ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №8

За курсом «Програмування»

Студента групи ПА-19-2

Москаленка Сергія Костянтиновича

Кафедра комп’ютерних технологій ДНУ

2020/2021 навч. рік

1. Постановка здачі

Мій варіант:(2011)

Задайте матрицю А розмірністю (n x m), де n і m ≥ 6 (n!=m). Виконайте такі завдання:

Завдання 1:

Обчисліть добуток всіх елементів матриці, що належать рядкам із парними індексами;

Завдання 2:

Знайдіть мінімальний елемент зі всіх елементів матриці, обидва індекси яких непарні;

Завдання 3:

Упорядкуйте за зростанням значення елементів рядка, де знаходиться мінімальний елемент.

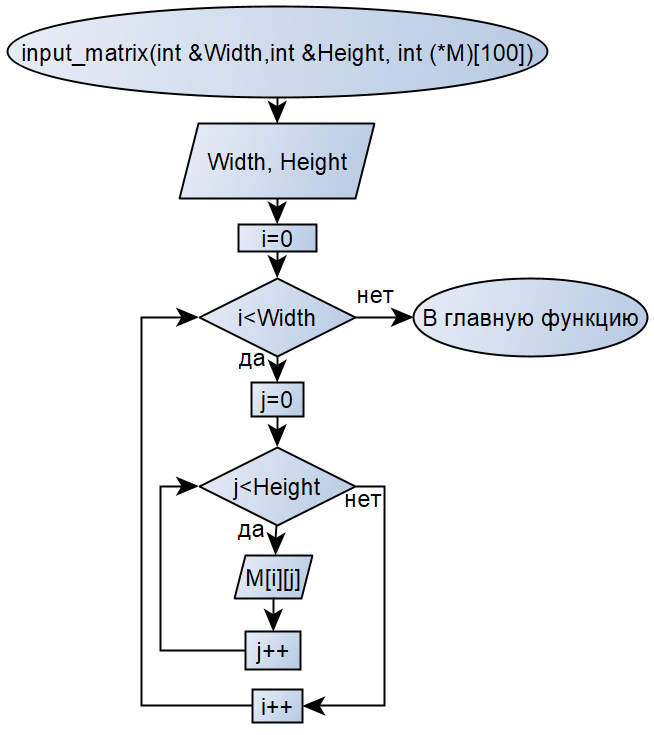
1. Опис розв’язку

main(int argc, char \*argv) – Головна функція, з якої все починається.

Для розв’язання даної задачі я розбив її на де-кілька підзадач - зробив окремі функції:

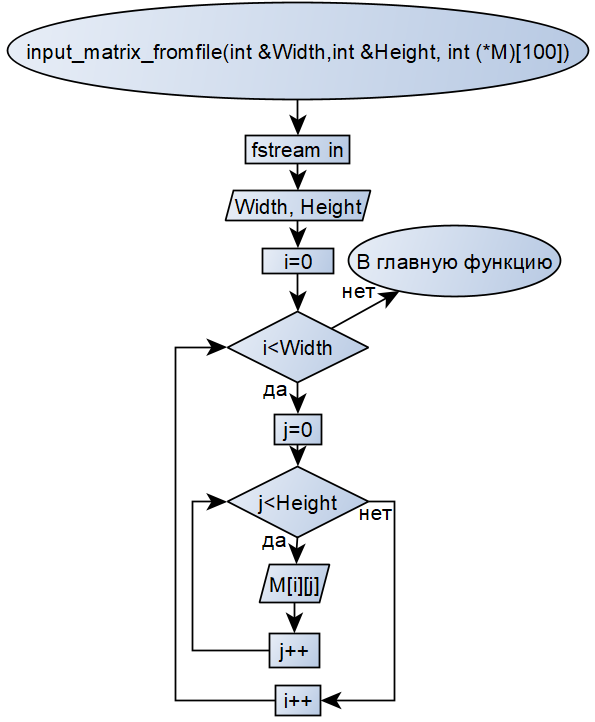
input\_matrix(int &Width,int &Height, int (\*M)[100]) – Викликається з мейну. У вхідних параметрах – розміри матриці та сама матриця. Через цю функцію користувач з консолі вводить матрицю з якою потім буде працювати.

Функція повертає нові розміри матриці та нову матрицю.



