МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ



**Дніпропетровський національний університет  
залізничного транспорту імені академіка В. Лазаряна**

Кафедра «Комп’ютерні інформаційні технології»

**Лабораторна робота №6**

**з дисципліни «Основи програмування»**

**на тему: «Динамічні масиви в мові С++»**

Виконав: студент гр. ПЗ1911

Сіньков Г.О.

Прийняла: ас. каф. КІТ

Нежуміра О. І.

Дніпро, 2019

**Лабораторна робота №6**

**Тема.** Динамічні масиви в мові С++.

**Мета.** Набути практичних навичок розробки алгоритмів, написання і налагодження програм для обробки різного типу структур динамічних матриць.

**Постановка задачи**

1.Загальна постановка: розробити два варіанти програми роботи з різними типами структури динамічної матриці.

2.Індивідуальна постановка:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| № | Діапазон значень елементів матриці \* | Рівень складності | | | | |
| E  (достатньо) | D  (задовільно) | C  (добре) | B  (дуже добре) | A  (відміно) |
| 9 | [-22.2; 10.25] | Підрахувати кількість елементів матриці, у яких ціла частина є непарним числом. | Обчислити суму від’ємних елементів матриці, в яких цифра в розряді одиниць не перевищує 2. | Виконати завдання D для кожного рядка матриці. Результати записати в динамічний масив. | Відсортувати рядки матриці за зростанням (непарний варіант) або за спаданням (парний варіант) відповідних елементів масиву, створеного у п.С. | Видалити з матриці рядки, в яких перший елемент менше другого елементу. |

3.Вимоги до програми:

- всі числа з завдання оголошуються як константи;

- розмірність матриці вводиться з клавіатури;

- передбачити обробку некоректного введення користувачем розмірності матриці;

- результати роботи програми виводяться на екран;

- для виділення і звільнення пам’яті для матриць і масиву використовуються оператори *new* і *delete*.

**4.Вимоги до тексту програми:**

- коментарі щодо призначення програми, її вхідних і вихідних даних;

- коментарі щодо призначення кожного блоку програми, дій окремих операторів для пояснення алгоритму;

- самодокументованість коду: всі ідентифікатори повинні мати назви, що відповідають суті змінних.

**Зовнішні специфікації**

1. Вхідні дані

Вхідні дані: arr.rows, arr.cols

**Формат вхідних даних**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Найменування даних | Умовне позначення | Вимоги до даних | Приклад |
| 1 | Кількість рядків матриці | arr.rows | Ціле число | 3 |
| 2 | Кількість стовпців матриці | arr.cols | Ціле число | 3 |

2. Вихідні дані: quantity, sum, Arr

**Формат вихідних даних**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № | Найменування даних | Умовне позначення | Вимоги до даних | Приклад |
| 1 | Кількість елементів матриці , у яких ціла частина є непарним числом | quantity | Ціле число | 4 |
| 2 | Cума від’ємних елементів матриці, в яких цифра в розряді одиниць не перевищує 2 | sum | Ціле число | 2 |
| 3 | Результати обчислення суми від’ємних елементів матриці в рядку, які записані в масив | Arr | Масив цілих чисел | 0 0 0 |

3. Функціональні вимоги до програми

Програма повинна реалізувати такі дії:

-введення даних у програму

-перевірку заданих даних

-вивід результату перевірки

-реалізація заповнення масиву випадковими числами

-вивід результату проробленого завдання над матрицею

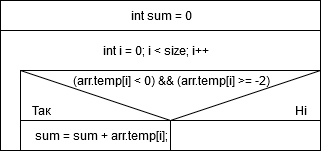
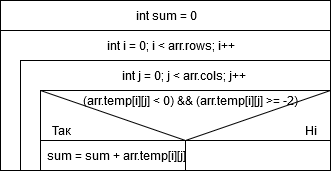
**Вибір методу рішення задачі**

Було використано математичну модель суми елементів матриці. Робилась перевірка кожного елемента та якщо умова виконувалась то сума елементів записувалась в зміну sum.

