МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ

Національний технічний університет України

"Київський політехнічний інститут імені Ігоря Сікорського"

Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

Програмування. Частина 1. Основи програмування

Лабораторна робота №7

Тема: Реалізація алгоритмів алгоритмів з використанням

багатовимірних масивів. Обробка двовимірних масивів

Варіант №11

Виконав: Коваль Б.А.

студент групи ІС-31

Перевірив:

Драган М.С.

Київ-2023

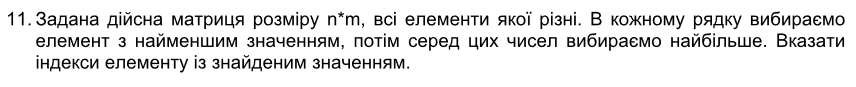
МЕТА :

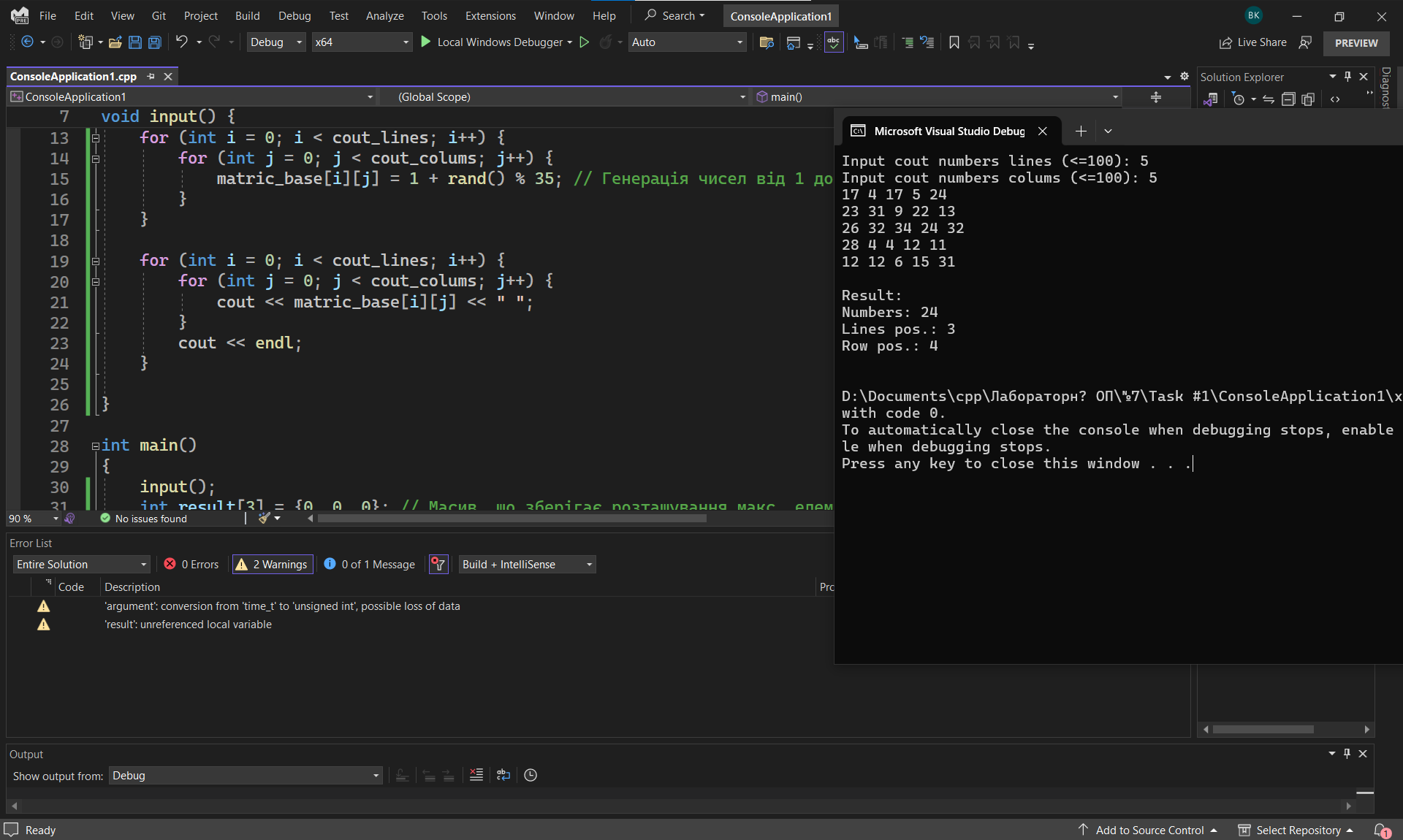
1. Ознайомитися з особливостями типу масиву;

