Московский авиационный институт

(Национальный исследовательский университет)

Факультет №3:

*Системы управления, информатика и электроэнергетика*.

Кафедра 304.

Отчёт по лабораторной работе

По учебной дисциплине «Программирование»

На тему

«Матрицы»

Группа: М3О-107Б-23

Вариант №6

Выполнили:

*Романов Д.И.*

*Ильин А.А.*

Приняла:

*Татарникова Е.М.*

Оглавление

[Постановка задачи: 4](#_Toc181646517)

[Блок-Схема 5](#_Toc181646518)

[Кодпрограммы 11](#_Toc181646519)

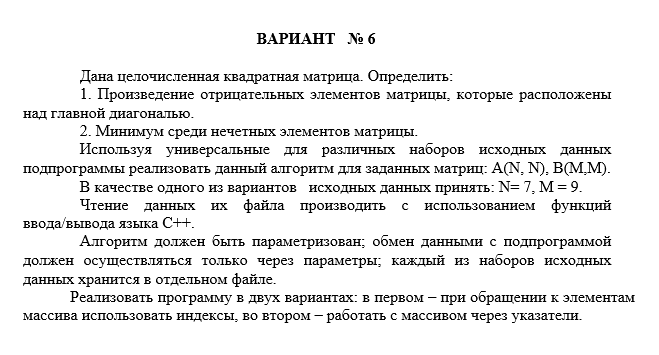
[Тестирование 16](#_Toc181646520)

[Область некорректных данных 16](#_Toc181646521)

[Область корректных данных 18](#_Toc181646522)

[Вывод 21](#_Toc181646523)

# Постановка задачи:



# Блок-Схема

Функция main

int main(int argc, char \*\*argv)

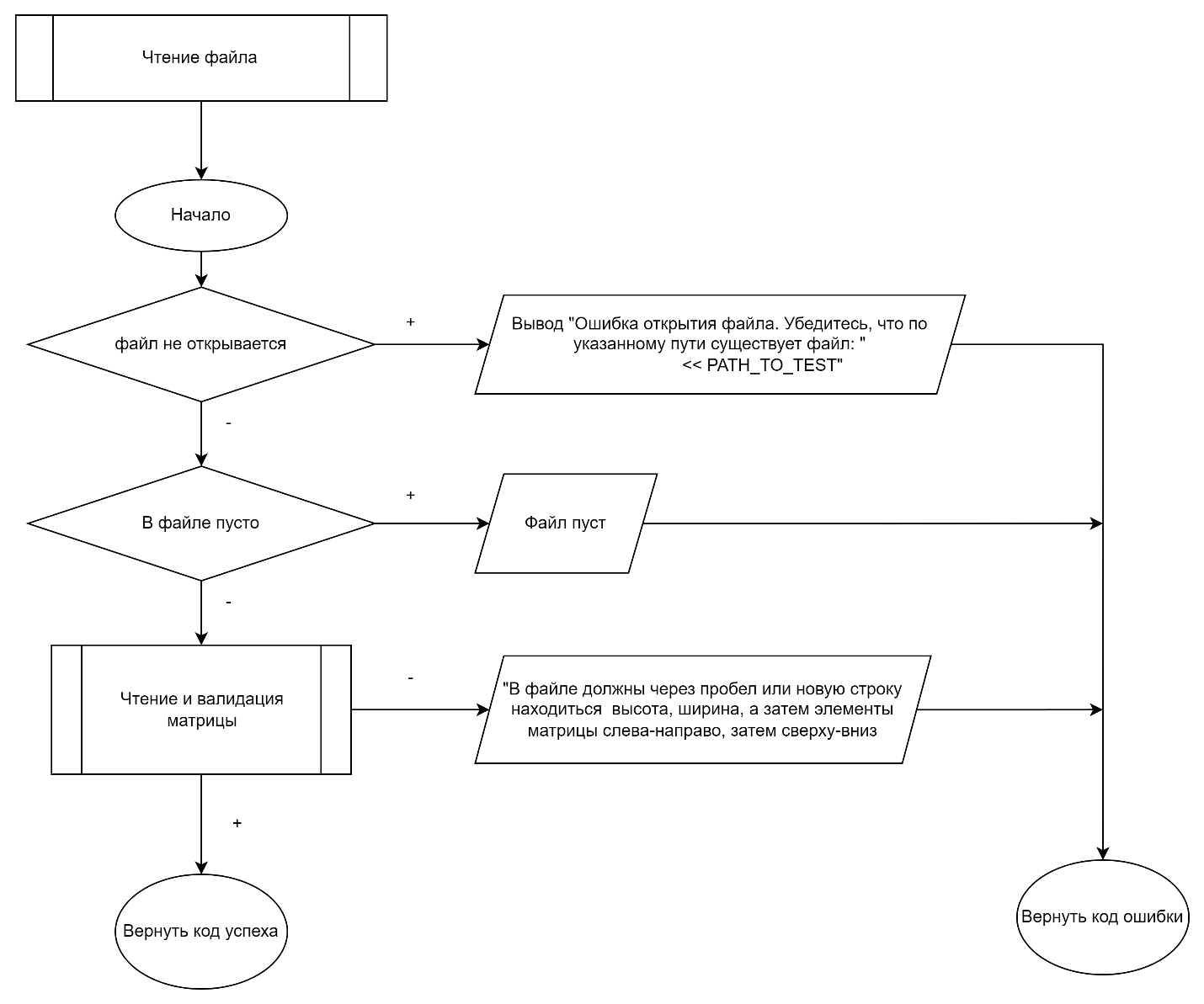
Функция обработки матрицы

void process\_matrix(int\*\* matrix, int height, int width)

|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип | Назначение | Входной/выходной |
| matrix | указатель на указатель на целое число | Хранение матрицы | Входной |
| height | целое число | Хранение высоты  матрицы | Входной |
| width | целое число | Хранение ширины  матрицы | Входной |

