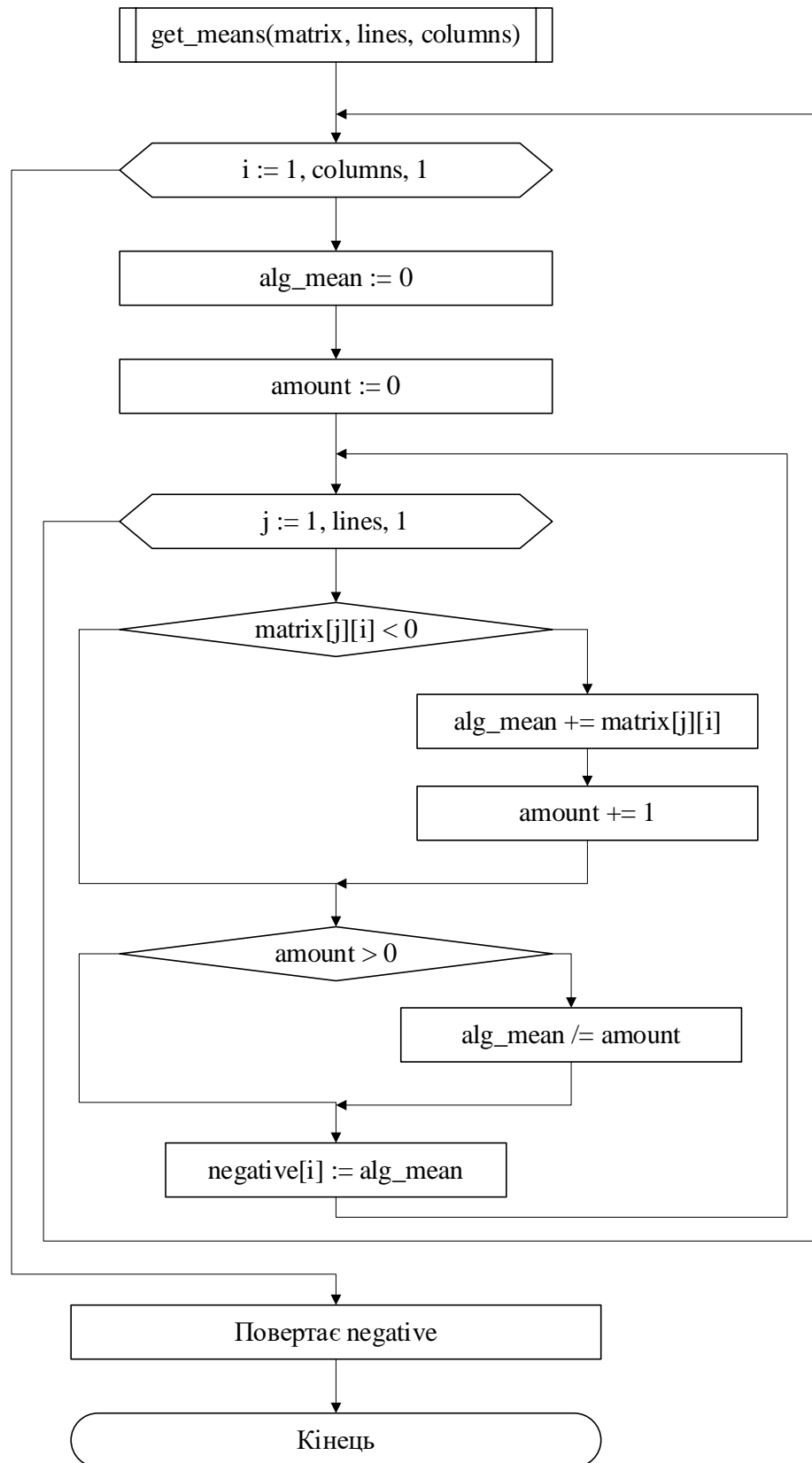
повернути data

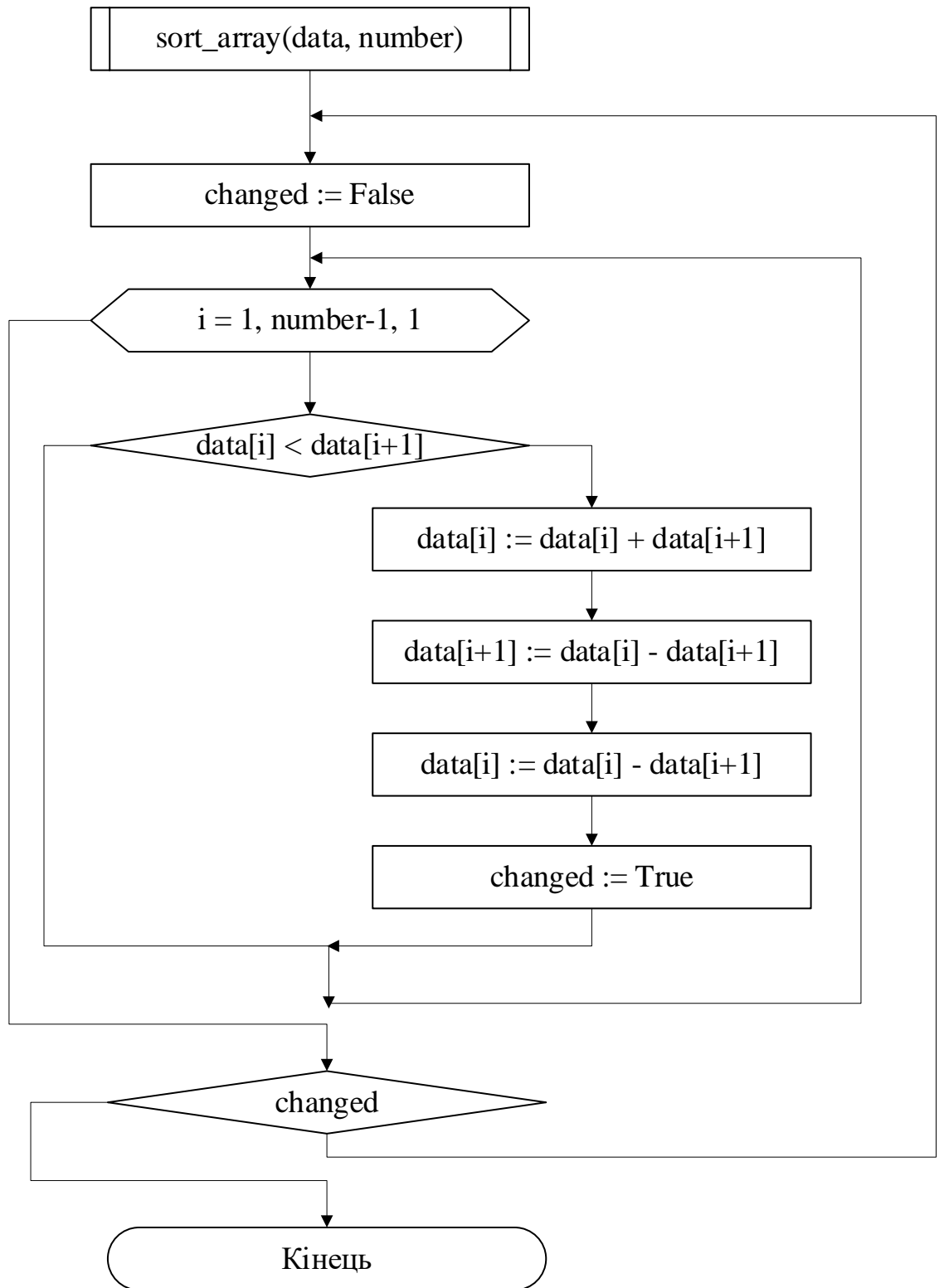
кінець

Блок-схема



Підпрограми





Код

```
#include <stdio.h>
#include <math.h>
#include <iostream>

using namespace std;
float* get_means(int**, int, int);
float* sort_array(float*, int);

int main()
{
    const int m = 5, n = 7;
    int** arr = new int* [m];
    for (int i = 0; i < m; i++)
    {
        arr[i] = new int[n];
    }

    srand(time(NULL));

    cout << "The main matrix: " << endl; //заповнення та виведення матриці
    for (int i = 0; i < m; i++)
    {
        for (int j = 0; j < n; j++)
        {
            arr[i][j] = rand() % 200 - 100;
            cout << arr[i][j] << " ";
        }
        cout << endl;
    }

    float *means = sort_array(get_means(arr, m, n), n);
    cout << "The negative means: "; //виведення другого масиву
    for (int i = 0; i < n; i++)
    {
        cout << means[i] << " ";
    }
}
```

```
float* get_means(int** matrix, const int lines, const int columns)
{
    float* negative = new float [columns];

    for (int i = 0; i < columns; i++)