Формула за якою виконувався алгоритм: sum = sum + arr.temp[i] ( або arr.temp[i][j])

Алгоритм:

а)для одновимірного масива б)для двовимірного масива

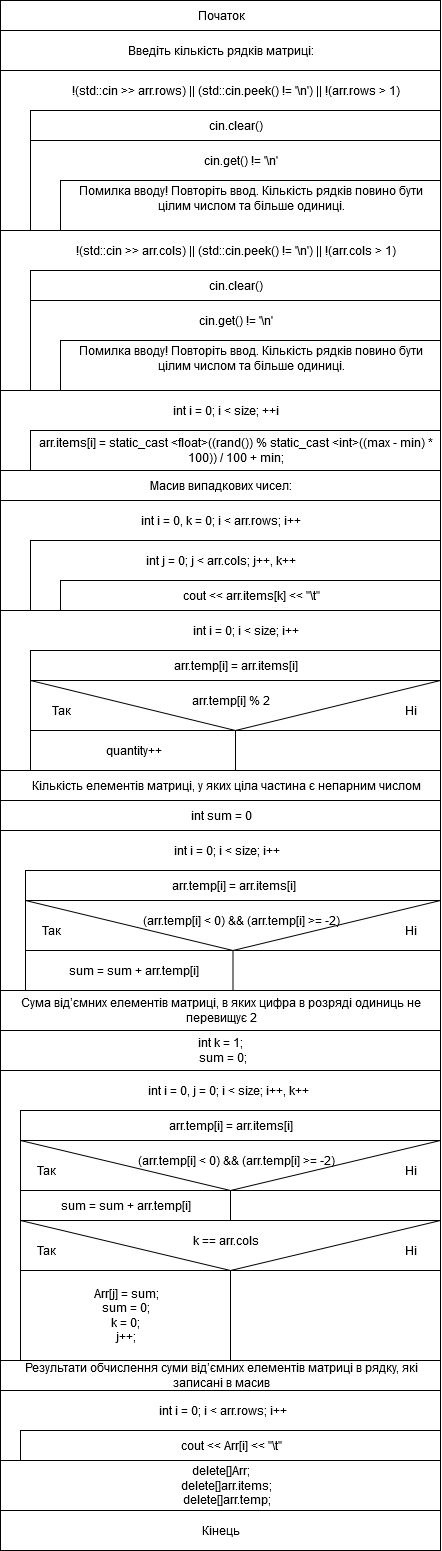
**Розробка тестів**

|  |  |  |  |
| --- | --- | --- | --- |
| № | Назва | Вхідні дані | Очікувані результати |
| 1 | Розмір матриці: |  |  |
|  | Квадратна | arr.rows = 4, arr.cols = 4 | Буде сформована випадкова матриця та зроблені всі необхідні дії над нею. |
|  | Рядків більше стовпців | arr.rows = 4, arr.cols = 2 | Буде сформована випадкова матриця та зроблені всі необхідні дії над нею. |
|  | Рядків менше стовпців | arr.rows = 2, arr.cols = 4 | Буде сформована випадкова матриця та зроблені всі необхідні дії над нею. |
| 2 | Некоректний розмір матриці | arr.rows = d, arr.cols = g  (arr.rows = 1, arr.cols = 1) | Помилка вводу! Повторіть ввод. Кількість стовпців повинно бути цілим числом та більше одиниці. |

