# 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 № 9 з дисципліни

«Алгоритми та структури даних-1.

Основи алгоритмізації»

«Дослідження лінійних алгоритмів»

Варіант 25

Виконав студент ІП-11 Прищепа Владислав Станіславович

(шифр, прізвище, ім'я, по батькові)

Перевірив Мартинова 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.

Варіант 25

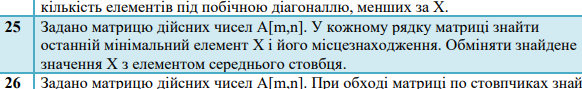
**Індивідуальне завдання:**

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. Обчислення змінної, що описана в п.1, згідно з варіантом (табл. 1).



**Постановка задачі:**

Ввести m, n. Отримати згенеровану з дійсних чисел матрицю m\*n. Вивести на екран найменші елементи кожного рядка матриці та його індекси. Замінити знайдені елементи матриці на елементи з середнього стовпчика, вивести отриману матрицю.

**Математична модель:**

|  |  |  |  |
| --- | --- | --- | --- |
| Зміна | Тип | Ім’я | Призначення |
| К-сть рядків | int | m | Вхідне значення |
| К-сть стовпців | int | n | Вхідне значення |
| Матриця | float\*\* | A | Проміжне значення, результат |
| Масив добутків | float\* | R | Проміжне значення, результат |
| Значення к-сті рядків | int | q | Змінна підпрограми |
| Значення к-сті стовпців | int | z | Змінна підпрограми |
| Покажчик на матрицю | float\*\* | M | Змінна підпрограми |
| Покажчик на масив | float\* | Q | Змінна підпрограми |
| Ітерація 1 | int | i | Змінна підпрограми |
| Ітерація 2 | int | j | Змінна підпрограми |
| Значення мінімального елемента рядка | float | min | Змінна підпрограми |
| Значення середнього елемента рядка | float | medium | Змінна підпрограми |
| Номер стовпця найменшого елемента рядка | int | y | Змінна підпрограми |
| Ітерація 3 | int | h | Змінна підпрограми |

Використаймо у основній програмі підпрограми для генерування динамічної матриці m\*n, її заповнення дійсними числами, виведення матриці, обходу матриці по рядках із лінійним пошуком найменшого елемента кожного рядка, заповненням масиву даними про такі елементи та заміною цих елементів на елементи середнього стовпця матриці, виведення масиву та видалення динамічної матриці. Всі підпрограми будуть використовувати арифметичні цикли або складні, що являють собою арифметичні цикли із вкладеними арифметичними циклами, оператори умовного типу.

**Розв’язання**

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

Крок 1: Визначимо основні дії

Крок 2: деталізуємо ініціалізацію динамічної матриці А

Крок 3: деталізуємо заповнення матриці А

Крок 4: деталізуємо виведення матриці А

Крок 5: деталізуємо заповнення масиву R і заміни елементів матриці А

Крок 6: деталізуємо виведення масиву R

Крок 7: деталізуємо видалення динамічної матриці А

**Псевдокод**

**Основна програма**

**Початок**

**Виведення** "Enter m: "

**Введення** m

**Виведення** "Enter n: "

**Введення** n

A := array\_2d(m, n)

generate\_2d\_array(A, m, n)

**Виведення** "Matrix initial:"

output\_2d\_array(m, n, A)

min(A, m, n, R)

output\_array(R)

**Виведення** "Matrix final:"

output\_2d\_array(m, n, A)

delition\_of\_2d\_array(A, m)

**Кінець**

**Підпрограми**

**array\_2d(q, z)**

int\*\* M := new int\* [q]

**для** і **від** 0 **до** q, **з кроком** 1 **повторити**

M[i] := new int[z]

**все повторити**

return M

**Кінець**

**generate\_2d\_array(M, q, z)**

srand(time(NULL))

**для** і **від** 0 **до** q, **з кроком** 1 **повторити**

**для** j **від** 0 **до** z, **з кроком** 1 **повторити**

M[i][j] := ((rand() % 5000) / 100.0)-15

**все повторити**

**все повторити**

**Кінець**

**output\_2d\_array(q, z, M)**

**для** і **від** 0 **до** q, **з кроком** 1 **повторити**

**для** j **від** 0 **до** z, **з кроком** 1 **повторити**

**Виведення** M[i][j], “\t”

**все повторити**

**все повторити**

**Виведення** endl

**Кінець**

**min(M, q, z, Q)**

h:=0

y:=0

**для** і **від** 0 **до** q, **з кроком** 1 **повторити**

min:=35

medium:= M[i][z / 2]

**для** j **від** 0 **до** z, **з кроком** 1 **повторити**

**Якщо** M[i][j] < min

Q[j] := M[i][j] \* Q[j]