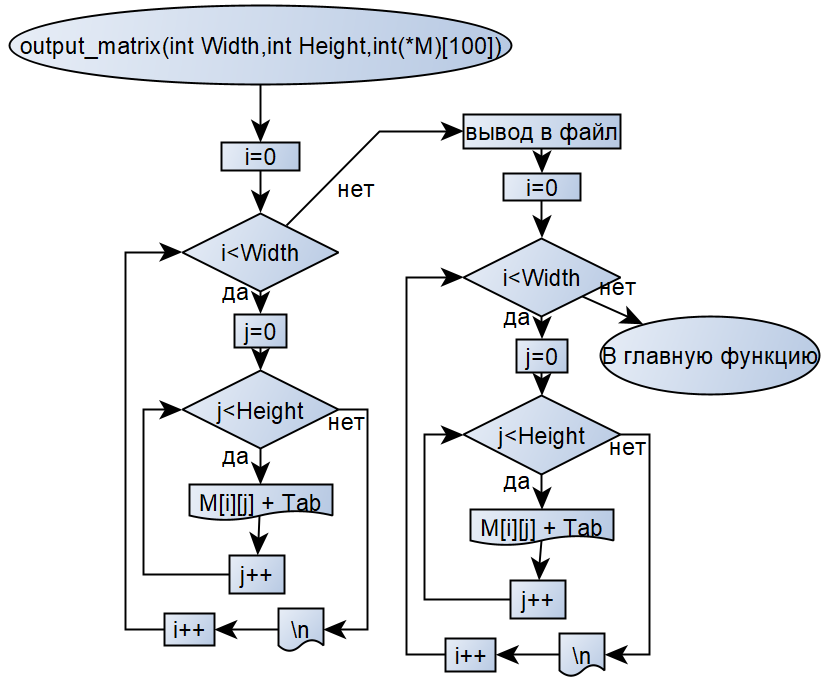
input\_matrix\_fromfile(int &Width,int &Height, int (\*M)[100]) – Викликається з мейну. У вхідних параметрах – розміри матриці та сама матриця. Через цю функцію користувач з файлу під назвою «Ввод данных» вводить матрицю з якою потім буде працювати.

Функція повертає нові розміри матриці та нову матрицю.



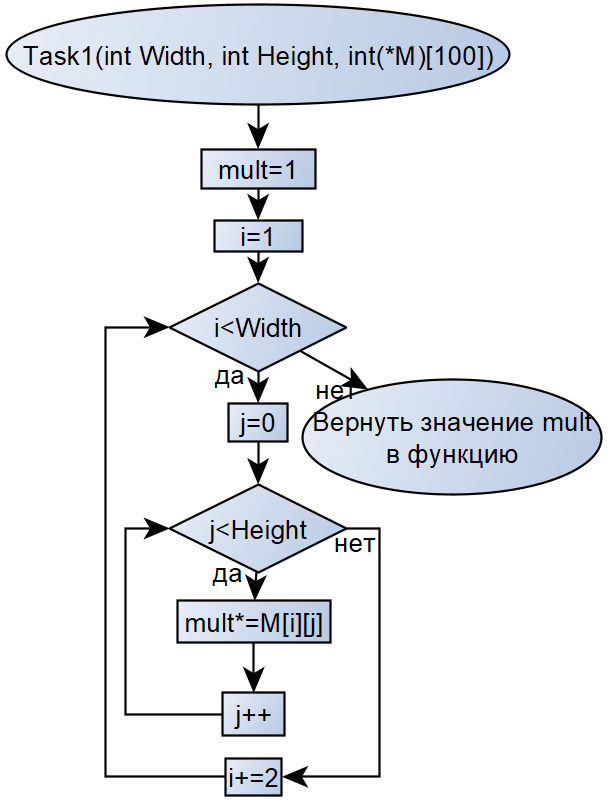
output\_matrix(int Width,int Height, int (\*M)[100]) – Викликається з мейну. У вхідних параметрах – розміри матриці та сама матриця. Через цю функцію користувач виводить оброблену матрицю на консоль та у файл під назвою «Вывод данных».

Функція нічого не повертає.