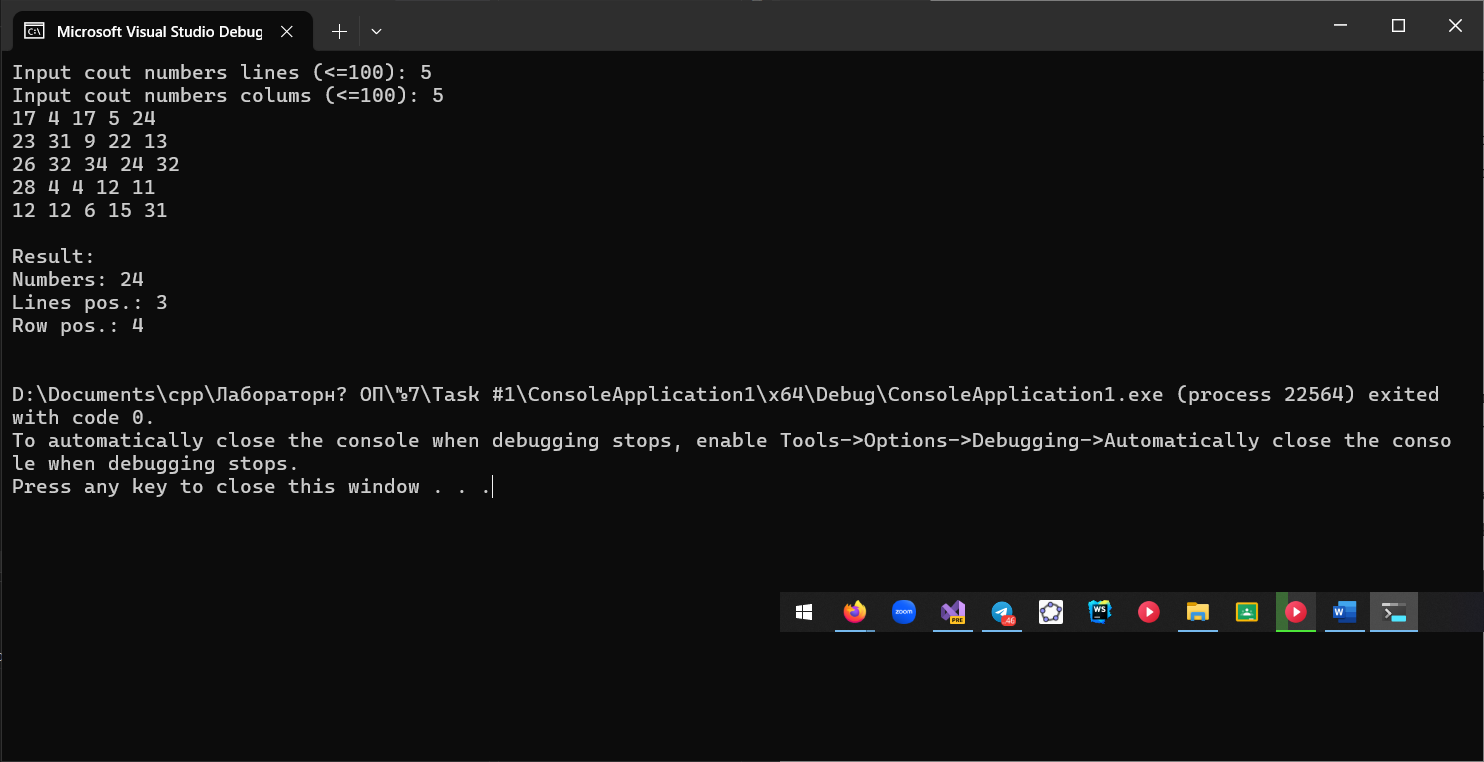
2. Опанувати технологію застосування масивів даних;

3. Навчитися розробляти алгоритми та програми із застосуванням двовимірних масивів

ЗАВДАННЯ №1







#include <iostream>

using namespace std;

int cout\_lines, cout\_colums;

int matric\_base[100][100];

void input() {

srand(time(NULL));

cout << "Input cout numbers lines (<=100): ";

cin >> cout\_lines;

cout << "Input cout numbers colums (<=100): ";

cin >> cout\_colums;

for (int i = 0; i < cout\_lines; i++) {

for (int j = 0; j < cout\_colums; j++) {

matric\_base[i][j] = 1 + rand() % 35; // Генерація чисел від 1 до 35

}

}

for (int i = 0; i < cout\_lines; i++) {

for (int j = 0; j < cout\_colums; j++) {

cout << matric\_base[i][j] << " ";

}

cout << endl;

}

}

int main()