Функция чтения данных из файла

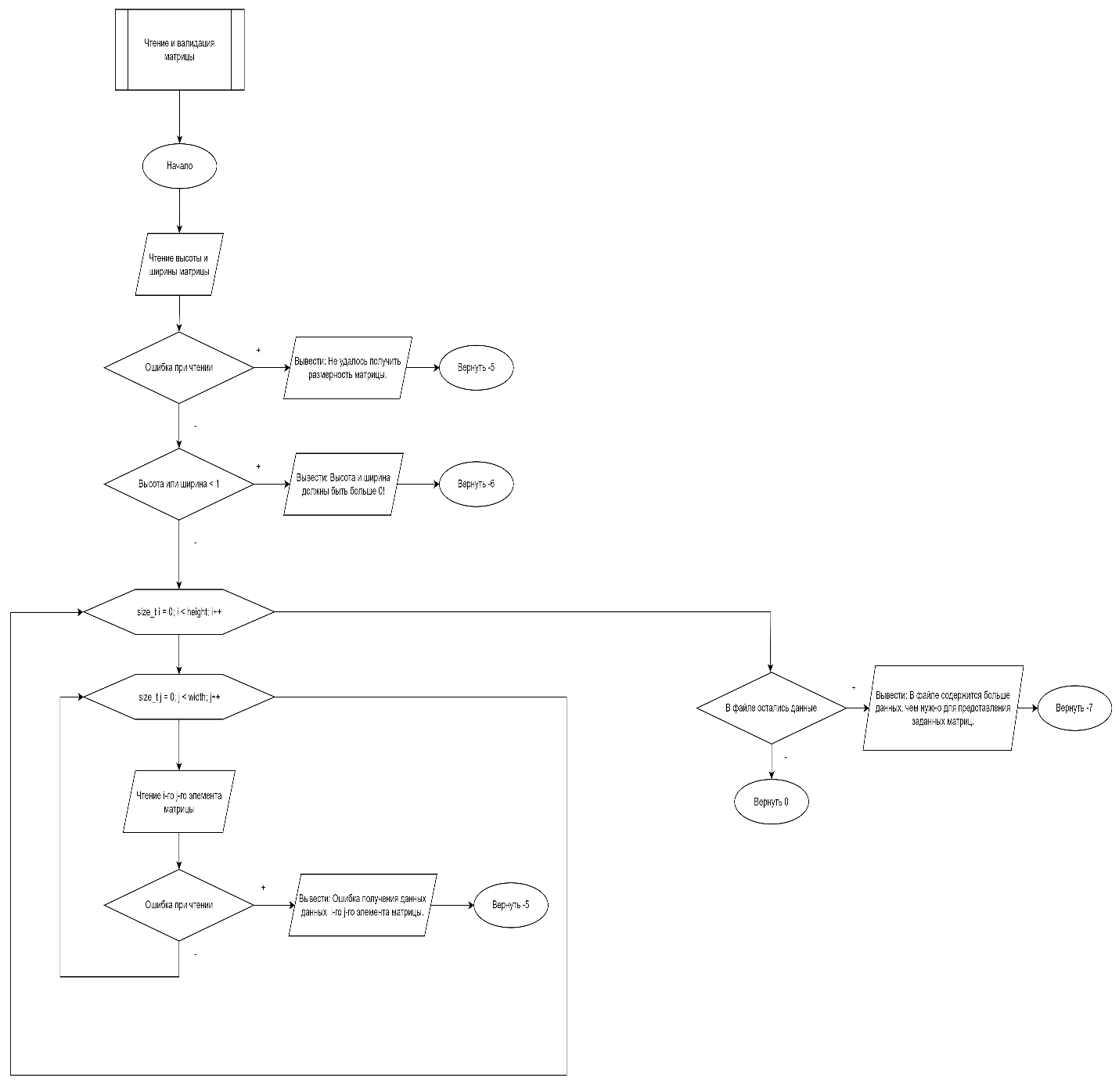
read\_file(const char\* file\_name, int\*\* &matrix, int &height, int &width);



|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип | Назначение | Входной/выходной |
| file\_name | Константная строка | Хранение матрицы | Входной |
| matrix | указатель на указатель на целое число | Хранение матрицы | Входной |
| height | Ссылка на целое число | Хранение высоты  матрицы | Входной |
| width | Ссылка на целое число | Хранение ширины  матрицы | Входной |
| - | Целое число | Код возврата | выходной |