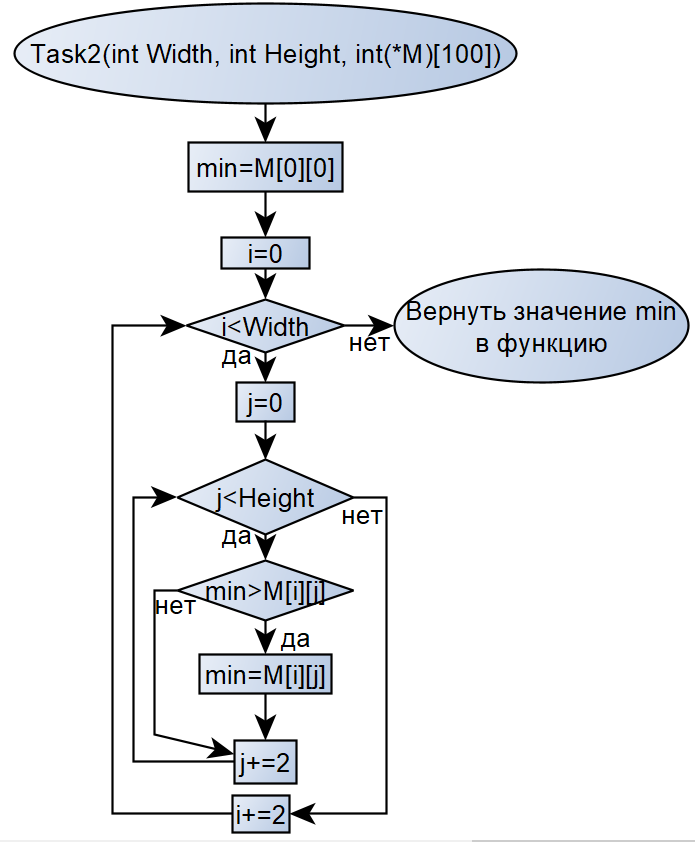
Task1(int Width, int Height, int(\*M)[100]) – Викликається з мейну. У вхідних параметрах – розміри матриці та сама матриця. Функція робить завдання1 і повертає відповідні данні до мейну.

Функція повертає добуток всіх елементів на непарних рядках.



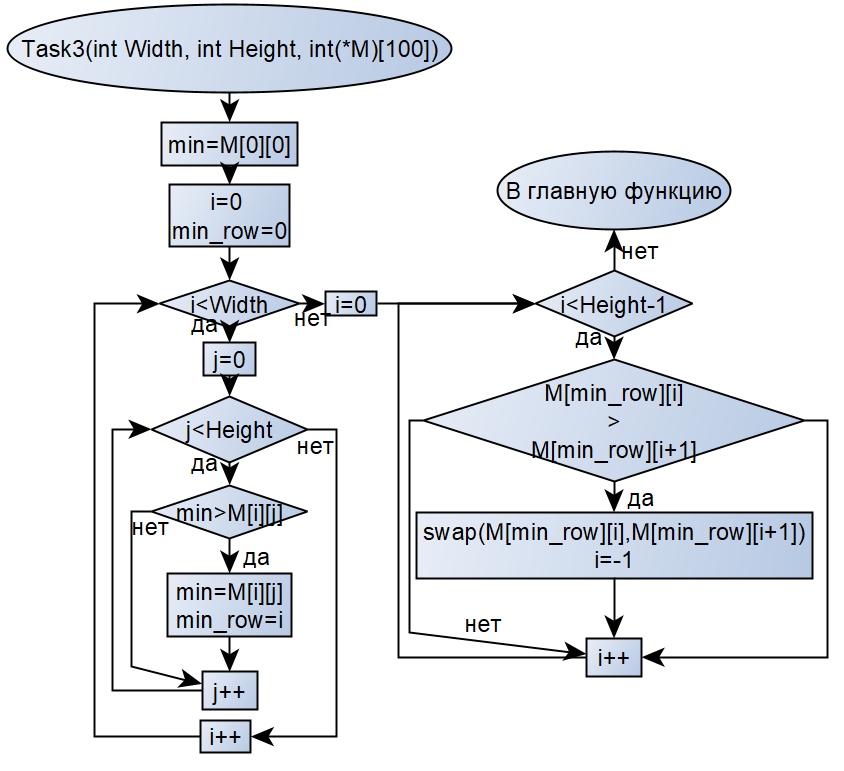
Task2(int Width, int Height, int(\*M)[100]) – Викликається з мейну. У вхідних параметрах – розміри матриці та сама матриця. Функція шукає мінімальній елемент матриці на непарних індексах.

Функція повертає відповідні данні до мейну.