{

input();

int result[3] = {0, 0, 0}; // Масив, що зберігає розташування макс. елем. серед мінім. елем. в рядках

for (int i = 0; i < cout\_lines; i++) {

int min\_elem\_lines[3] = { matric\_base[i][0], 0, 0 }; // Масив, що зберігає мінім елем. та його позицію

for (int j = 0; j < cout\_colums; j++) {

if (min\_elem\_lines[0] > matric\_base[i][j]) {

min\_elem\_lines[0] = matric\_base[i][j];

min\_elem\_lines[1] = i;

min\_elem\_lines[2] = j;

}

}

if (min\_elem\_lines[0] > result[0]) {

result[0] = min\_elem\_lines[0];

result[1] = min\_elem\_lines[1];

result[2] = min\_elem\_lines[2];

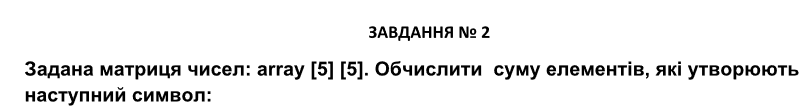
}

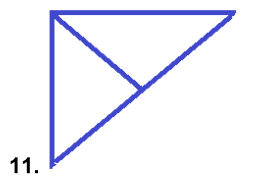
}

cout << endl << "Result: " << endl << "Numbers: " << result[0] << endl << "Lines pos.: " << result[1] + 1 << endl << "Row pos.: " << result[2] + 1 << endl << endl;

}

ЗАВДАННЯ №2





#include <iostream>

using namespace std;

int cout\_lines, cout\_colums;

int matric\_base[101][101];

void input() {

srand(time(NULL));

cout << "Input cout numbers lines (<=101 and only odd ones): ";

cin >> cout\_lines;

cout << "Input cout numbers colums (<=101 and only odd ones): ";

cin >> cout\_colums;

for (int i = 0; i < cout\_lines; i++) {

for (int j = 0; j < cout\_colums; j++) {

matric\_base[i][j] = 1 + rand() % 9; // Генерація чисел від 1 до 9

}

}

}

int main()

{

int res = 0;

input();

for (int i = 0; i < cout\_lines; i++) {

for (int j = 0; j < cout\_colums; j++) {

if (i == 0 || j == 0 || i == j && (i + j - cout\_lines < 1) || (i + j == cout\_lines - 1)) {

res += matric\_base[i][j];

cout << matric\_base[i][j] << " ";

}

else

cout << 0 << " ";

