

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 No 9 з дисципліни
«Алгоритми та структури даних-1.
Основи алгоритмізації»

«Дослідження алгоритмів обходу масивів»

Варіант 7

Виконав студент ІП-1407 Грицина Діана Русланівна (шифр, прізвище,
ім'я, по батькові)

Перевірів Мартінова Оксана Петрівна (прізвище, ім'я, по батькові)

Київ 2021

Лабораторна робота 9

Дослідження алгоритмів обходу масивів

Мета – дослідити алгоритми обходу масивів, набути практичних навичок використання цих алгоритмів під час складання програмних специфікацій.

Завдання:

7	Задано матрицю дійсних чисел $A[m,n]$. В кожному рядку матриці знайти останній від'ємний елемент і його місцезнаходження. Обміняти знайдене значення X з елементом головної діагоналі.
---	---

Постановка задачі: З використанням арифметичних циклів послідовно заповнюємо матрицю значеннями. Оскільки потрібно обміняти останній від'ємний елемент в кожному рядку з елементом головної діагоналі, доречно встановити умову, що матриця квадратна, тобто кількість рядків є такою ж як кількість стовпців. У кожному рядку матриці перевіряємо послідовно чи елемент є від'ємним і в такому разі записуємо його індекс. Після цього обміняємо значення знайденого елемента з елементом головної діагоналі.

Побудова математичної моделі

Змінна	Тип	Призначення
m	int	Рядки матриці
n	int	Стовпці матриці
a	double	Матриця
Функція make_matrix()	void	Заповнення матриці значеннями
Функція matrix_out()	void	Виведення матриці
Функція change_el()	void	Обмін елемента за умовою задачі

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

Крок 1. Визначимо основні дії

Крок 2. Ініціювання та виведення матриці

Крок 3. Обмін елемента за умовою задачі

Псевдокод

Крок 1

початок

Ініціювання та виведення матриці

Обмін елемента за умовою задачі

кінець

Крок 2

початок

повторити

введення m

введення n

поки m!=n

все повторити

a [m][n]

make_matrix (a, m, n)

matrix_out(a, m, n)

функція make_matrix (a, m, n)

повторити

для i від i1 до i2

повторити

для j від j1 до j2

a[i][j] = rand()

все повторити

все повторити

кінець make_matrix

функція matrix_out (a, m, n)

повторити

для i від i1 до i2

повторити

для j від j1 до j2

виведення a[i][j]

все повторити

все повторити

кінець matrix_out

Обмін елемента за умовою задачі
кінець

Крок 3

початок

повторити

введення m

введення n

поки m!=n

все повторити

a [m][n]

make_matrix (a, m, n)

matrix_out(a, m, n)

change_el (a, m, n)

matrix_out(a, m, n)

функція make_matrix (a, m, n)

повторити

для i від i1 до i2

повторити

для j від j1 до j2

matrix[i][j] = rand()

все повторити

все повторити

кінець make_matrix

функція matrix_out (a, m, n)

повторити

для i від i1 до i2

повторити

для j від j1 до j2

виведення matrix[i][j]

все повторити

все повторити

кінець matrix_out

функція change_el (a, m, n)