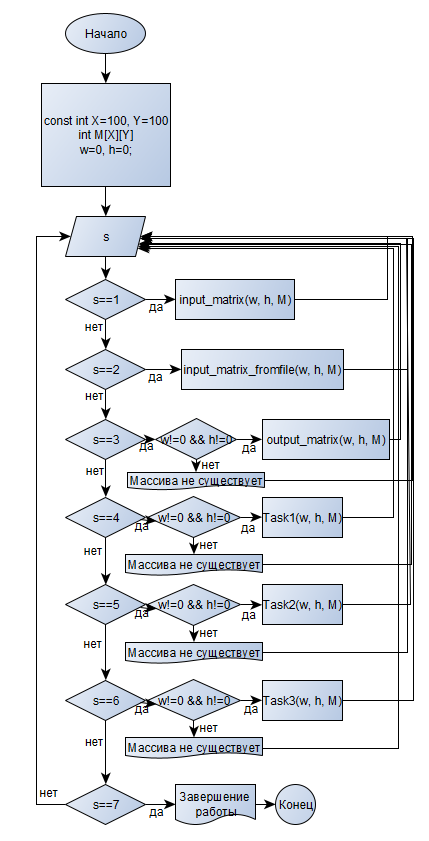


Task3(int Width, int Height, int(\*M)[100]) – Викликається з мейну. У вхідних параметрах – розміри матриці та сама матриця. Функція шукає рядок з найменшим елементом та сортує цей рядок за зростанням.

Функція повертає змінену матрицю.



У комбінації з мейном всі ці функції пов’язані так:



1. Вихідний текст програми розв’язку задачі

#include <iostream>

#include <fstream>

using namespace std;

//Variant 15

void input\_matrix(int &Width,int &Height, int (\*M)[100])

{//вводим матрицу вручную

cout << "Введите высоту матрицы (больше 5):\n";

cin >> Width;

cout << "Введите ширину матрицы (больше 5 и не = "<<Width<<"):\n";

cin >> Height;

cout << "Введите матрицу:\n";

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

for (int j = 0; j < Height; j++)

cin >> M[i][j];

cout << "\nМатрица успешно заполнена!\n";

}

void input\_matrix\_fromfile(int& Width, int& Height, int(\*M)[100])

{//вводим матрицу из файла

fstream in("Вводные данные.txt", ios::in);

if (!in.eof())

{

in >> Width;

in >> Height;

int i = 0, j = 0;

while (in >> M[i][j])

{

j++;

if (j > Height - 1)

{

j = 0;

i++;

}

}

if (j != 0 || i != Width)

{

cout << "В файле недостаточное количество элементов " <<endl;

while (i<Height || j<Width)

{

M[i][j] = 0;

j++;

if (j > Height - 1)

{

j = 0;

i++;

}

}

}

}

else

{

cout << "Файл пуст!";

}

in.close();

}void output\_matrix(int Width,int Height,int(\*M)[100])

{

//выводим матрицу на консоль

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

{

for (int j = 0; j < Height; j++)

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

cout << "\n";

}

//...и на выводной файл

fstream f("Выводные данные.txt",ios::out);

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

{

for (int j = 0; j < Height; j++)

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

f << "\n";

}

f.close();

}

double Task1(int Width, int Height, int(\*M)[100])

{

double mult = 1;

for (int i = 1; i < Width; i+=2)

for (int j = 0; j < Height; j++)

mult \*= M[i][j];

return mult;

}

int Task2(int Width, int Height, int(\*M)[100])

{

int min\_el = M[0][0];

for (int i = 0; i < Width; i += 2)

for (int j = 0; j < Height; j += 2)

if (min\_el > M[i][j])

min\_el = M[i][j];

return min\_el;

}

void Task3(int Width, int Height, int(\*M)[100])

{

//поиск минимального элемента:

int min\_row = 0,min\_el=M[0][0];

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

for (int j = 0; j < Height; j++)

if (min\_el > M[i][j])

{

min\_el = M[i][j];

min\_row = i;

}

//сортировка от меньшего к большему рядка с индексом min\_row

for (int i = 0; i < Height-1; i++)

if (M[min\_row][i] > M[min\_row][i + 1])

{//если 2 соседних ел. такие, что:

//левый>правого, то меняем их местами и начинаем сортировку с начала

M[min\_row][i] += M[min\_row][i + 1];

M[min\_row][i + 1] = M[min\_row][i] - M[min\_row][i + 1];

M[min\_row][i] -= M[min\_row][i + 1];

i = -1;

}

cout << "\nСортировка прошла успешно!\n\n";

}

int main()

{

setlocale(LC\_ALL, "Russian");

ifstream file;

file.open("Вводные данные.txt");

file.close();

if (!file)