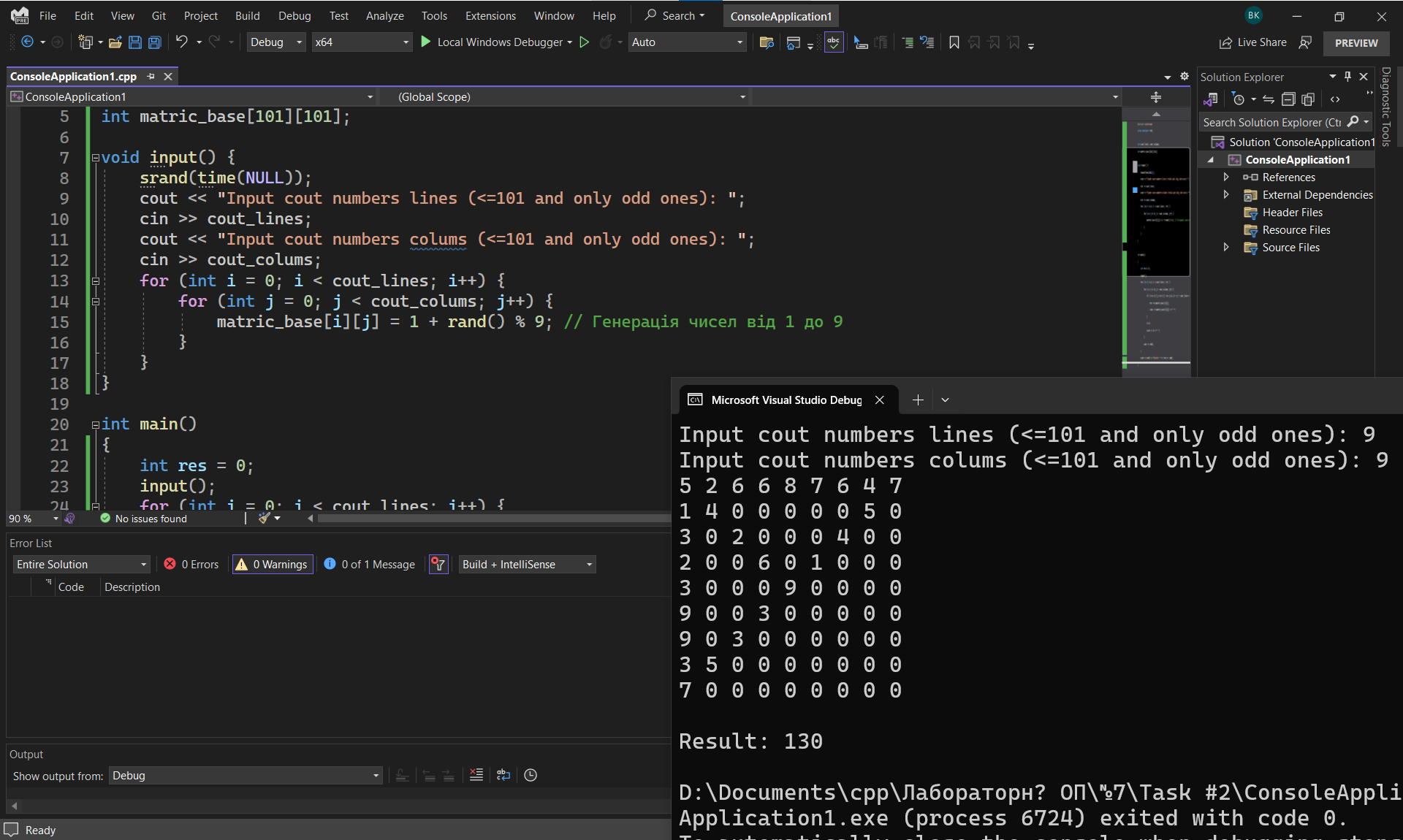
}

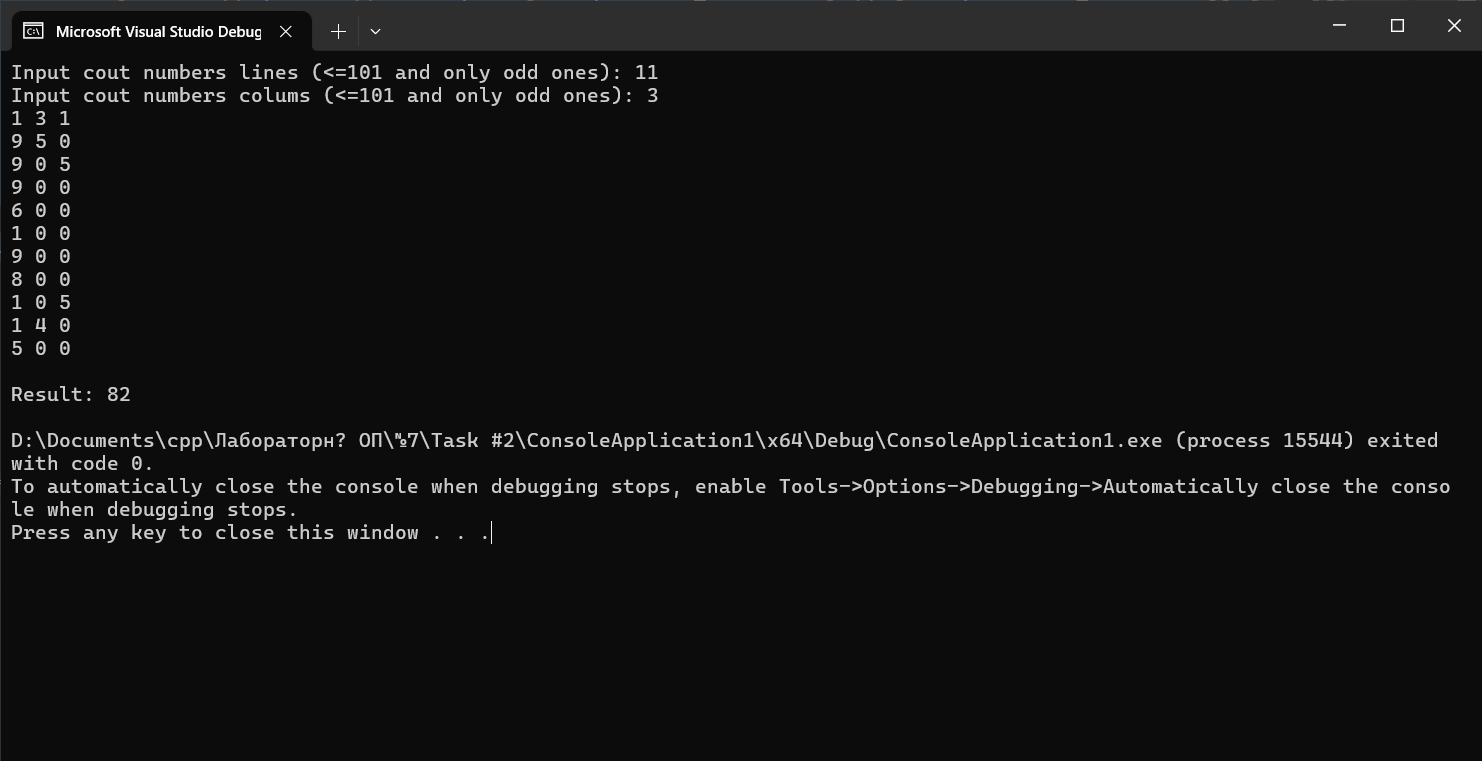
cout << endl;