Функция чтения и валидации матрицы

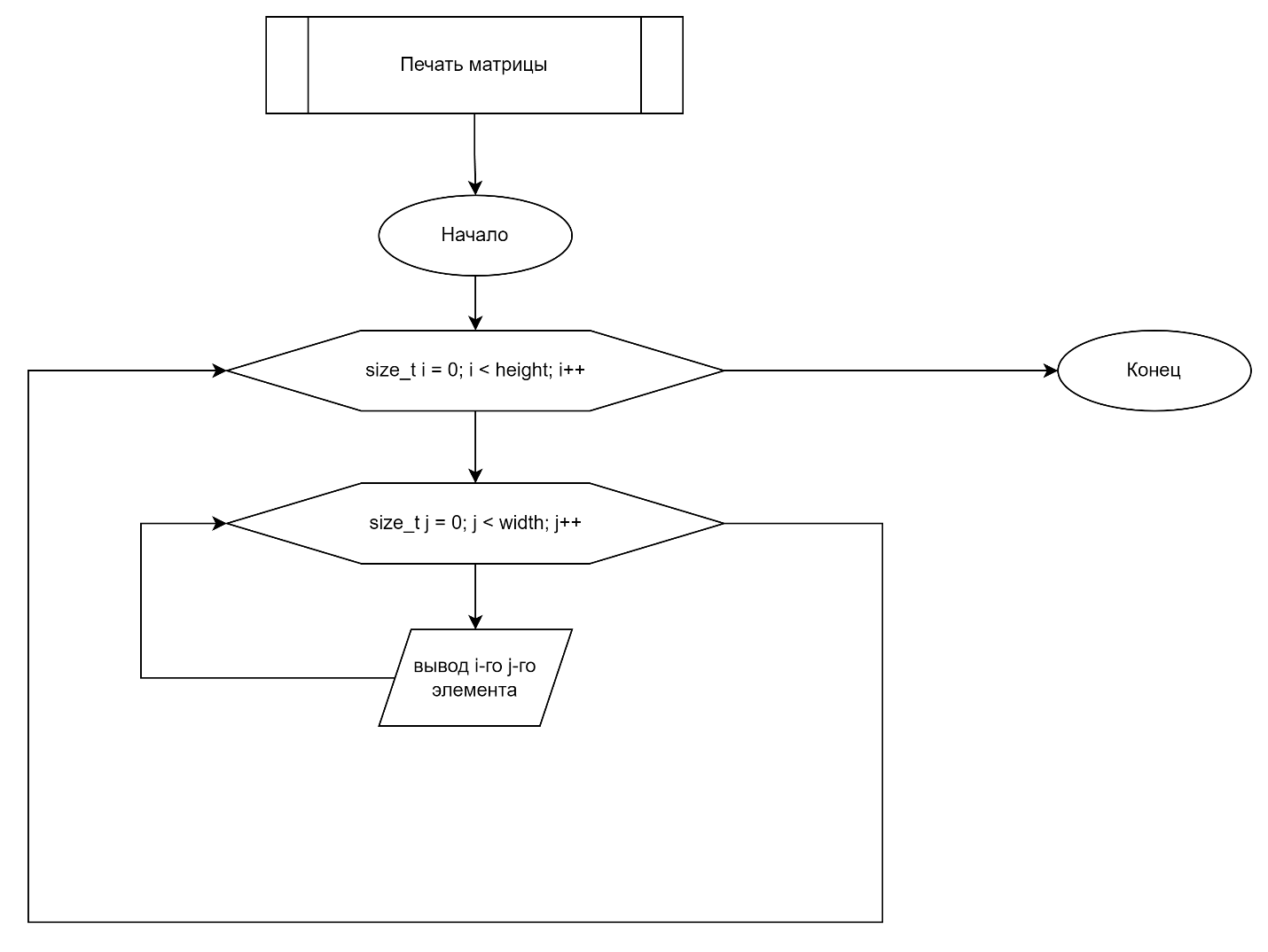
int validate\_and\_read\_matrix(std::ifstream& in, int\*\* &matrix, int &height, int &width);



|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип | Назначение | Входной/выходной |
| in | Ссылка на поток | Чтение данных из файла | Входной |
| matrix | Ссылка на указатель на указатель на целое число | Хранение матрицы | Входной |
| height | Ссылка на целое число | Хранение высоты  матрицы | Входной |
| width | Ссылка на целое число | Хранение ширины  матрицы | Входной |
| - | Целое число | Код завершения работы | Выходной |

Функция печати матрицы

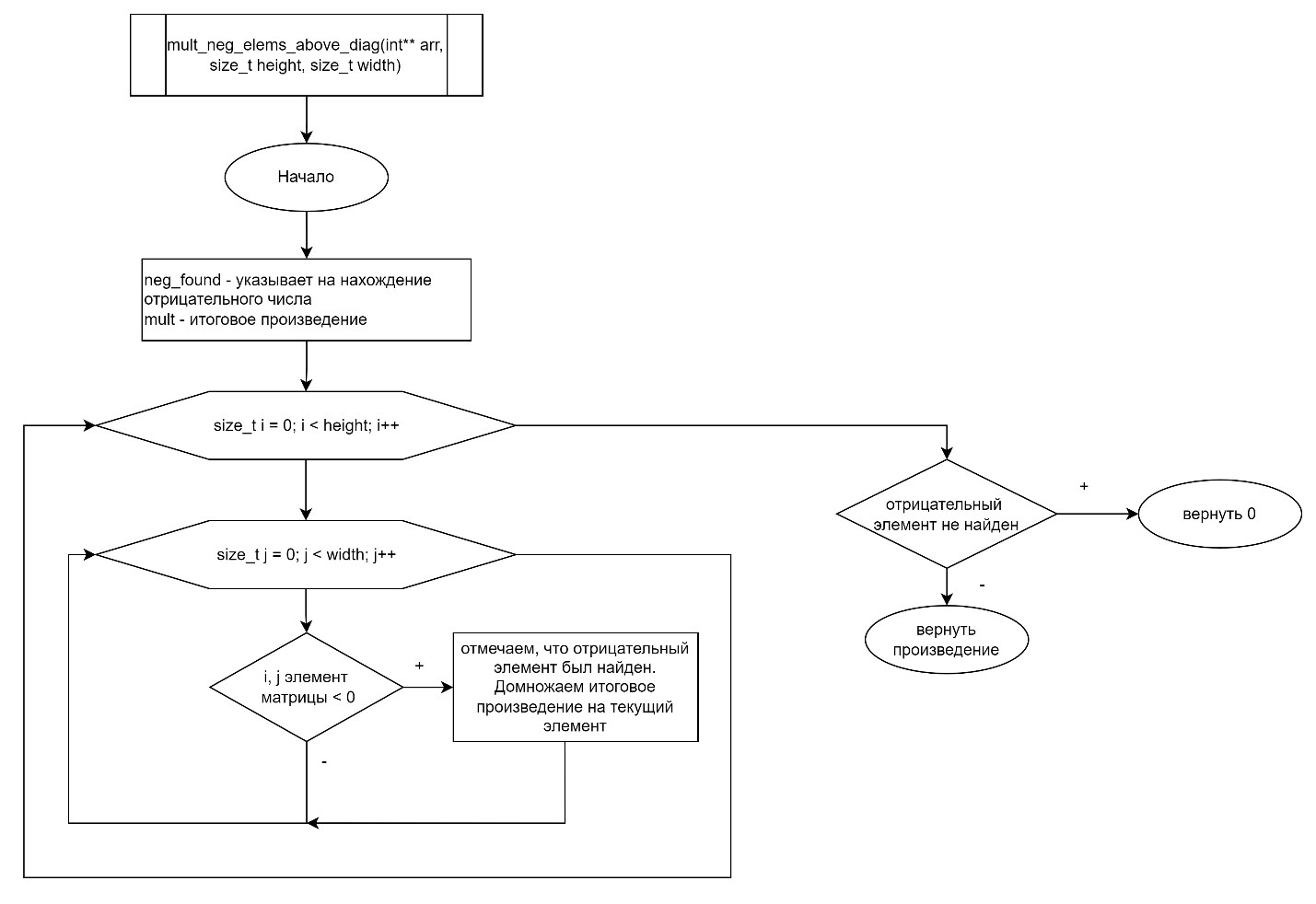
void print\_matrix(int const \*const \*matrix, int height, int width);