{//если нету файлов для ввода/вывода - создать их

ofstream f;

f.open("Вводные данные.txt");

f << "";

f.close();

}

file.open("Выводные данные.txt");

file.close();

if (!file)

{

ofstream f;

f.open("Выводные данные.txt");

f << "";

f.close();

}

int s = 0;

const int X = 100, Y = 100;

int w = 0, h = 0;

int M[X][Y];

do

{

cout << "Введите 1, для ручного введения матрицы\n";

cout << "Введите 2, для введения матрицы с помощью файла\n";

cout << "Введите 3, для демонстрации матрицы\n";

cout << "Введите 4, для выполнения задания№1\n";

cout << "Введите 5, для выполнения задания№2\n";

cout << "Введите 6, для выполнения задания№3\n";

cout << "Введите 7, для завершения работы\n";

cout << "Ввод> ";

cin >> s;

if (s == 1)

input\_matrix(w, h, M);

else if (s == 2)

input\_matrix\_fromfile(w, h, M);

else if (s == 3)

{

if (w != 0 && h != 0)

output\_matrix(w, h, M);

else

cout << "\nДля демонстрации матрицы необходимо создать матрицу\n\n";

}

else if (s == 4)

{

if (w != 0 && h != 0)

cout << "\nПроизведение всех эл., находящихся на четных рядках = " << Task1(w, h, M)<<"\n\n";

else

cout << "\nДля нахождения произведения ел. матрицы необходимо создать матрицу\n\n";

}

else if (s == 5)

{

if (w != 0 && h != 0)

cout << "\nМинимальный элемент с нечетными индексами = " << Task2(w, h, M)<<"\n\n";

else

cout << "\nДля сортировеи матрицы необходимо создать матрицу\n\n";

}

else if (s == 6)

{

if (w != 0 && h != 0)

Task3(w, h, M);

else

cout << "\nДля сортировеи матрицы необходимо создать матрицу\n\n";

}

else if (s == 7)

cout << "\nЗавершение работы\n\n";

else

cout << "\nТакой команды нету\n\n";

} while (s != 7);

return 0;

}

1. Опис інтерфейсу

Перед розв’язуванням задачі користувач має ввести одну із семи команд:

1) Ввести 1 – для ручного вводу матриці

2) Ввести 2 – для вводу матриці з підготовленого файла:

Перші дві цілі цифри – розміри матриці (висота і ширина матриці відповідно), інші цифри – елементи матриці.

3) Ввести 3 – для демонстрації матриці.

4) Ввести 4 – для пошуку добутку всіх елементів на парних рядках.

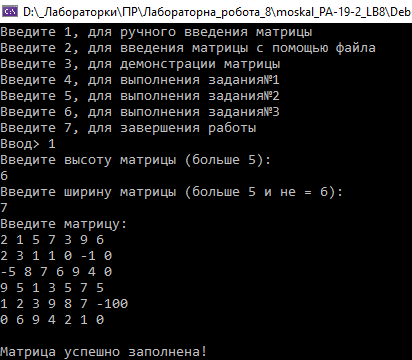
5) Ввести 5 – для пошуку мінімального елементу з двома непарними індексами.

6) Ввести 6 – для сортування рядка з мінімальним елементом за зростанням.

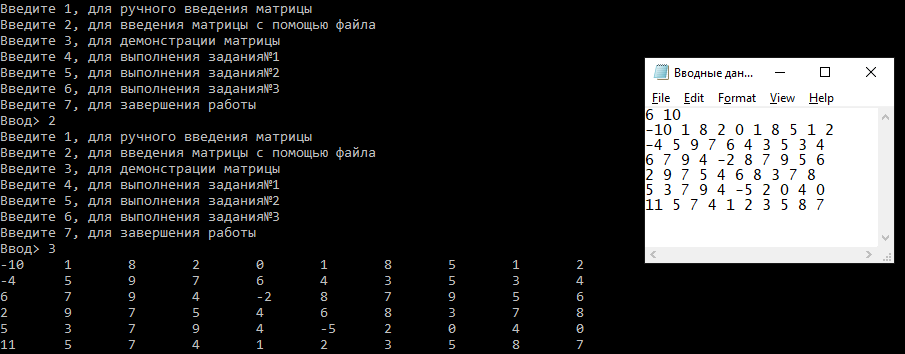
7) Ввести 7 – для завершення роботи програми.

