Міністерство освіти і науки України

Івано-Франківський національний технічний університет нафти і газу

Інститут інформаційних технологій

Кафедра інженерії програмного забезпечення

Звіт

з лабораторної роботи №7

«Алгоритми пошуку»

з дисципліни «Алгоритми та структури даних»

Виконала:

ст.гр. ІП-22-4

Томин Є.В.

Перевірив:

Григорчук Л. І.

Івано-Франківськ

2023

**Тема**. Алгоритми пошуку

**Мета і зміст роботи:** Застосування алгоритму лінійного (прямого) пошуку та

лінійного (прямого) пошуку з барʼєром, створення, налагодження та виконання

програм, що використовують ці алгоритми.

**В результаті виконання роботи необхідно:**

• вміти застосовувати алгоритм лінійного пошуку при розвʼязанні задач;

•вміти застосовувати алгоритм лінійного пошуку з барʼєром при розвʼязанні задач;

• вміти обирати оптимальний тип для представлення вхідних та вихідних даних;

• вміти реалізовувати розроблені алгоритми на мові програмування;

• вміти виконувати тестування програми;

• вміти пояснювати хід виконання алгоритму та принцип роботи програми.

**Варіант 6**

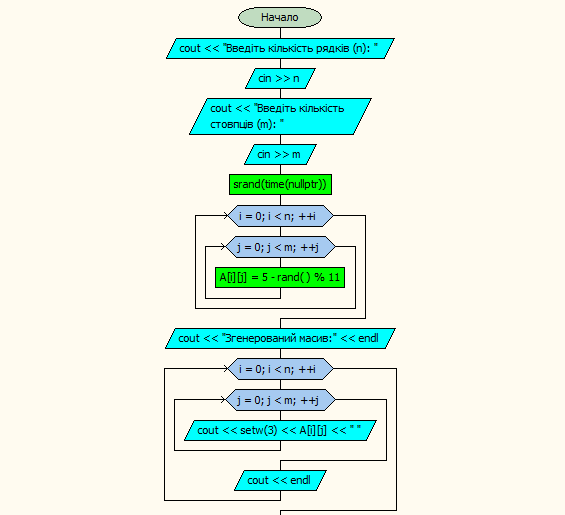
Задано двовимірний масив цілих чисел A[n, п]. При обході масиву по рядках

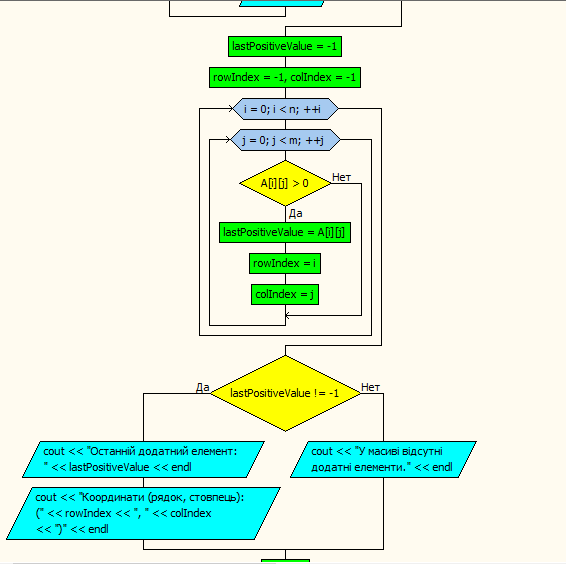
зайти останній додатний елемент та його координати (номер рядка та номер

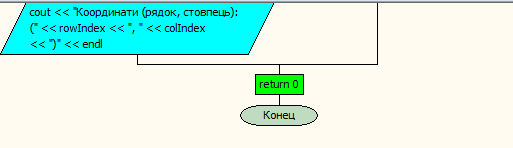
стовпця).