|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип | Назначение | Входной/выходной |
| matrix | Конст. указатель на конст. указатель на целое число | Хранение матрицы | Входной |
| height | Целое число | Хранение высоты  матрицы | Входной |
| width | Целое число | Хранение ширины  матрицы | Входной |

Функция вычисления произведения негативных эл-в над главной диагональю

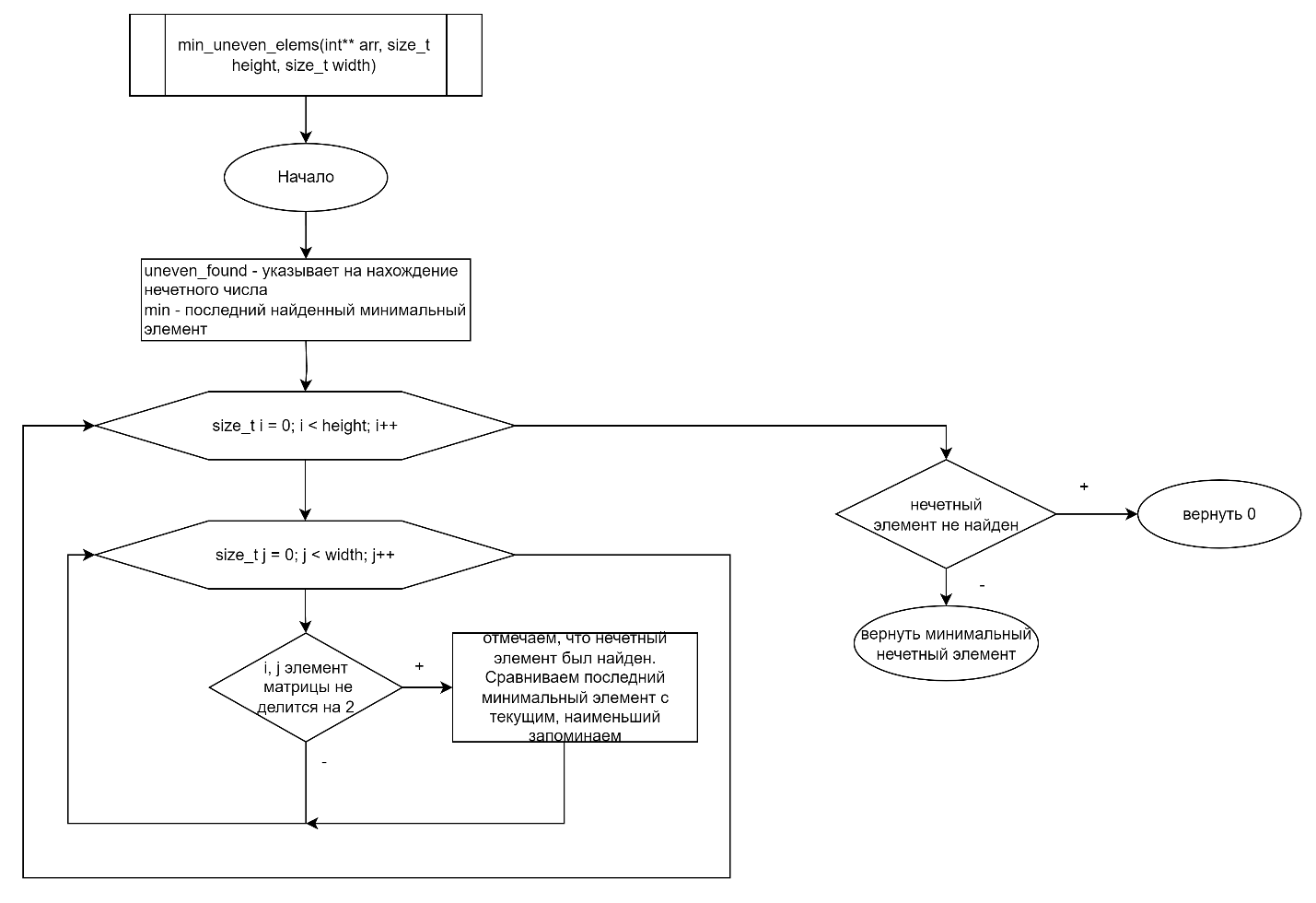
int mult\_neg\_elems\_above\_diag(int const \*const \*matrix, int height, int width);



|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип | Назначение | Входной/выходной |
| arr | Конст. указатель на конст. указатель на целое число | Хранение матрицы | Входной |
| height | Целое число | Хранение высоты  матрицы | Входной |
| width | Целое число | Хранение ширины  матрицы | Входной |
| - | Целочисленный | Итоговое произведение | Выходной |