**все якщо**

**все повторити**

Q[h] := min

Q[h + 1] := i + 1

Q[h + 2] := y + 1

h := 3 + h

M[i][y] := medium

**все повторити**

Q[h] := -100

**Кінець**

**output\_array(Q)**

h := 0

**Виведення** "Minimal elements:"

**повторити поки** Q[h] != -100

**Виведення** Q[h] , '\t' , Q[h + 1] , '\t' , Q[h + 2] , endl

h := 3 + h

**все повторити**

**Кінець**

**delition\_of\_2d\_array(M, q)**

**для** і **від** 0 **до** q, **з кроком** 1 **повторити**

delete[] M[i]

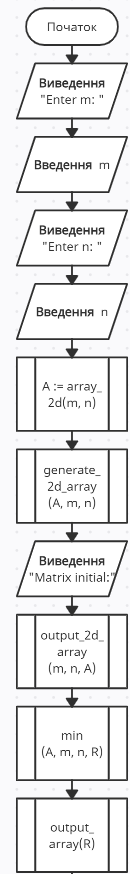
**все повторити**

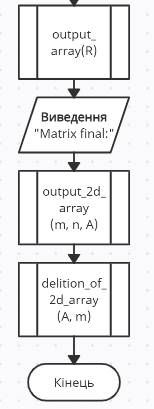
delete[] M

**Кінець**

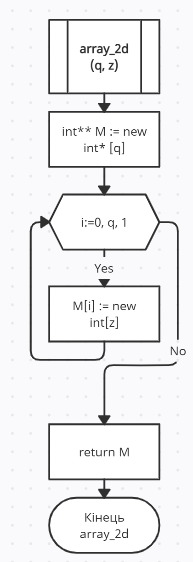
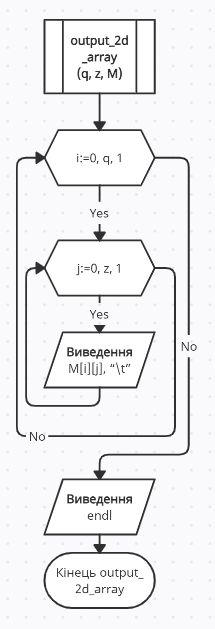
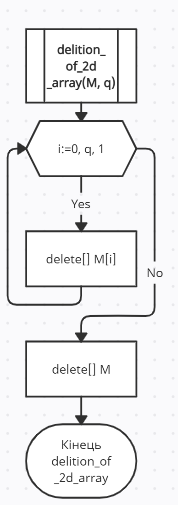
**Блок-схема алгоритму**

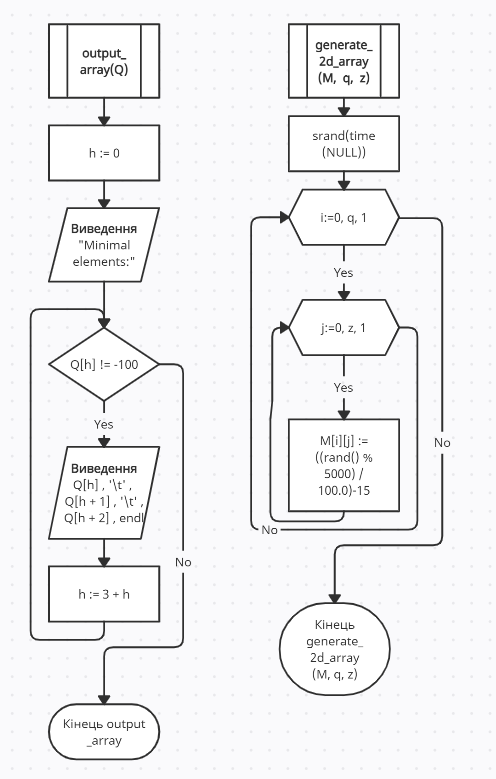
**Основна програма**

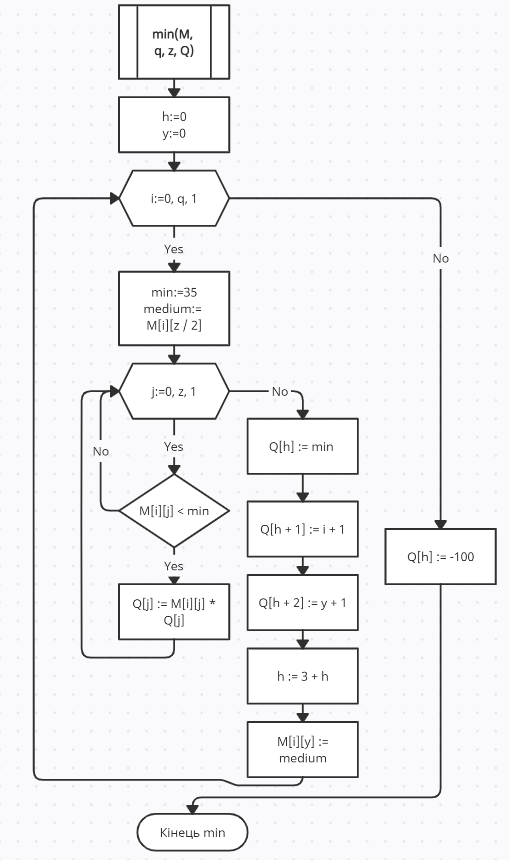
****

****

**Підпрограми**

**** **** 





**Код та виконання коду на С++:**

#include <iostream>

#include <time.h>

using namespace std;