**Виконання**

**Блок-схема**

****

****

****

**Алгоритм виконання:**

1. Початок

2. Вхід: розмірність масиву [n,m], двовимірний масив А

3.Проміжні змінні i,j; rowIndex, colIndex

4. для i від 0 до n виконати

5. для j від 0 до m виконати

6. A[n][m] = 5 - rand() % 11;

7. кінець циклу по і

8. кінець циклу по j

9.для і від 0 до n виконати

10. Якщо A[n][m]=0 то = rowIndex i вихід

11. для і від 0 до n виконати

12. Якщо A[n][m]<0 то , colIndex = I

13.Вихід: rowIndex , colIndex

14. Кінець

**Код програми**

#include <iostream>

#include <iomanip>

#include <cstdlib>

#include <ctime>

using namespace std;

int main() {

// Визначення розмірності масиву

int n, m;

cout << "Введіть кількість рядків (n): ";

cin >> n;

cout << "Введіть кількість стовпців (m): ";

cin >> m;

// Ініціалізація масиву та генерація випадкових чисел

int A[n][m];

srand(time(nullptr));

// Заповнення масиву випадковими числами від -5 до 5

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

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

A[i][j] = 5 - rand() % 11;

}

}

// Виведення масиву

cout << "Згенерований масив:" << endl;

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

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

cout << setw(3) << A[i][j] << " ";

}

cout << endl;

}

// Знайдемо останній додатній елемент та його координати

int lastPositiveValue = -1;

int rowIndex = -1, colIndex = -1;

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

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

if (A[i][j] > 0) {

lastPositiveValue = A[i][j];

rowIndex = i;

colIndex = j;

}

}

}

// Виведення результатів

if (lastPositiveValue != -1) {

cout << "Останній додатний елемент: " << lastPositiveValue << endl;

cout << "Координати (рядок, стовпець): (" << rowIndex << ", " << colIndex << ")" << endl;

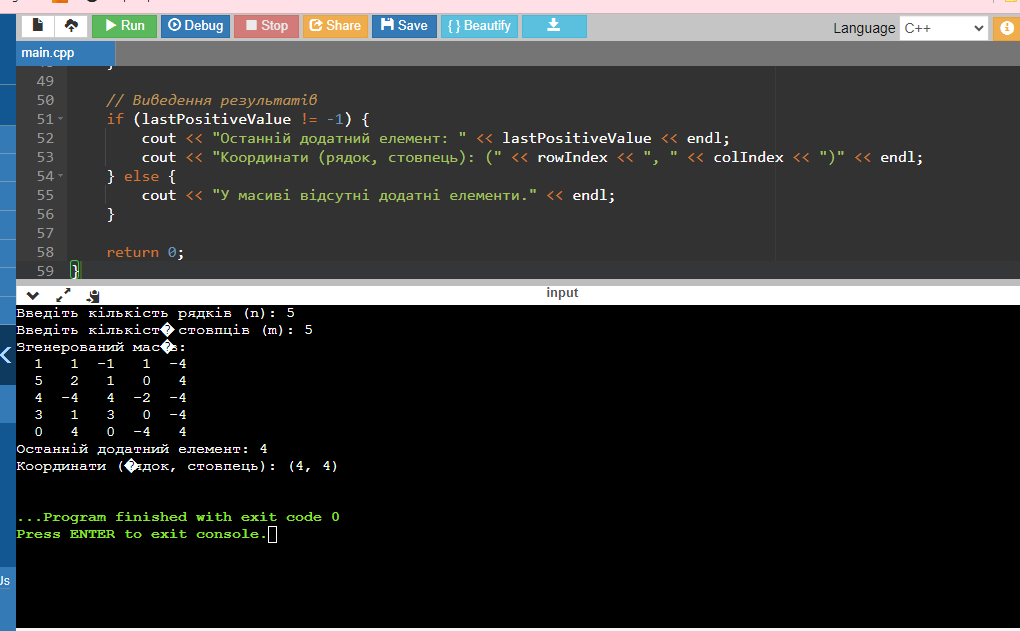
} else {

cout << "У масиві відсутні додатні елементи." << endl;