Функция для поиска минимального нечетного эл-та

int min\_uneven\_elems(int const \*const \*matrix, int height, int width);



|  |  |  |  |
| --- | --- | --- | --- |
| Название | Тип | Назначение | Входной/выходной |
| arr | Конст. указатель на конст. указатель на целое число | Представляет собой матрицу | Входной |
| height | Целое число | Хранение высоты  матрицы | Входной |
| width | Целое число | Хранение ширины  матрицы | Входной |
|  | Целое число | Минимальное нечетное | Выходной |

# Кодпрограммы

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* КАФЕДРА № 304 1 КУРС \*

\*---------------------------------------------------------------\*

\* Project Type : Win32 Console Application \*

\* Project Name : LW2 \*

\* File Name : LW2.cpp \*

\* Language : C/C++ \*

\* Programmer(s) : Романов Д.И., Ильин А.А \*

\* Created at : 20/02/24 \*

\* Last Revision : 05/03/24 \*

\* Comment(s) : Двумерные массивы \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

#include <iostream>

#include <fstream>

#include <limits>

// all matrices are in m[h][w] format

// reads the file and sets the matrix data. returns an error code

int read\_file(const char\* file\_name, int\*\* &matrix, int &height, int &width);

// processes the matrix using functions from the lw

void process\_matrix(int\*\* matrix, int height, int width);

// initializes matrix, then validates and reads a matrix from an ifstream. returns an error code

int validate\_and\_read\_matrix(std::ifstream& in, int\*\* &matrix, int &height, int &width);

// prints a matrix

void print\_matrix(int const \*const \*matrix, int height, int width);

// returns a multiple of all negative elems above the main diagonal. returns 0 if no negative elements found

int mult\_neg\_elems\_above\_diag(int const \*const \*matrix, int height, int width);

// returns the minimum of all uneven elements. return 0 if no uneven elements found

int min\_uneven\_elems(int const \*const \*matrix, int height, int width);

int main(int argc, char \*\*argv) {

if(argc != 3) {

std::cout << "Usage: " << argv[0] << " test\_file\_A test\_file\_B" << '\n';

return 0;

}

int error;

std::cout << "Файл 1: " << argv[1] << '\n';

int \*\*matrix1, height1, width1;

error = read\_file(argv[1], matrix1, height1, width1);

if(!error)

process\_matrix(matrix1, height1, width1);

std::cout << "Файл 2: " << argv[2] << '\n';

int \*\*matrix2, height2, width2;

error = read\_file(argv[2], matrix2, height2, width2);

if(!error)

process\_matrix(matrix2, height2, width2);

}

int read\_file(const char\* file\_name, int\*\* &matrix, int &height, int &width) {

std::ifstream input\_file(file\_name);

if (!input\_file.is\_open()) {

std::cerr << "Ошибка открытия файла. Убедитесь, что по указанному пути существует файл: "

<< file\_name << '\n';

return -1;

}

if (input\_file.peek() == EOF) {

std::cerr << "Файл " << file\_name << " пуст.\n";

return -2;

}

int e = validate\_and\_read\_matrix(input\_file, matrix, height, width);

if(0 != e) {

std::cerr << "В файле должны через пробел или новую строку находиться количество матриц, а затем для каждой высота, ширина, а затем элементы матрицы слева-направо, затем сверху-вниз. Пример:\n";

std::cerr << "1\n2 3\n1 2 3\n4 5 6\n";

std::cerr << "Исходный файл:\n";

input\_file.clear();

input\_file.seekg(std::ios\_base::beg);

char c;

while(input\_file.get(c))

std::cerr << c;

std::cerr << '\n';

return e;

}

return 0;

}

void process\_matrix(int\*\* matrix, int height, int width) {

std::cout << "Исходная матрица:\n";

print\_matrix(matrix, height, width);

int task1 = mult\_neg\_elems\_above\_diag(matrix, height, width);

if(task1 == 0)

std::cout << "Негативных элементов выше главной диагонали нет!\n";

else

std::cout << "Произведение негативных элементов выше главной диагонали: " << task1 << '\n';

int task2 = min\_uneven\_elems(matrix, height, width);

if(task2 == 0)

std::cout << "Нечётных элементов нет!\n";

else

std::cout << "Минимум среди нечётных элементов: " << task2 << '\n';

}

int validate\_and\_read\_matrix(std::ifstream& in, int\*\* &matrix, int &height, int &width) {

int n;

in >> height >> width;

if(in.fail()) { // not numbers? not enough data? bad either way

std::cerr << "Не удалось получить размерность матрицы.\n";

if(in.eof())

std::cerr << "Недостаточно данных в файле.\n";

return -5;

}

std::cout << "Полученная размерность матрицы: " << height << 'x' << width << '\n';

if(height < 1 || width < 1) {

std::cerr << "Высота и ширина должны быть больше 0!'\n";

return -6;

}

matrix = new int\*[height];

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

matrix[i] = new int[width];

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

in >> n;

if(in.fail()) { // not numbers? not enough data? bad either way

std::cerr << "Ошибка получения данных данных " << i << "-го элемента матрицы.\n";

if(in.eof())

std::cerr << "Недостаточно данных в файле.\n";

return -5;

}

matrix[i][j] = n;

}

}

in >> n;

if(!in.eof()) { // there is more data?

std::cerr << "В файле содержится больше данных, чем нужно для представления заданных матриц.\n";

return -7;

}

return 0;

}

void print\_matrix(const int\*const\* const matrix, int height, int width) {

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

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

std::cout << matrix[i][j] << ' ';