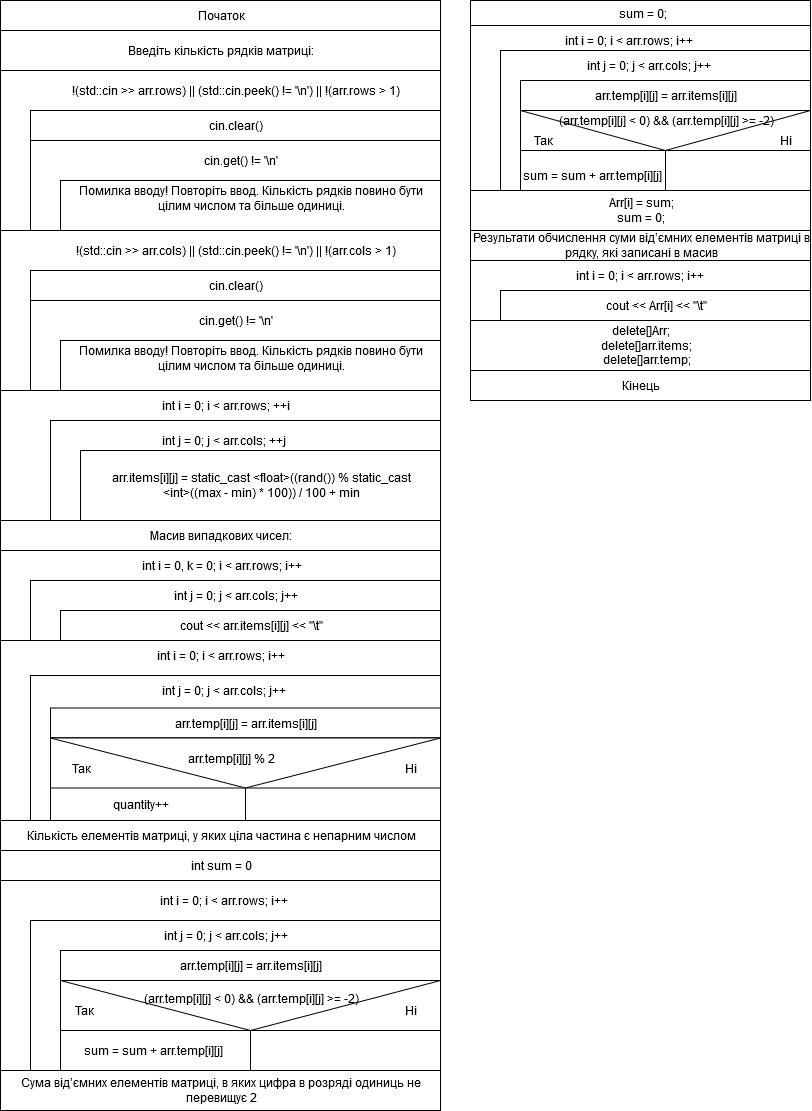
**Алгоритм програм**

1.Діаграма Н-Ш

Перша програма



Друга програма:



2.Текст програми

Перша програма:

/\* Вариант 9

Реалізувати заповнення масиву випадковими числами.

З елементів масива робимо наступні дії:

- кількість елементів матриці, у яких ціла частина є непарним числом

- сума від’ємних елементів матриці, в яких цифра в розряді одиниць не перевищує 2

- результати обчислення суми від’ємних елементів матриці в рядку, які записані в масив

- відсортувати рядки матриці за зростанням (непарний варіант) відповідних елементів масиву, створеного у п.С.

- видалити з матриці рядки, в яких перший елемент менше другого елементу

\*/

#include <iostream>

#include <Windows.h>

#include <ctime>

struct Matrix {

int rows;

int cols;

float\* items;

int\* temp;

};

const float min = -22.20;

const float max = 10.25;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

Matrix arr;

int quantity = 0; //лічильник

//ввод та перевірка розміра матриці

std::cout << "Введіть кількість рядків матриці: ";

while (!(std::cin >> arr.rows) || (std::cin.peek() != '\n') || !(arr.rows > 1)) // цикл для перевірки введеної зміної на символи

{

std::cin.clear();

while (std::cin.get() != '\n');

system("cls");

std::cerr << "Помилка вводу! Повторіть ввод. Кількість рядків повино бути цілим числом та більше одиниці." << std::endl;

std::cout << std::endl;

std::cout << "Введіть кількість рядків матриці: ";

