# 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 № з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«ДОСЛІДЖЕННЯ АЛГОРИТМІВ ОБХОДУ МАСИВІВ

Варіант 7

Виконав студент ІП-12 Васильєв Єгор Костянтинович

(шифр, прізвище, ім'я, по батькові)

Перевірив

( прізвище, ім'я, по батькові)

Київ 2021

**Лабораторна робота №9**

**Дослідження лінійних алгоритмів**

**Мета –** дослідити алгоритми обходу масивів, набути практичних

навичок використання цих алгоритмів під час складання програмних

специфікацій.

**Індивідуальне завдання:** 

**Постановка задачі**

Результатом розв’язку є значення та індекс останнього від’ємного елемента кожного рядка та двовимірний масив у якому замінені елементи головної діагоналі на відповідні від’ємні значення.

**Побудова математичної моделі**

Оскільки ми працюватимемо з головною діагоналлю матриці, то вона має бути квадратна, але для зручності сприйняття я все одно використовую дві змінні з однаковим значенням для кількості рядків і кількості стовбців матриці. Також я створюю копію даного двовимірного масиву і переставляю елементи вже в ньому, для того, щоб в одному циклі і шукати, і переставляти елементи, початковий слугує лише для пошуку від’ємних елементів. Я використовую функцію rand() і ділю отримані значення на 2 для заповнення матриці псевдовипадковими дійсними числами. Також мені знадобиться змінна типу bool для того, щоб у випадку відсутності від’ємних елементів у рядку алгоритм працював правильно, яка прийматиме істинне значення лише у випадку присутності шуканого елемента, а у циклі буде скидатися до стану false.

*Складемо таблицю змінних*

|  |  |  |  |
| --- | --- | --- | --- |
| Змінна | Тип | Ім’я | Призначення |
| Змінна що вказується користувачем | Цілий | size | Розмір масиву |
| Номер рядка | Цілий | first\_index | Перший індекс шуканого елемента |
| Номер стовпця | Цілий | second\_index | Другий індекс шуканого елемента |
| Розмірність масиву | Цілий | rows | Кількість рядків |
| Розмірність масиву | Цілий | cols | Кількість стовпців |
| Змінна, що приймає значення «true» чи «false» | Логічний | flag | Перевірка успішності пошуку від’ємного елемента |
| Лічильник для обходу по рядкам | Цілий | i | Лічильник |
| Лічильник для обходу по стовпцям | Цілий | j | Лічильник |
| Згенерована матриця | Індексований | A | Початковий двовимірний масив |
| Копія згенерованої матриці | Індексований | B | Двовимірний масив у якому переставляються елементи |
| Значення шуканого елемента | Дійсний | value | Значення від’ємного елемента |

**Розв’язання**

Програмні специфікації запишемо у псевдокоді та графічній формі у вигляді блок-схеми.

*Крок 1.* Деталізація основних дій

*Крок 2.* Деталізація дії формування масивів

*Крок 3.* Деталізація дії пошуку від’ємних елементів

*Крок 4.* Заміна елементів

**Псевдокод**

Крок 1

**початок**

A[rows][cols], B[rows][cols]

tmp

size

first\_index

second\_index

rows=size

cols=size

value

-деталізація дії формування масивів

-деталізація дії пошуку від’ємних елементів

-заміна елементів

**кінець**

Крок 2

**Початок**

A[rows][cols], B[rows][cols]

tmp

size

first\_index

second\_index

rows=size

cols=size

value

**повторити** (i=0, rows) **раз**

**повторити** (j=0, cols) **раз**

A[i][j]=(rand()%100-50)/2

B[i][j]=A[i][j]

-деталізація дії пошуку від’ємних елементів

**кінець повторити**

-заміна елементів

**кінець повторити**

**кінець**

Крок 4

**початок**

A[rows][cols], B[rows][cols]

tmp

size

first\_index

second\_index

rows=size

cols=size

value

**повторити** (i=0, rows) **раз**

flag=false

**повторити** (j=0, cols) **раз**

A[i][j]=(rand()%100-50)/2

B[i][j]=A[i][j]

**якщо** A[i][j] < 0

**то**

first\_index = i

second\_index = j

value = A[i][j]

flag = true

**все якщо**

**все повторити**

-заміна елементів

**все повторити**

**кінець**

Крок 3

**початок**

A[rows][cols], B[rows][cols]

tmp

size

first\_index

second\_index

rows=size

cols=size

value

**повторити** (i=0, rows) **раз**

flag=false

**повторити** (j=0, cols) **раз**

A[i][j]=(rand()%100-50)/2

B[i][j]=A[i][j]

**якщо** A[i][j] < 0

**то**

first\_index = i

second\_index = j

value = A[i][j]

flag = true

**все якщо**

**все повторити**

**якщо** flag==true

**то**

tmp = B[i][i]