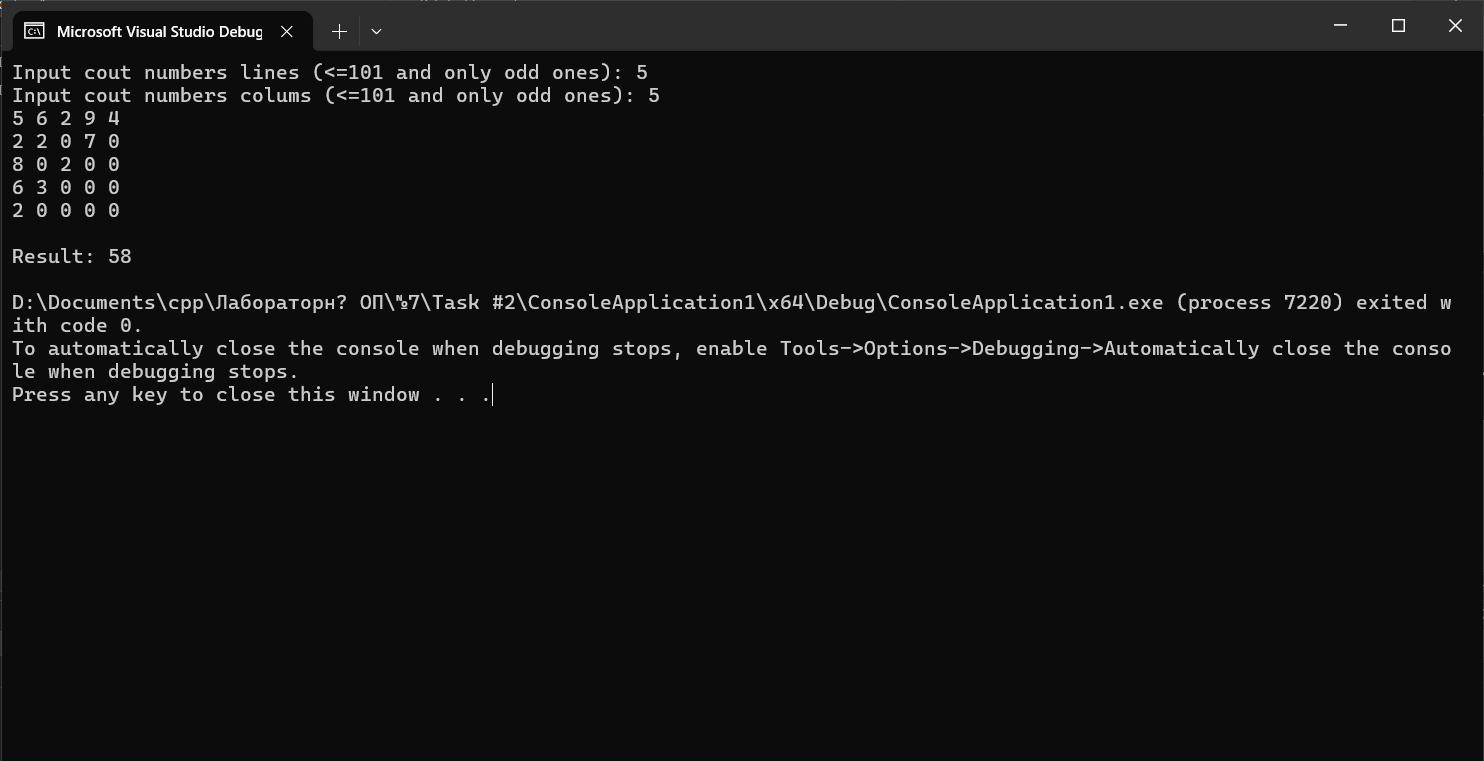
}

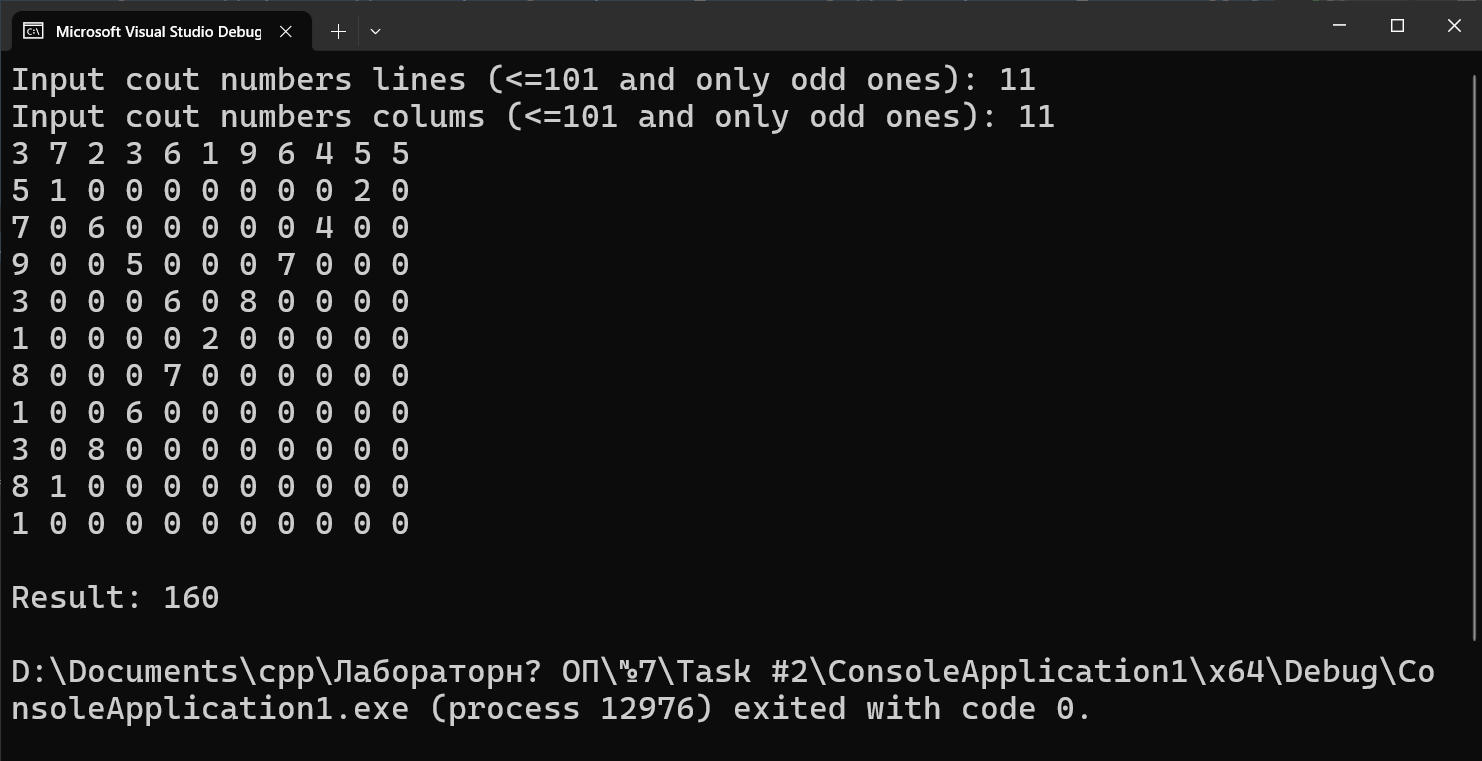
cout << endl << "Result: " << res << endl;

}

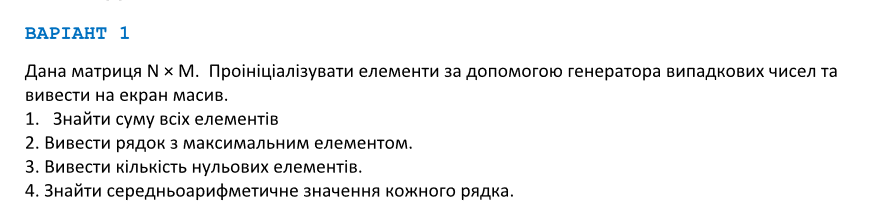








ЗАВДАННЯ №3



#include <iostream>

using namespace std;

int cout\_row, cout\_colums;

int matric\_base[100][100];

void input() {

srand(time(NULL));

cout << "Input cout numbers row (<=100): ";

cin >> cout\_row;

cout << "Input cout numbers colums (<=100): ";

cin >> cout\_colums;

for (int i = 0; i < cout\_row; i++) {

for (int j = 0; j < cout\_colums; j++) {

matric\_base[i][j] = 0 + rand() % 35; // Генерація чисел від 1 до 35

}

}

}

void out\_row(int index\_elem) {

int i = 0;

cout << "Row with maximum element: ";

for (auto& elem : matric\_base[index\_elem]) {

if (i < cout\_colums)

cout << elem << " ";

else

break;

i++;

}

}

int main()