}

std::cout << "Введіть кількість стовпців матриці: ";

while (!(std::cin >> arr.cols) || (std::cin.peek() != '\n') || !(arr.cols > 1)) // цикл для перевірки введеної зміної на символи

{

std::cin.clear();

while (std::cin.get() != '\n');

system("cls");

std::cerr << "Помилка вводу! Повторіть ввод. Кількість стовпців повино бути цілим числом та більше одиниці." << std::endl;

std::cout << std::endl;

std::cout << "Введіть кількість стовпців матриці: ";

}

int size;//розмір масива

size = arr.rows \* arr.cols;

arr.items = new float[size];

arr.temp = new int[size];

srand((unsigned int)time(NULL));

for (int i = 0; i < size; ++i)

{

arr.items[i] = static\_cast <float>((rand()) % static\_cast <int>((max - min) \* 100)) / 100 + min;

}

std::cout << std::endl;

std::cout << "Масив випадкових чисел:" << std::endl;

for (int i = 0, k = 0; i < arr.rows; i++)

{

for (int j = 0; j < arr.cols; j++, k++)

{

std::cout << arr.items[k] << "\t";

}

std::cout << std::endl;

}

//////////////////////////////

// завдання E

for (int i = 0; i < size; i++)

{

arr.temp[i] = arr.items[i];

//std::cout << arr.temp[i] << " ";

if (arr.temp[i] % 2)

{

quantity++;

}

}

std::cout << std::endl;

std::cout << "Кількість елементів матриці, у яких ціла частина є непарним числом: " << quantity << std::endl;

std::cout << std::endl;

//////////////////////////////

// завдання D

int sum = 0;

for (int i = 0; i < size; i++)

{

arr.temp[i] = arr.items[i];

//std::cout << arr.temp[i] << " ";

if ((arr.temp[i] < 0) && (arr.temp[i] >= -2))

{

sum = sum + arr.temp[i];

}

}

std::cout << "Cума від’ємних елементів матриці, в яких цифра в розряді одиниць не перевищує 2: " << sum << std::endl; // можно поставити знак -

std::cout << std::endl;

//////////////////////////////

// завдання C

int\* Arr;

Arr = new int[arr.rows];// створенння масиву для суми кожного рядка

int k = 1;

sum = 0;

for (int i = 0, j = 0; i < size; i++, k++)

{

arr.temp[i] = arr.items[i];

//std::cout << arr.temp[i] << " ";

if ((arr.temp[i] < 0) && (arr.temp[i] >= -2))

{

sum = sum + arr.temp[i];

}

if (k == arr.cols)

{

Arr[j] = sum; // можно поставити знак -

sum = 0;

k = 0;

j++;

}

//cout << endl;

}

std::cout << "Результати обчислення суми від’ємних елементів матриці в рядку, які записані в масив: " << std::endl;

std::cout << std::endl;

for (int i = 0; i < arr.rows; i++)

{

std::cout << Arr[i] << "\t"; // можно поставити знак -

//cout << endl;

}

std::cout << std::endl;

delete[]Arr;

delete[]arr.items;

delete[]arr.temp;

system("pause");

return 0;

}

Друга програма:

/\* Вариант 9

Реалізувати заповнення масиву випадковими числами.

З елементів масива робимо наступні дії:

- кількість елементів матриці, у яких ціла частина є непарним числом

- сума від’ємних елементів матриці, в яких цифра в розряді одиниць не перевищує 2

- результати обчислення суми від’ємних елементів матриці в рядку, які записані в масив

- відсортувати рядки матриці за зростанням (непарний варіант) відповідних елементів масиву, створеного у п.С.