float\*\* array\_2d(int, int);

void generate\_2d\_array(float\*\*, int, int);

void output\_2d\_array(int, int, float\*\*);

void min(float\*\*, int, int, float\*);

void output\_array(float\*);

void delition\_of\_2d\_array(float\*\*, int);

int main() {

float\*\* A;

int m, n;

float R[100];

cout << "Enter m: ";

cin >> m;

cout << "Enter n: ";

cin >> n;

A = array\_2d(m, n);

generate\_2d\_array(A, m, n);

cout << "Matrix initial:" << endl;

output\_2d\_array(m, n, A);

min(A, m, n, R);

output\_array(R);

cout << "Matrix final:" << endl;

output\_2d\_array(m, n, A);

delition\_of\_2d\_array(A, m);

system("pause");

}

float\*\* array\_2d(int q, int z) {

float\*\* M = new float\* [q];

for (int i = 0; i < q; i++) {

M[i] = new float[z];

}

return M;

}

void generate\_2d\_array(float\*\* M, int q, int z) {

srand(time(NULL));

for (int i = 0; i < q; i++) {

for (int j = 0; j < z; j++) {

M[i][j] = ((rand() % 5000) / 100.0)-15;

}

}

}

void output\_2d\_array(int q, int z, float\*\* M) {

for (int i = 0; i < q; i++) {

for (int j = 0; j < z; j++) {

cout << M[i][j] << "\t";

}

cout << endl;

}

}

void min(float\*\* M, int q, int z, float\* Q) {

float min, medium;

int y=0, h = 0;

for (int i = 0; i < q; i++) {

min = 35;

medium = M[i][z / 2];

for (int j = 0; j < z; j++) {

if (M[i][j] < min) {

min = M[i][j];

y = j;

}

}

Q[h] = min;

Q[h + 1] = i + 1;

Q[h + 2] = y + 1;

h += 3;

M[i][y] = medium;

}

Q[h] = -100;

}

void output\_array(float\* Q) {

int h = 0;

cout << "Minimal elements:" << endl;

while (Q[h] != -100) {

cout << Q[h] << '\t' << Q[h + 1] << '\t' << Q[h + 2] << endl;

h += 3;

}

}

void delition\_of\_2d\_array(float\*\* M, int q) {

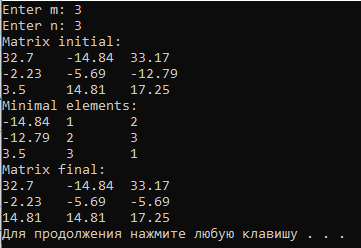
for (int i = 0; i < q; i++) {

delete[] M[i];

}

delete[] M;

}

****

**Випробування алгоритму**

|  |  |
| --- | --- |
| **Крок** | **Дія** |
|  | **Початок** |
| **1** | Введення розміру матриці, генерація і виведення матриці  32.7 -14.84 33.17  -2.23 -5.69 -12.79  3.5 14.81 17.25 |
| **2** | Виведення мінімальних елементів рядків та їх положення:  -14.84 1 2  -12.79 2 3  3.5 3 1 |
| **3** | Виведення кінцевої матриці:  32.7 -14.84 33.17  -2.23 -5.69 -5.69  14.81 14.81 17.25 |
|  | **Кінець** |

**Висновок**

Отже, у результаті лабораторної роботи ми дослідили алгоритми обходу масивів та набути практичних навичок їх використання під час складання програмних специфікацій підпрограм; навчилися будувати математичну модель, що відповідає постановці задачі, псевдокод та блок-схеми, які пояснюють логіку алгоритму. В результаті виконання лабораторної роботи ми отримали алгоритм для знаходження найменших елементів кожного рядка матриці та заміни їх на значення елемента середнього стовпця, декомпозували задачу на 7 кроків: визначили основні дії, деталізували ініціалізацію динамічної матриці А, її заповнення та виведення, заповнення масиву R і заміни шуканих елементів матриці А на елементи середнього стовпця, виведення масиву R та видалення динамічної матриці А.

У алгоритмі використовується підпрограма обходу матриці по рядках із лінійним пошуком найменшого елемента кожного рядка. Оскільки формула перевірені вручну результати розрахунку членів послідовності відповідають дійсності, то алгоритм правильно подає результат.