    {
        float alg_mean = 0, amount = 0;

        for (int j = 0; j < lines; j++)
        {
            if (matrix[j][i] < 0)
            {
                alg_mean += (float)matrix[j][i];
                amount += 1;
            }
        }

        if (amount > 0)
        {
            alg_mean /= amount;
        }

        negative[i] = alg_mean;
    }

    return negative;
}
```

```

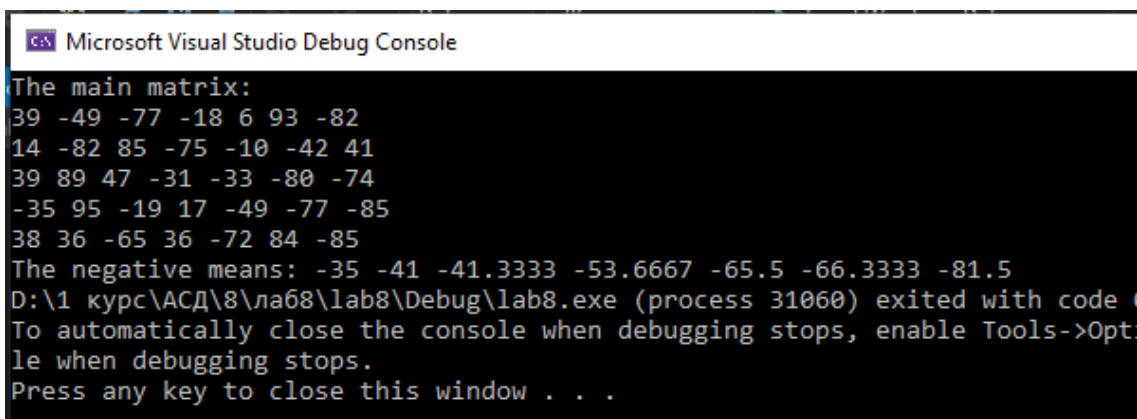
float* sort_array(float* data, int number)
{
    bool changed;
    do
    {
        changed = false;

        for (int i = 0; i < number - 1; i++)
        {
            if (data[i] < data[i + 1])
            {
                data[i] = data[i] + data[i + 1];
                data[i + 1] = data[i] - data[i + 1];
                data[i] = data[i] - data[i + 1];
                changed = true;
            }
        }
    } while (changed);

    return data;
}

```

Тестування коду



```

Microsoft Visual Studio Debug Console
The main matrix:
39 -49 -77 -18 6 93 -82
14 -82 85 -75 -10 -42 41
39 89 47 -31 -33 -80 -74
-35 95 -19 17 -49 -77 -85
38 36 -65 36 -72 84 -85
The negative means: -35 -41 -41.3333 -53.6667 -65.5 -66.3333 -81.5
D:\1 курс\АСД\8\лаб8\lab8\Debug\lab8.exe (process 31060) exited with code 0
To automatically close the console when debugging stops, enable Tools->Options->
To automatically close the console when debugging stops.
Press any key to close this window . . .

```

Висновок

Було досліджено алгоритми пошуку та сортування, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій. В результаті виконання лабораторної роботи було ініційовано

двовимірний масив, для кожного стовпця стовпця розраховано середнє арифметичне, отримані значення збережено у другому масиві та відсортовано методом обміну, розділивши задачу на 3 кроки: визначення основних дій, деталізація генерації другого масиву за допомогою підпрограми, сортування елементів другого масиву. В процесі випробування було розглянуто матрицю

«39 -49 -77 -18 6 93 -82

14 -82 85 -75 -10 -42 41

39 89 47 -31 -33 -80 -74

-35 95 -19 17 -49 -77 -85

38 36 -65 36 -72 84 -85»

і отримано масив «-35 -41 -41.3333 -53.6667 -65.5 -66.3333 -81.5».