B[i][i] = value

B[first\_index][second\_index] = tmp

**вивести** first\_index, second\_index, value

**інакше**

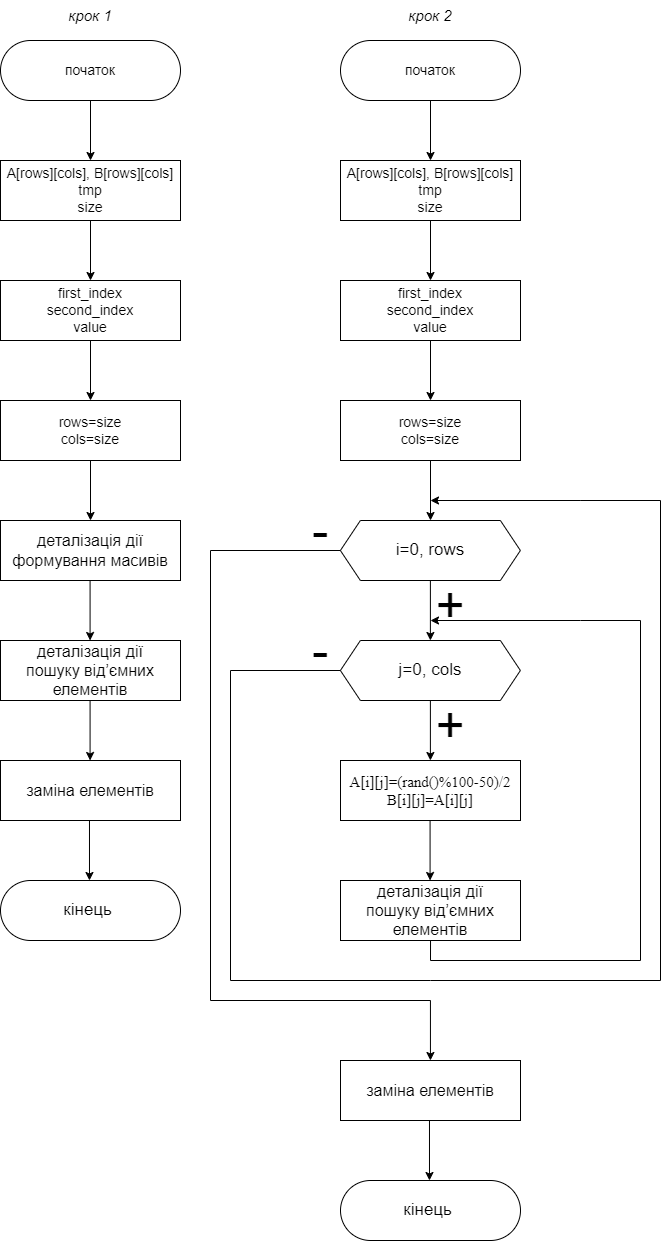
**вивести** «No negative elements in row»