}

std::cout << '\n';

}

}

int mult\_neg\_elems\_above\_diag(int const \*const \*matrix, int height, int width) {

bool neg\_found = false;

int mult = 1;

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

for(int j = i + 1; j < width; j++) {

if(matrix[i][j] < 0) {

neg\_found = true;

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

}

}

}

if(!neg\_found)

return 0;

return mult;

}

int min\_uneven\_elems(int const \*const \*matrix, int height, int width) {

bool uneven\_found = false;

int min = std::numeric\_limits<int>::max();

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

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

if(matrix[i][j] % 2 != 0) {

uneven\_found = true;

min = matrix[i][j] < min ? matrix[i][j] : min;

}

}

}

if(!uneven\_found)

return 0;

return min;

}

}

# Тестирование

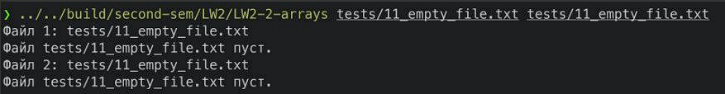
# Область некорректных данных

1. Пустой файл

Название файла: 11\_empty\_file.txt

Ожидаемый вывод: Файл пуст.

Полученный вывод:



Результат совпал с ожидаемым, тест пройден.

1. Ширина и высота матрицы меньше 1

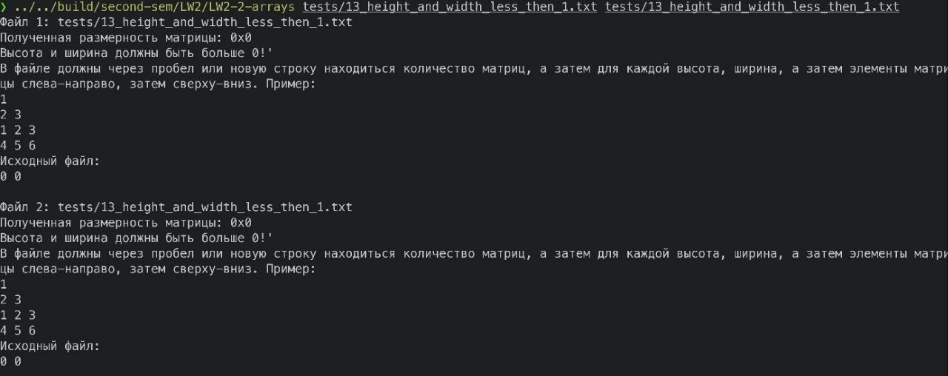
Название файла: 13\_height\_and\_width\_less\_than\_1.txt

Содержимое файла:

0 0

Ожидаемый вывод: Высота и ширина должны быть больше 0!'

Полученный вывод:



Результат совпал с ожидаемым, тест пройден.

1. Нет элементов матрицы

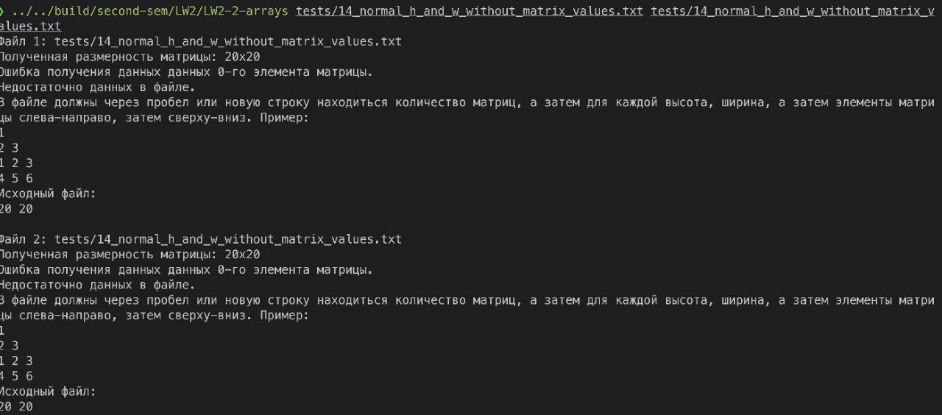
Название файла: 14\_normal\_h\_and\_w\_without\_matrix\_values.txt

Содержимое файла:

20 20

Ожидаемый вывод: недостаточно данных в файле

Полученный вывод:



Результат совпал с ожидаемым, тест пройден.

1. Первая матрица корректная, другой файл не содержит матрицу

Название файлов: 14\_normal\_h\_and\_w\_without\_matrix\_values.txt 21\_unit\_matrix.txt

Содержимое файла a:

20 20

Содержимое файла b:

3 3

1 1 1

1 1 1

1 1 1

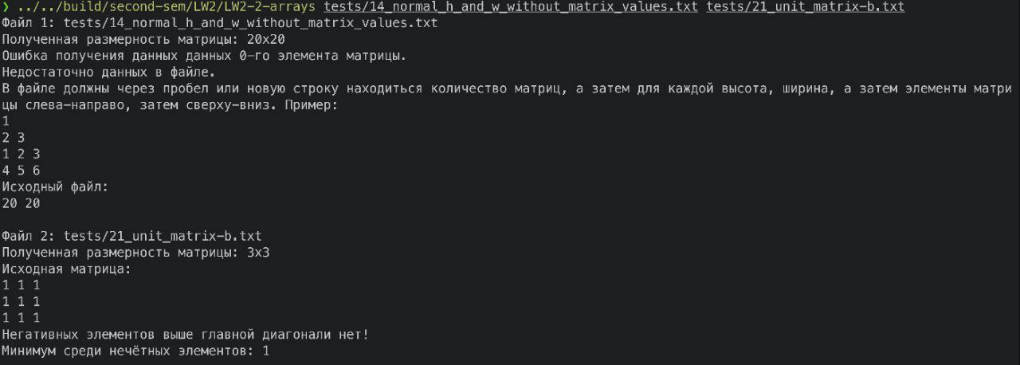
Ожидаемый вывод:

недостаточно данных в файле

Негативных элементов выше главной диагонали нет!