повторити

для i від i1 до i2

index = -1

повторити

для j від j1 до j2

якщо a[i][j]<0

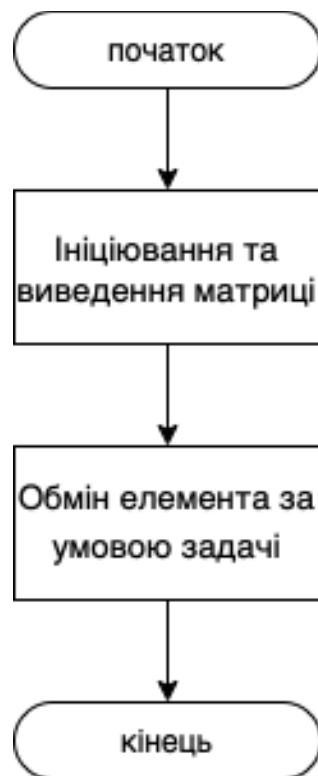
то

index = j

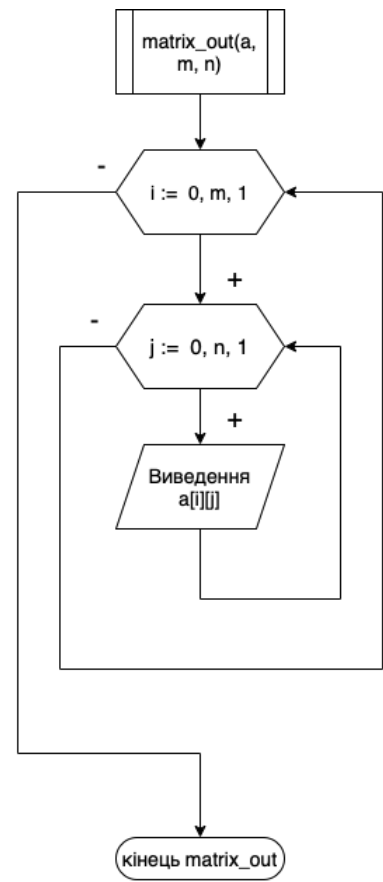
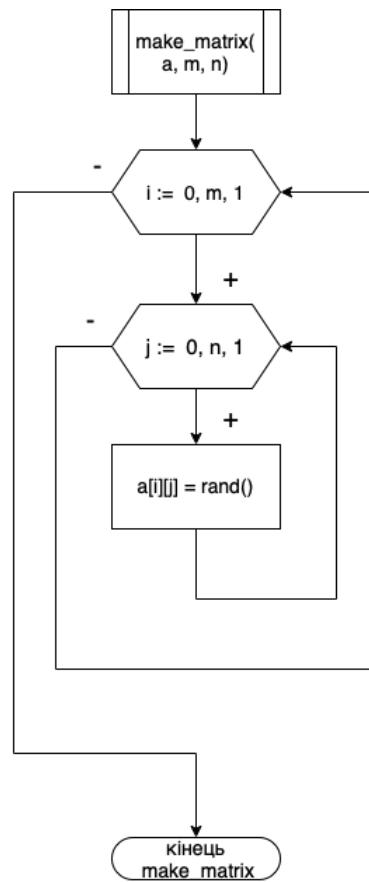
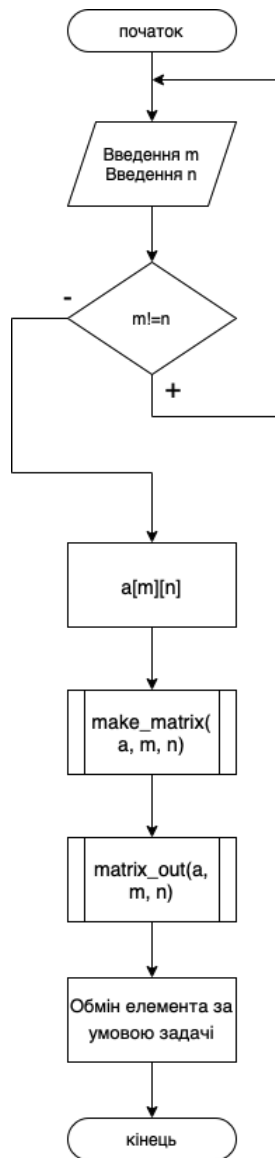
```
        все якщо  
        все повторити  
        якщо  $\text{index} \geq 0$   
        то  
             $\text{tmp} = a[i][i]$   
             $a[i][i] = a[i][\text{index}]$   
             $a[i][\text{index}] = \text{tmp}$   
        все якщо  
        все повторити  
        кінець  $\text{change\_el}$   
        кінець
```