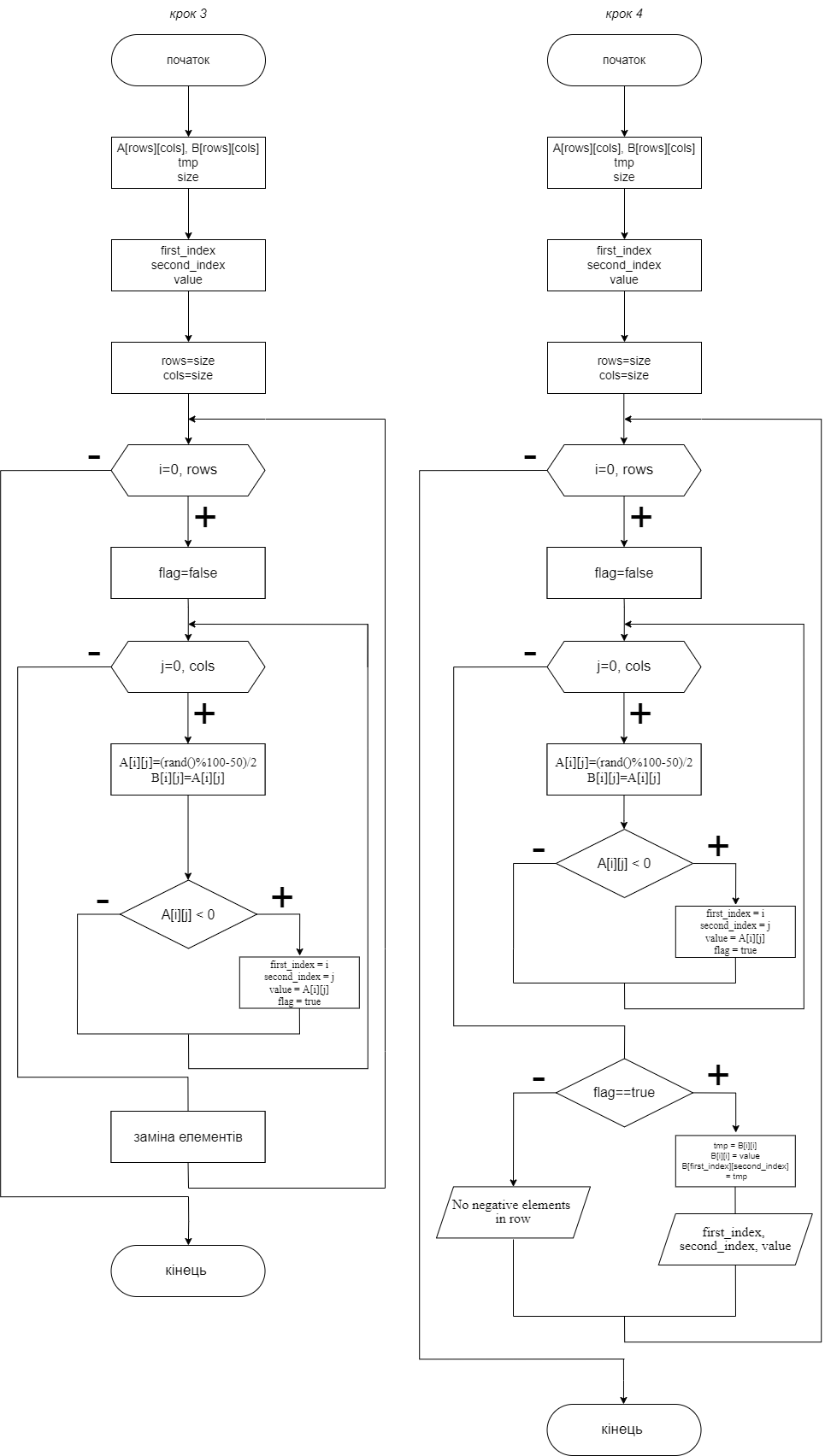
**все якщо**

**все повторити**

**кінець**

**Блок-схема**



****

**Перевірка**

*крок 1*

**початок**

**-**деталізація основних дій

*-*деталізація дії формування масивів

-деталізація дії пошуку від’ємних елементів

-заміна елементів

**кінець**

*крок 2*

**початок**

**-**введення size=2

i=0

flag=false

j=0

A[0][0]=22,5

B[0][0]=22,5

A[0][0]>0

j=1

A[0][1]=22

B[0][1]=22

A[0][1]>0

j=cols

flag!=true

виведення «No negative elements in row»

i=1

flag=false

j=0

A[1][0]=-23

B[1][0]=-23

A[1][0]<0

first\_index=1

second\_index=0

value=-23

flag=true

j=1

A[1][1]=4,5

B[1][1]=4,5

A[1][1]>0

j=cols

flag==true

tmp=B[1][1]

B[1][1]=-23

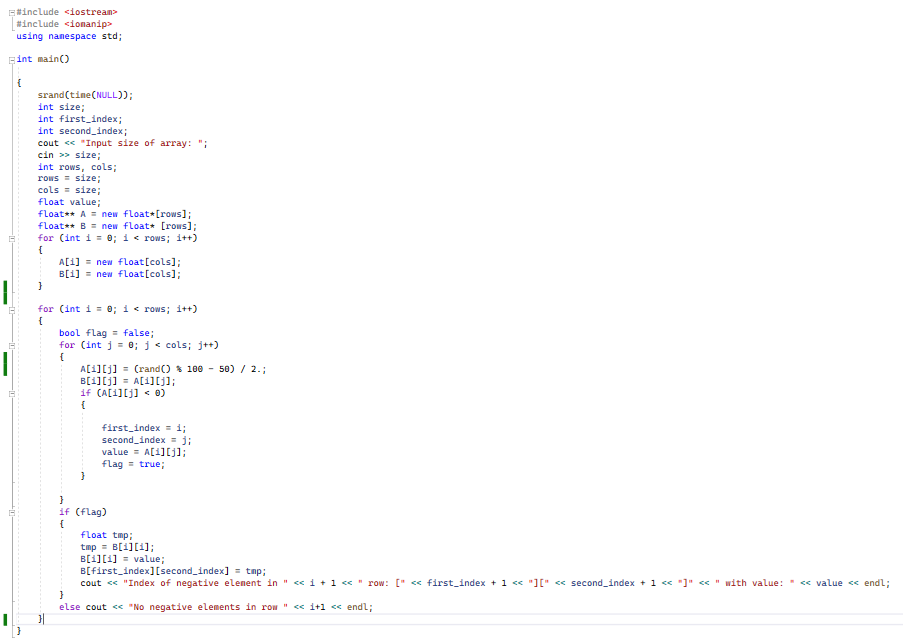
B[1][0]=tmp

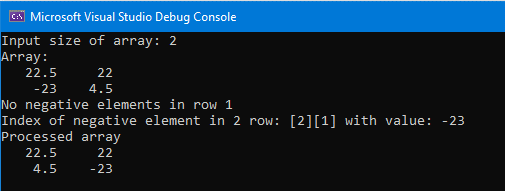
виведення 1,0,-23

i=rows

**кінець**

**Код програми**





**Висновок**

Отже, було досліджено алгоритми обходу масивів, набуто практичних

навичок використання цих алгоритмів під час складання програмних

специфікацій. Було використано вкладені цикли для створення двовимірного масиву та пошуку від’ємних елементів. Було використано змінну логічного типу для фіксації наявності від’ємного елемента у рядку та копію заданого масиву для перестановки елементів у ньому.