1. Опис тестових прикладів

Демонстрація вводу матриці вручну:

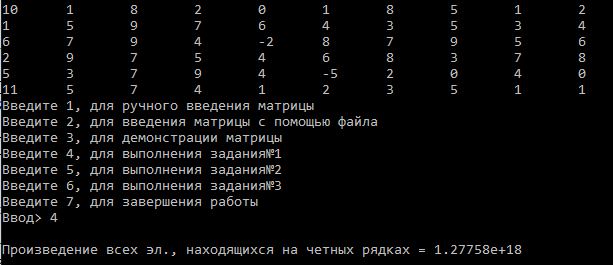


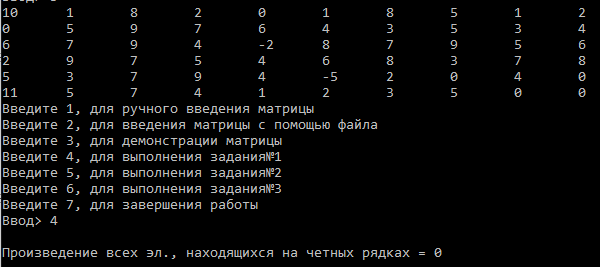
Демонстрація вводу матриці з текстового документу:



Задача 1:

1)

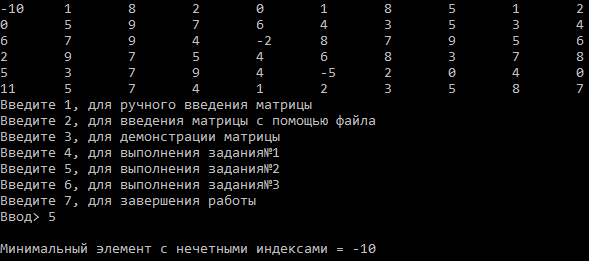


2) 

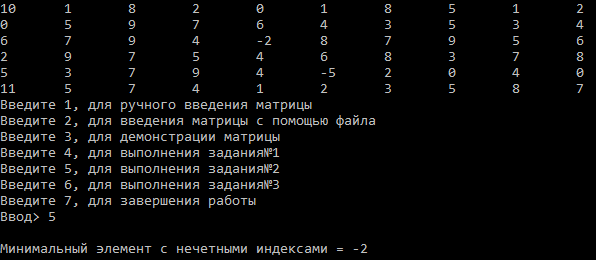
Для перевірки поміняємо (-4) на 0 і бачимо, що дійсно: перемножуються тільки парні рядки.

Завдання 2:

1)

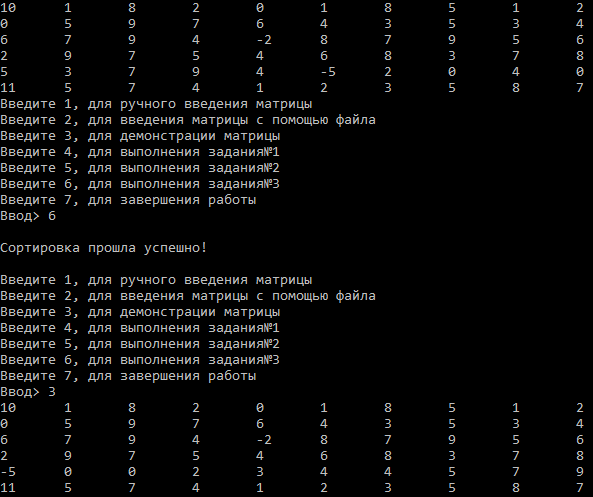


2)



Завдання 3:

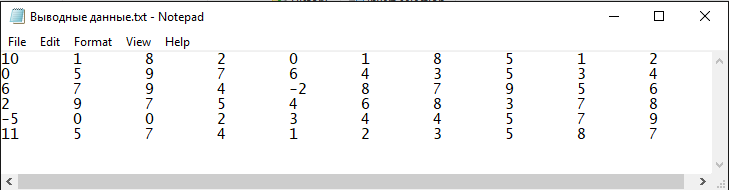
1)



Було відсортовано на зростанням п’ятий рядок, оскільки (-5) – найменший елемент матриці і знаходиться у п’ятому рядку.

Вивід відповіді до текстового файлу

З минулого прикладу – при виводі матриці на консоль – вона водночас виводиться до файлу:



У цій версії було змінено ввід матриці з файлу ([функція](#input_from_file)):

1) перевірка, чи пустий файл

2) перевірка, чи всі елементи були введені. Якщо – ні, то недозаповнені елементи замінюємо на нулі.