- видалити з матриці рядки, в яких перший елемент менше другого елементу

\*/

#include <iostream>

#include <Windows.h>

#include <ctime>

struct Matrix {

int rows;

int cols;

float\*\* items;

int\*\* temp;

};

const float min = -22.20;

const float max = 10.25;

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);

Matrix arr;

int quantity = 0; //лічильник

//ввод та перевірка розміра матриці

std::cout << "Введіть кількість рядків матриці: ";

while (!(std::cin >> arr.rows) || (std::cin.peek() != '\n') || !(arr.rows > 1)) // цикл для перевірки введеної зміної на символи

{

std::cin.clear();

while (std::cin.get() != '\n');

system("cls");

std::cerr << "Помилка вводу! Повторіть ввод. Кількість рядків повино бути цілим числом та більше одиниці." << std::endl;

std::cout << std::endl;

std::cout << "Введіть кількість рядків матриці: ";

}

std::cout << "Введіть кількість стовпців матриці: ";

while (!(std::cin >> arr.cols) || (std::cin.peek() != '\n') || !(arr.cols > 1)) // цикл для перевірки введеної зміної на символи

{

std::cin.clear();

while (std::cin.get() != '\n');

system("cls");

std::cerr << "Помилка вводу! Повторіть ввод. Кількість стовпців повино бути цілим числом та більше одиниці." << std::endl;

std::cout << std::endl;

std::cout << "Введіть кількість стовпців матриці: ";

}

arr.items = new float \*[arr.rows];

for (int i = 0; i < arr.rows; i++)

{

arr.items[i] = new float[arr.cols];

}

arr.temp = new int \*[arr.rows];

for (int i = 0; i < arr.rows; i++)

{

arr.temp[i] = new int[arr.cols];

}

srand((unsigned int)time(NULL));

for (int i = 0; i < arr.rows; ++i)

{

for (int j = 0; j < arr.cols; ++j)

{

arr.items[i][j] = static\_cast <float>((rand()) % static\_cast <int>((max - min) \* 100)) / 100 + min;

}

}

std::cout << std::endl;

std::cout << "Масив випадкових чисел:" << std::endl;

for (int i = 0; i < arr.rows; i++)

{

for (int j = 0; j < arr.cols; j++)

{

std::cout << arr.items[i][j] << "\t";

}

std::cout << std::endl;

}

//////////////////////////////

// завдання E

for (int i = 0; i < arr.rows; i++)

{

for (int j = 0; j < arr.cols; j++)

{

arr.temp[i][j] = arr.items[i][j];

//std::cout << arr.temp[i] << " ";

if (arr.temp[i][j] % 2)

{

quantity++;

}

}

}

std::cout << std::endl;

std::cout << "Кількість елементів матриці, у яких ціла частина є непарним числом: " << quantity << std::endl;

std::cout << std::endl;

//////////////////////////////

// завдання D

int sum = 0;

for (int i = 0; i < arr.rows; i++)

{

for (int j = 0; j < arr.cols; j++)

{

arr.temp[i][j] = arr.items[i][j];

//std::cout << arr.temp[i] << " ";

if ((arr.temp[i][j] < 0) && (arr.temp[i][j] >= -2))

{

sum = sum + arr.temp[i][j];

}

}

}

std::cout << "Cума від’ємних елементів матриці, в яких цифра в розряді одиниць не перевищує 2: " << sum << std::endl; // можно поставити знак -

std::cout << std::endl;

//////////////////////////////

// завдання C

int\* Arr;

Arr = new int[arr.rows];// створенння масиву для суми кожного рядка

sum = 0;

for (int i = 0; i < arr.rows; i++)

{

for (int j = 0; j < arr.cols; j++)

{

arr.temp[i][j] = arr.items[i][j];

//std::cout << arr.temp[i] << " ";

if ((arr.temp[i][j] < 0) && (arr.temp[i][j] >= -2))

{

sum = sum + arr.temp[i][j];

}

}

Arr[i] = sum; // можно поставити знак -

sum = 0;

}

std::cout << "Результати обчислення суми від’ємних елементів матриці в рядку, які записані в масив: " << std::endl;

std::cout << std::endl;

for (int i = 0; i < arr.rows; i++)

{

std::cout << Arr[i] << "\t"; // можно поставити знак -

//cout << endl;