}

return 0;

}



**Мал.1.** На малюнку зображено матрицю, де вході виконання програма шукає останній додатній елемент та виводить його координати

**Код, де елементи матриці вводяться з масиву**

#include <iostream>

#include <iomanip>

using namespace std;

int main() {

// Визначення розмірності масиву

int n, m;

cout << "Введіть кількість рядків (n): ";

cin >> n;

cout << "Введіть кількість стовпців (m): ";

cin >> m;

// Ініціалізація масиву та введення елементів з клавіатури

int A[n][m];

cout << "Введіть елементи масиву:" << endl;

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

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

cout << "A[" << i << "][" << j << "]: ";

cin >> A[i][j];

}

}

// Виведення масиву

cout << "Ваш масив:" << endl;

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

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

cout << setw(3) << A[i][j] << " ";

}

cout << endl;

}

// Знайдемо останній додатній елемент та його координати

int lastPositiveValue = -1;

int rowIndex = -1, colIndex = -1;

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

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

if (A[i][j] > 0) {

lastPositiveValue = A[i][j];

rowIndex = i;

colIndex = j;

}

}

}

// Виведення результатів

if (lastPositiveValue != -1) {

cout << "Останній додатний елемент: " << lastPositiveValue << endl;

cout << "Координати (рядок, стовпець): (" << rowIndex << ", " << colIndex << ")" << endl;

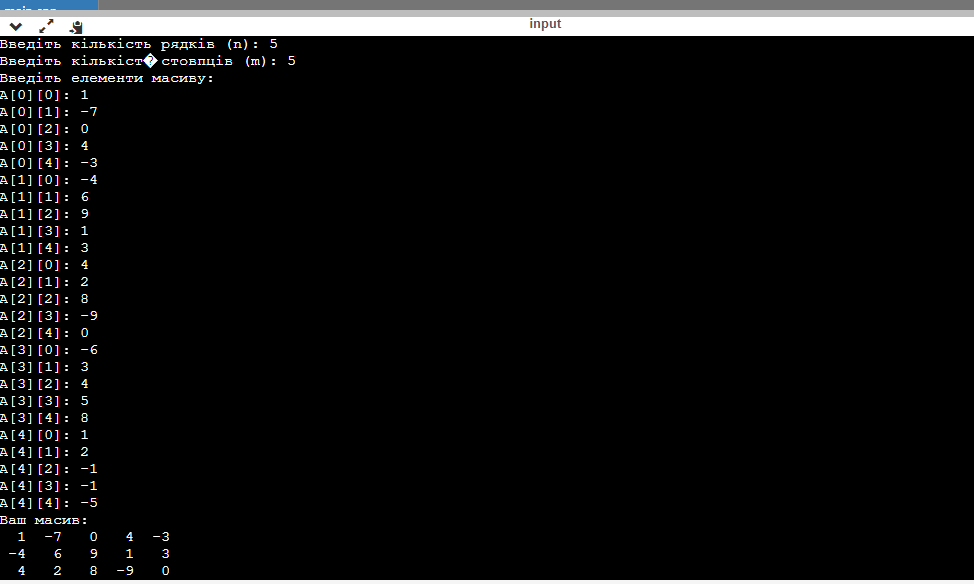
} else {

cout << "У масиві відсутні додатні елементи." << endl;

}

return 0;

}



**Мал.2.** На малюнку зображено матрицю, де вході виконання програма шукає останній додатній елемент та виводинь його координати

**Висновок**

Виконуючи лабораторну роботу, я навчилася застосовувати алгоритми лінійного (прямого) пошуку та лінійного (прямого) пошуку з барʼєром, створювати, налагоджувати та виконувати програму, що використовує ці алгоритми.