Минимум среди нечётных элементов: 1

Полученный вывод:



Результат совпал с ожидаемым, тест пройден.

1. Размерность матрицы больше, чем указанная

Название файла: 16\_expected\_h\_and\_w\_not\_match\_with\_real.txt

Содержимое файла:

2 2

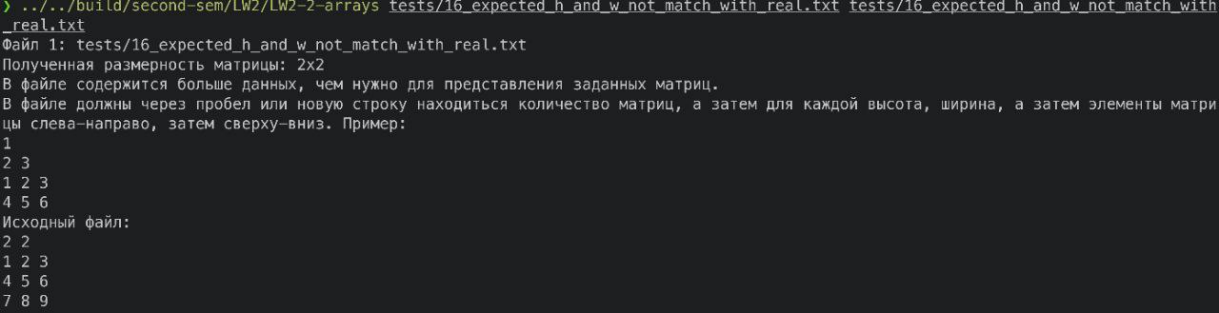
1 2 3

4 5 6

7 8 9

Ожидаемый вывод: В файле содержится больше данных, чем нужно для представления заданных матриц.

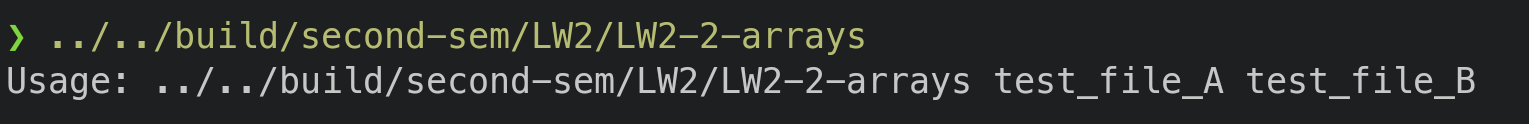
Полученный вывод:



Результат совпал с ожидаемым, тест пройден.

1. Файлы не переданы

Ожидаемый вывод: Usage: path/to/program test\_file\_A test\_file\_B

Полученный вывод: 

Результат совпал с ожидаемым, тест пройден.

# Область корректных данных

1. Нулевая матрица и единичная матрица

Название файла: 21\_zero\_matrix.txt 21\_unit\_matrix.txt

Содержимое файла a:

3 3

0 0 0

0 0 0

0 0 0

Содержимое файла b:

3 3

1 1 1

1 1 1

1 1 1

Ожидаемый вывод:

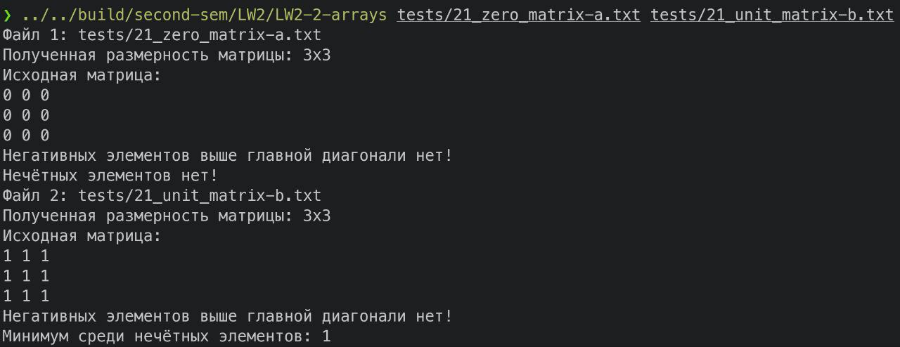
> Негативных элементов выше главной диагонали нет!

> Нечётных элементов нет!

> Негативных элементов выше главной диагонали нет!

> Минимум среди нечётных элементов: 1

Полученный вывод:



Результат совпал с ожидаемым, тест пройден.

1. Без отрицательных

Название файла: 22\_without\_negativ-a.txt 22\_without\_negativ-b.txt

Содержимое файла a:

3 3

1 2 3

4 5 6

7 8 9

Содержимое файла b:

2 2

1 0

0 1

Ожидаемый вывод:

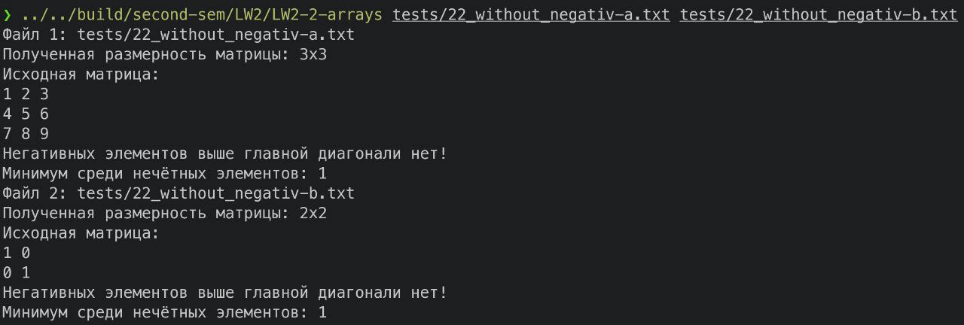
> Негативных элементов выше главной диагонали нет!