}

std::cout << std::endl;

//////////////////////////////

delete[]Arr;

delete[]arr.items;

delete[]arr.temp;

system("pause");

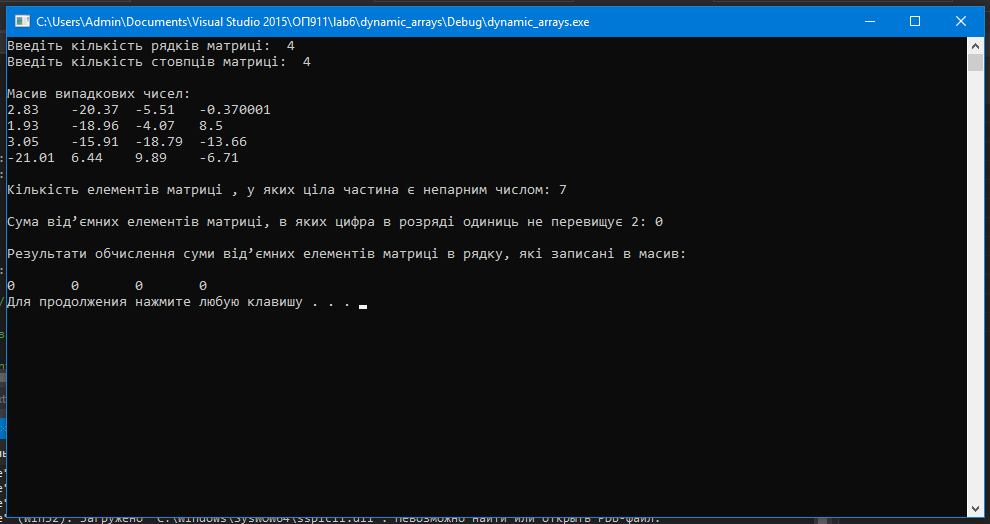
return 0;

}

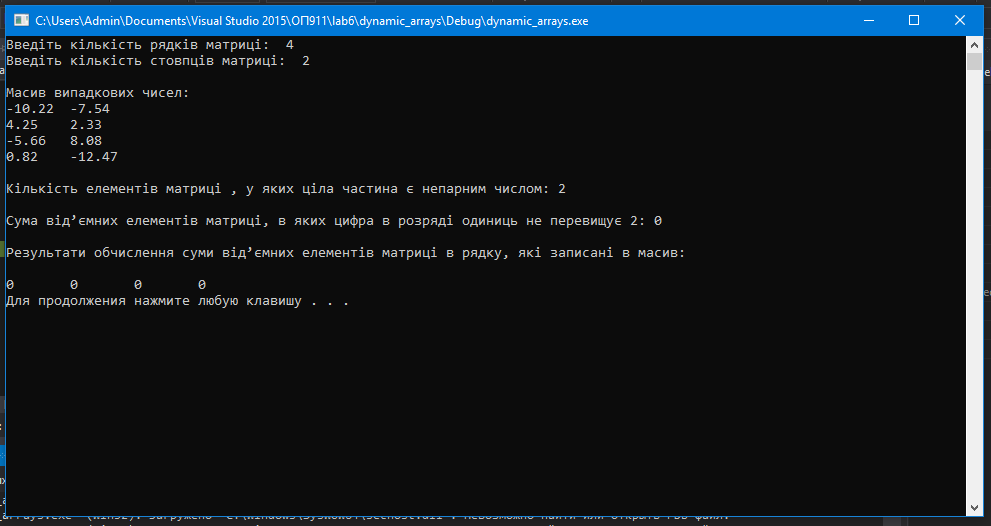
4.Результати виповнення програми

Перший тест:

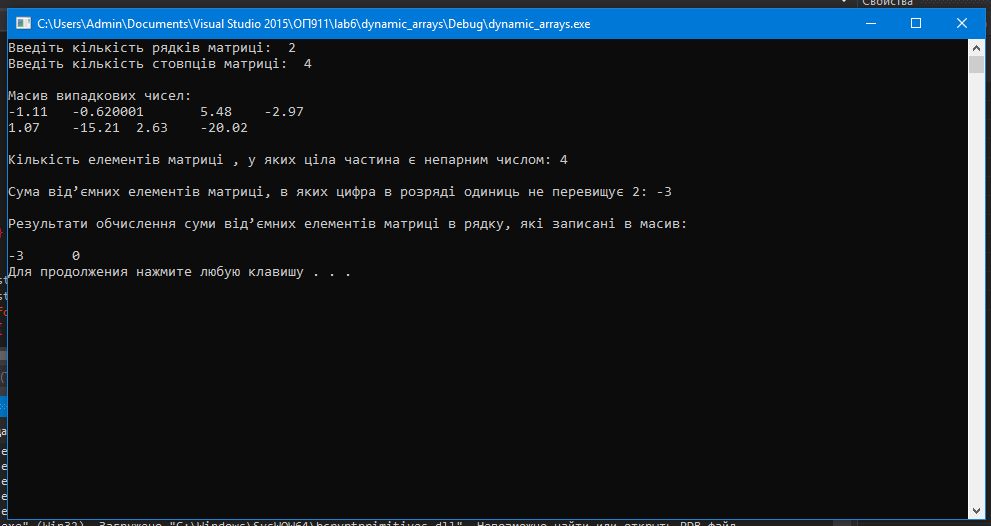
а)



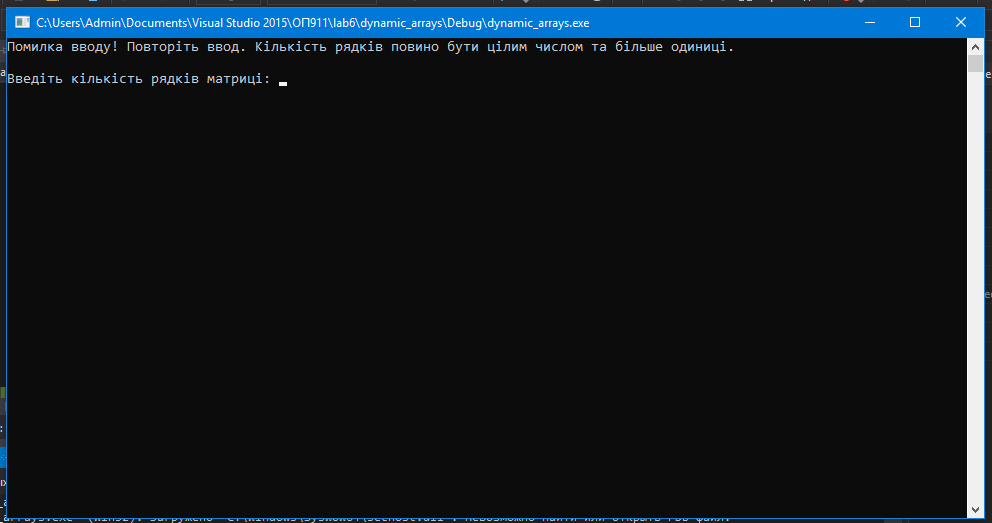
б)



в)



Другий тест



Висновок: динамічним називають такий [масив](https://uk.wikipedia.org/wiki/%D0%9C%D0%B0%D1%81%D0%B8%D0%B2_(%D1%81%D1%82%D1%80%D1%83%D0%BA%D1%82%D1%83%D1%80%D0%B0_%D0%B4%D0%B0%D0%BD%D0%B8%D1%85)), розмір якого можна змінювати під час виконання програми. Динамічні масиви надають змогу більш гнучко працювати з даними, оскільки дозволяють вводити довільний розмір. В даній лабораторній роботі ми створювали динамічна масиви, перезаписували їх та робили над ними різні дії, такі як:

-перевірка елементів масива на парність;

-сума елементів масива, які в межах перевірки.