Блок-схема

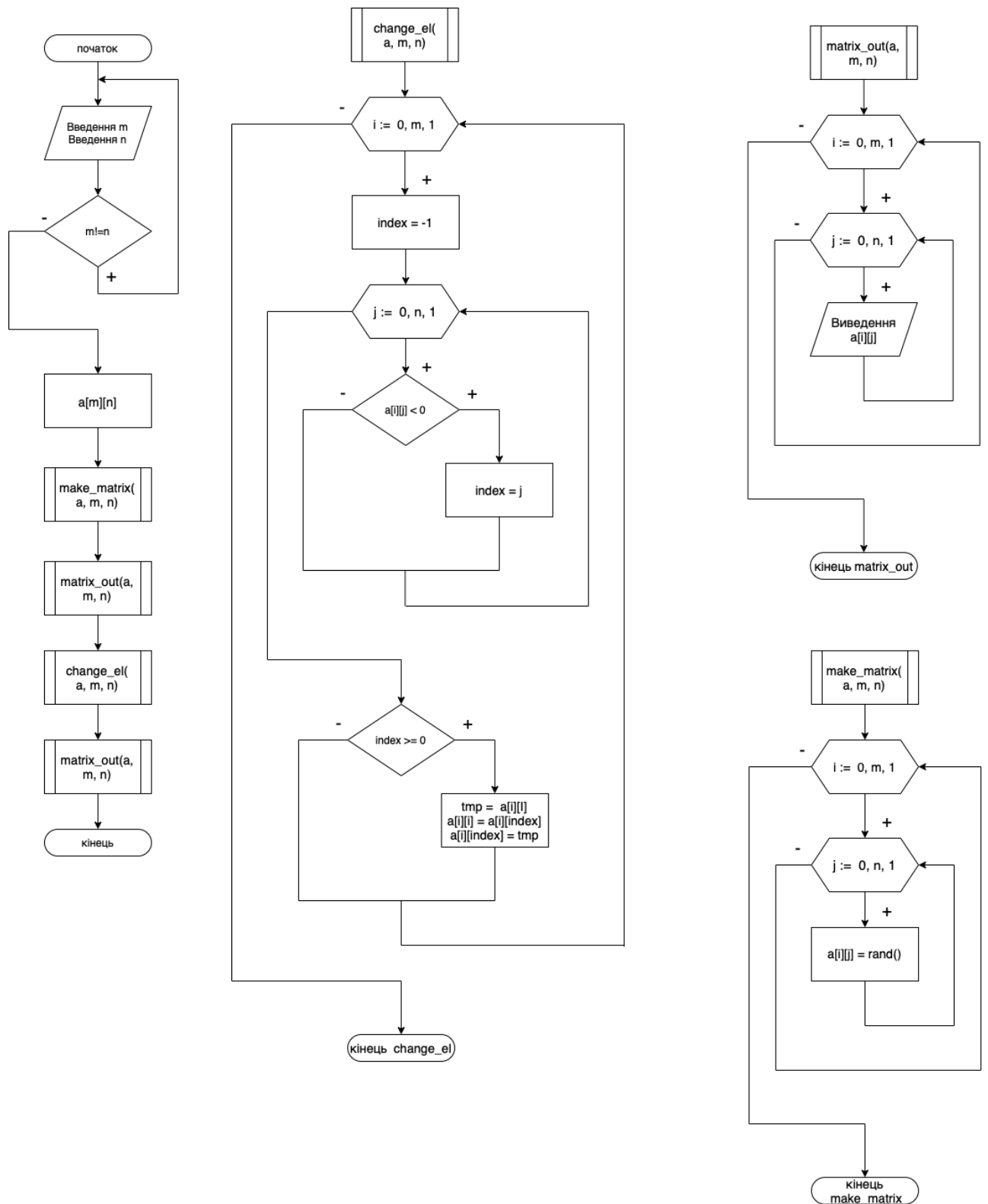
Крок 1



Крок 2



Крок 3



Код програми на C++

```
#include <iostream>
#include <cstdlib>
#include <ctime>
using namespace std;
void make_matrix(double **a, int m, int n);
void matrix_out(double **a, int m, int n);
void change_el(double **a, int m, int n);
int main() {
    srand((unsigned int)time(0));
    int m, n;
    do{
        cout<<"Enter m ";
        cin>>m;
        cout<<"Enter n ";
        cin>>n;
    }while(m!=n);
    double **a;
    a = new double *[m];
    for(int i=0; i<m; i++){
        a[i] = new double [n];
    }
    make_matrix(a, m, n);
    matrix_out(a, m, n);
    cout<<"Changed matrix "<<"\n";
    change_el(a, m, n);
    matrix_out(a, m, n);
    for(int i=0; i<m; i++){
        delete []a[i];
    }
    delete []a;
    return 0;
}

void make_matrix(double **a, int m, int n){
    for(int i=0; i<m; i++){
        for(int j=0; j<n; j++){
            a[i][j] = rand()%41-20;
        }
    }
}
```

```
void matrix_out(double **a, int m, int n){
    for(int i=0; i<m; i++){
        for(int j=0; j<n; j++){
            cout<<a[i][j]<<"\t";
        }
        cout<<"\n";
    }
}

void change_el(double **a, int m, int n){
    for(int i=0; i<m; i++){
        int index = -1;
        for(int j=0; j<n; j++){
            if(a[i][j]<0){
                index = j;
            }
        }
        if(index>=0){
            double tmp = a[i][i];
            a[i][i] = a[i][index];
            a[i][index] = tmp;
        }
    }
}
```


Результат

```
Enter m 4
Enter n 4
-4  -1  -10 -16
18  16  17  20
6   -11 -11 -17
-5  -7  16  14
Changed matrix
-16 -1  -10 -4
18  16  17  20
6   -11 -17 -11
-5  14  16  -7
```

Перевірка:

Матриця:

1 рядок: Останній від'ємний елемент -16, елемент головної діагоналі -4.

Отже змінюємо значення цих елементів.

2 рядок: У рядку немає від'ємних елементів, отже ніяких операцій не виконуємо.

3 рядок: Останній від'ємний елемент -17, елемент головної діагоналі -11.

Отже змінюємо значення цих елементів.

4 рядок: Останній від'ємний елемент -7, елемент головної діагоналі 14.

Отже змінюємо значення цих елементів.

Висновок: У роботі досліджено алгоритми обходу масивів, використано арифметичні цикли та цикл з післяумовою. В кожному рядку матриці визначаємо чи від'ємний елемент та послідовно визначаємо останній, потім обмінюємо його з елементом головної діагоналі. Повторюємо такі дії для кожного рядка матриці з використанням арифметичного циклу.