> Минимум среди нечётных элементов: 1

> Негативных элементов выше главной диагонали нет!

> Минимум среди нечётных элементов: 1

Полученный вывод:



Результат совпал с ожидаемым, тест пройден.

1. C нечетными и отрицательными

Название файла: 23\_with\_uneven\_and\_negative-a.txt 23\_with\_uneven\_and\_negative-b.txt

Содержимое файла a:

7 7

1 -1 2 -2 3 -3 4

0 0 0 0 0 0 0

-100 2 -2 -5 5 6 -7

12 21 34 -20 77 -99 0

13 13 13 13 13 13 13

52 -52 52 -52 52 -52 52

1 1 0 1 0 -1 0

Содержимое файла b:

9 9

1 -1 1 1 1 1 -1 1 1

2 2 -2 2 2 2 -2 2 2

3 3 3 3 3 3 3 3 3

4 4 4 4 -4 4 4 4 4

5 5 -5 5 5 5 5 -5 5

6 6 6 -6 6 6 6 6 6

7 -7 7 7 7 7 7 7 7

8 8 -8 -8 8 8 -8 8 8

9 9 9 -9 -9 9 9 -9 9

Ожидаемый вывод:

-1\*(-2)\*(-3)\*(-5)\*(-7)\*(-99) = 20790

> Произведение негативных элементов выше главной диагонали: 20790

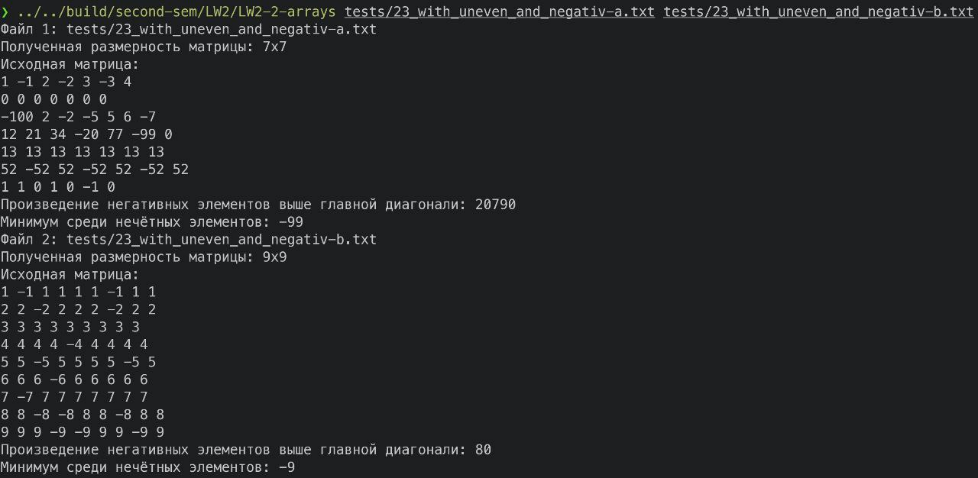
> Минимум среди нечётных элементов: -99

-1\*(-1)\*(-2)\*(-2)\*(-4)\*(-5) = 80

> Произведение негативных элементов выше главной диагонали: 80

> Минимум среди нечётных элементов: -9

Полученный вывод:



Результат совпал с ожидаемым, тест пройден.

1. Только четные

Название файла: 24\_without\_uneven-a.txt 24\_without\_uneven-b.txt

Содержимое файла a:

3 3

2 4 6

8 2 4

6 8 2

Содержимое файла b:

3 3

2 2 -2

2 -2 2

-2 2 2

Ожидаемый вывод:

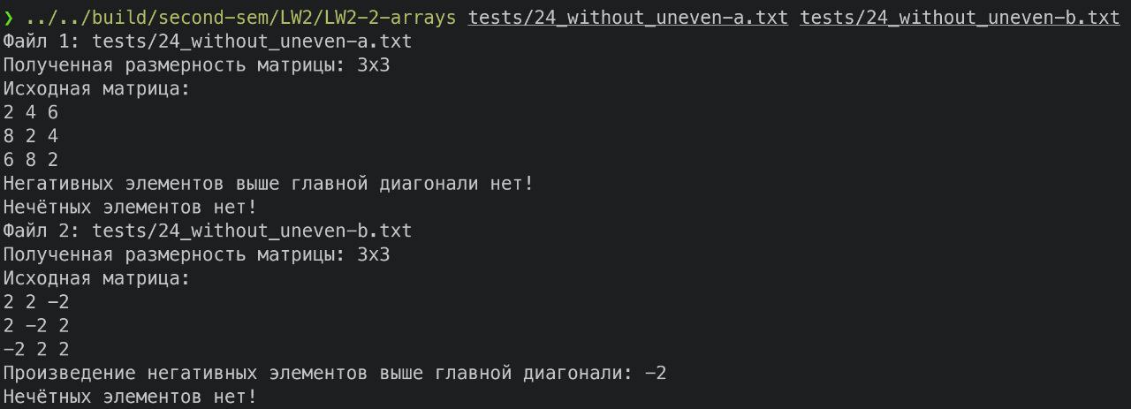
> Негативных элементов выше главной диагонали нет!

> Нечётных элементов нет!

> Произведение негативных элементов выше главной диагонали: -2

> Нечётных элементов нет!

Полученный вывод:



Результат совпал с ожидаемым, тест пройден.

# Вывод

Был проведен ряд тестов, охватывающих область корректных и некорректных входных данных. Программа выполняет поставленную задачу, а тесты ошибки не выявили, в связи с чем было принято решение о прекращении дальнейшей разработки.