{

input();

int all\_sum = 0, cout\_zero = 0;

int row\_max\_elem[2] = {0, 0};

for (int i = 0; i < cout\_row; i++) {

float row\_sum = 0;

for (int j = 0; j < cout\_colums; j++) {

cout << matric\_base[i][j] << " ";

row\_sum += matric\_base[i][j];

all\_sum += matric\_base[i][j];

if (matric\_base[i][j] > row\_max\_elem[0]) {

row\_max\_elem[0] = matric\_base[i][j];

row\_max\_elem[1] = i;

}

if (matric\_base[i][j] == 0)

cout\_zero++;

}

cout << endl << "Arithmetic mean of the row: " << row\_sum / cout\_colums << endl << endl;

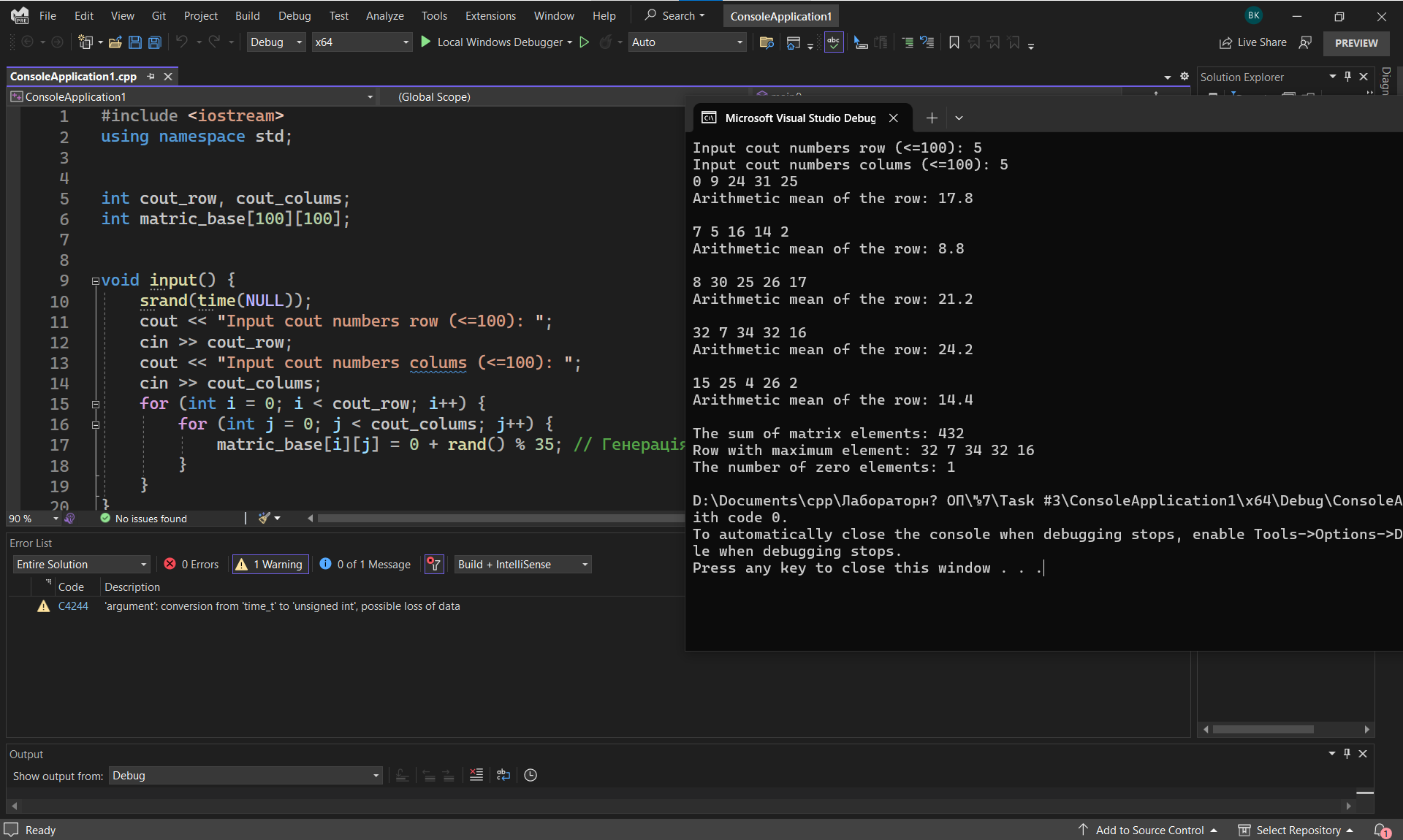
}

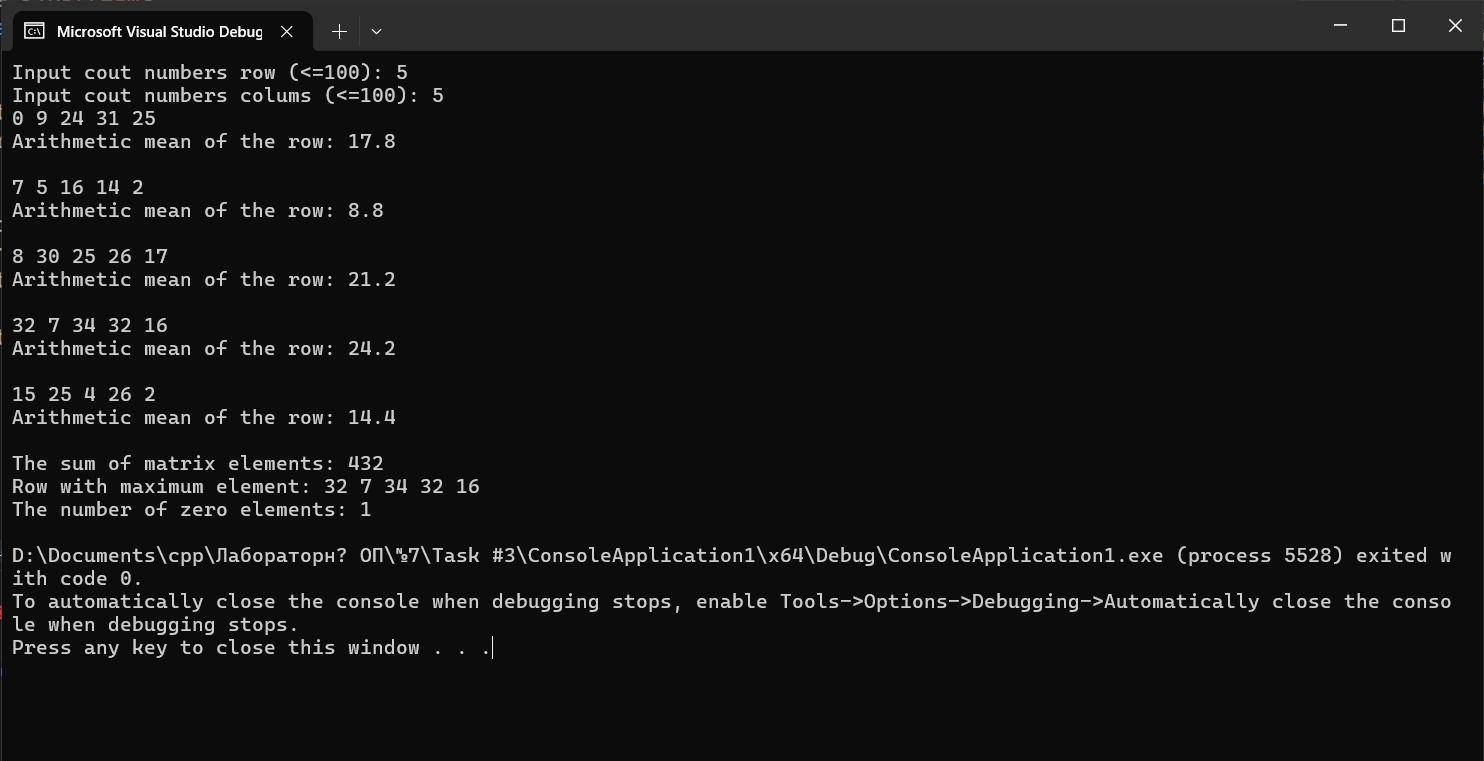
cout << "The sum of matrix elements: " << all\_sum << endl;

out\_row(row\_max\_elem[1]);

cout << endl << "The number of zero elements: " << cout\_zero << endl;

}





КОНТРОЛЬНІ ЗАПИТАННЯ:

1. **Означення матриці в C++:** В мові програмування C++, матриця представляється як двовимірний масив.
2. **Представлення матриці в оперативній пам'яті:** Елементи матриці в C++ зберігаються в оперативній пам'яті послідовно, рядок за рядком. Значення елементів можна звертатися за допомогою індексів рядка та стовпця.
3. **Доступ до елементів масиву чи матриці:** Доступ до елементів матриці здійснюється за допомогою двох індексів (рядка та стовпця).
4. **Доступ до елемента матриці за допомогою одного індексу:** У C++ можна використовувати один індекс для доступу до елементів матриці, якщо мати на увазі одновимірний масив, утворений послідовним розташуванням рядків матриці.
5. **Типові операції над матрицями:** Операції, які можна виконувати над матрицями включають: Додавання матриць, Віднімання матриць, Множення матриць, Транспонування матриці, Знаходження визначника, Розв'язання систем лінійних рівнянь, Інші лінійно-алгебраїчні операції, Зазвичай для реалізації цих операцій використовують цикли та інші конструкції мови C++.

ВИСНОВОК: