**Московский авиационный институт (национальный исследовательский университет)**

Институт №3. «Системы управления, информатика и электроэнергетика». Кафедра №304

Лабораторная работа на тему «Двумерные массивы»

Группа М30-107Б-18

Бригада №1

Выполняли:

Кривонос Александр

Принял:

Чечиков Юрий Борисович

Москва 2018

**Содержание**

1. Задание………………………………………………………………………………….2

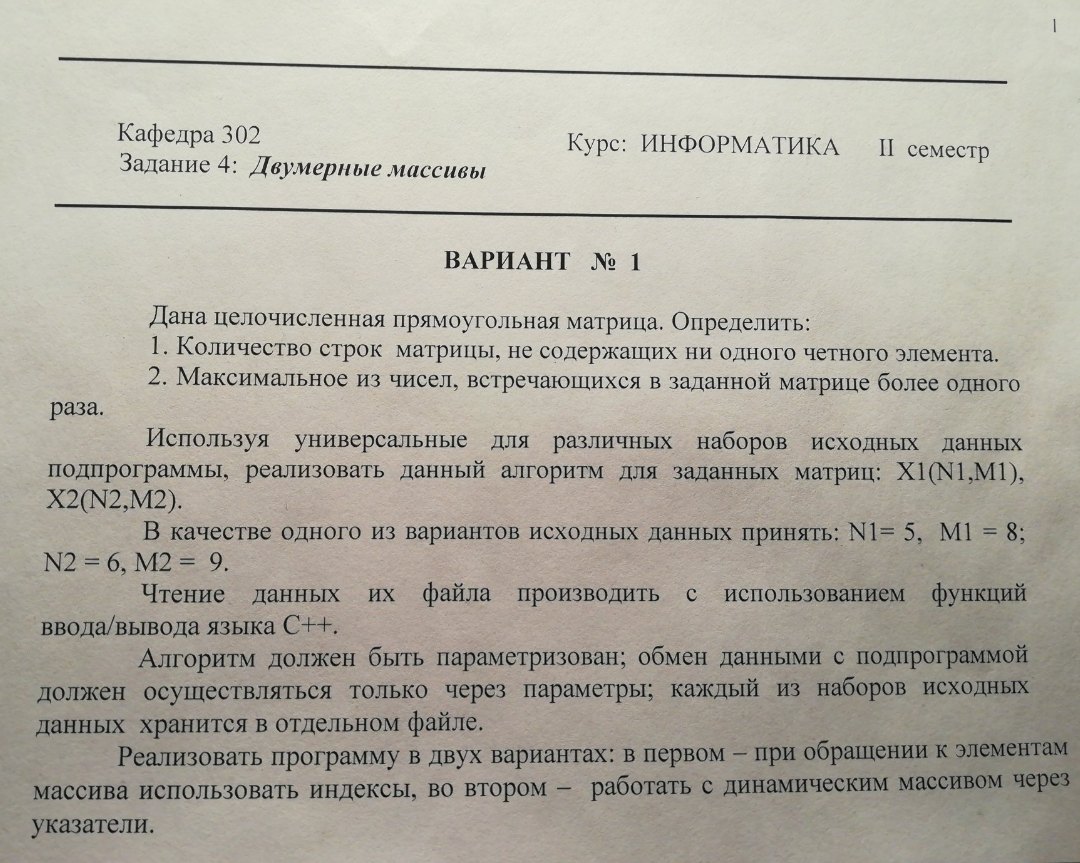
2. Блок-схема алгоритма…………………………………………………………………3

3. Псевдокод……………………………………………………………………………...10

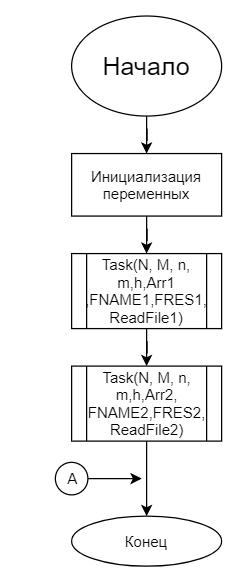
3. Текст программы………………………………………………………………………15

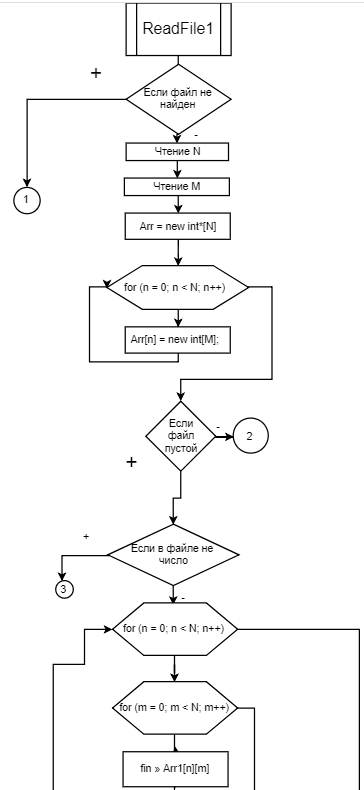
4. Тест (некорректный и корректный)………………………………………………….19

5. Вывод…………………………………………………………………………………..24

**Задание**

**Блок-схема алгоритма**

****

****

**Описание функций:**

1. ReadFile1
   1. Назначение: проверяет корректность введенных данных
   2. Прототип: int ReadFile1 (int n, int m, int &N, int &M, const char\* FNAME,

int \*\*&Arr)

* 1. Обращение: ErrFile1 (…)
  2. Описание параметров:

int xN - число строк,

int xM - число столбцов,

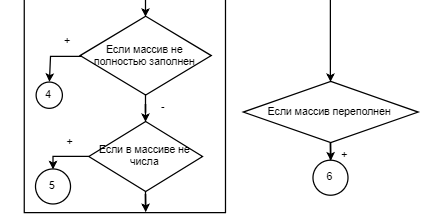
int xn - строковый индекс,

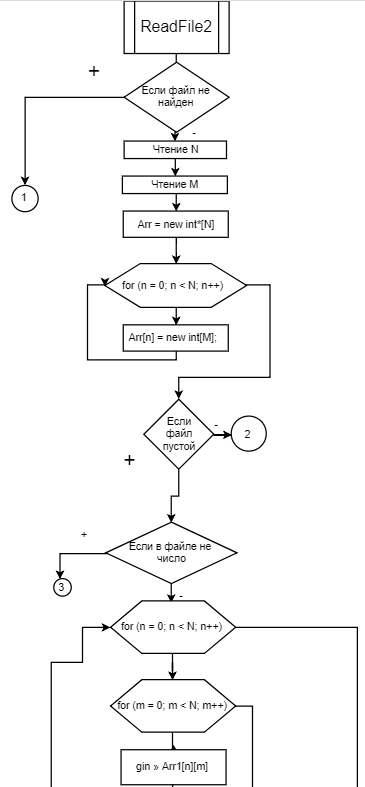
int xm - столбцовый индекс,

const char \*FNAME – файловая переменная,

int Arr1[][] - матрица

* 1. Блок-схема



1. ErrFile2
   1. Назначение: проверяет корректность введенных данных
   2. Прототип: int ErrFile2 (int n, int m, int &N, int &M, const char\* FNAME, int &\*\*Arr)
   3. Обращение: ErrFile2 (…)
   4. Описание параметров:

int xN - число строк,

int xM - число столбцов,

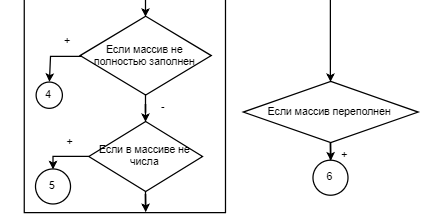
int xn - строковый индекс,

int xm - столбцовый индекс,

const char \*FNAME – файловая переменная,

int Arr2[][] - матрица

* 1. Блок-схема

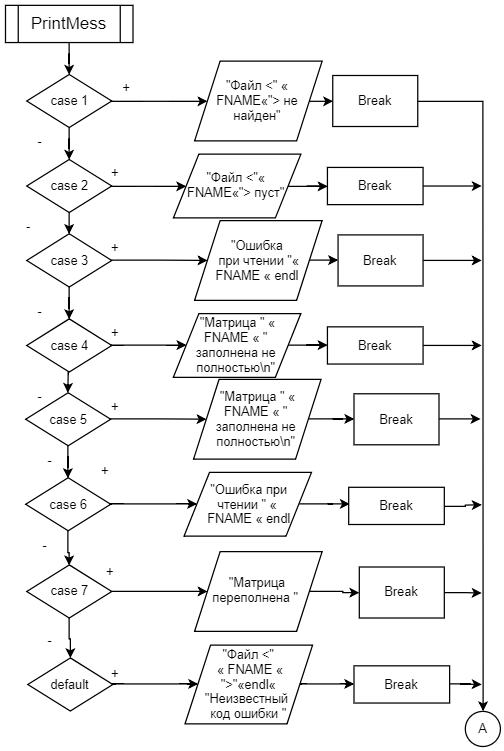


1. PrintMess
   1. Назначение: вывод сообщения об ошибке
   2. Прототип: void PrintMess(int ErrCode, const char\* FNAME)
   3. Обращение: PrintMess (…)
   4. Описание параметров:

int ErrCode – код ошибки

const char \*FNAME – файловая переменная

* 1. Блок-схема

****

1. FT
   1. Назначение: выявляет в строках количество четных элементов
   2. Прототип: void FT (int &n, int &m, int &N, int &M, int &h, int &\*\*Arr)
   3. Обращение: FT (…)
   4. Описание параметров:

int &N - число строк,

int &M - число столбцов,

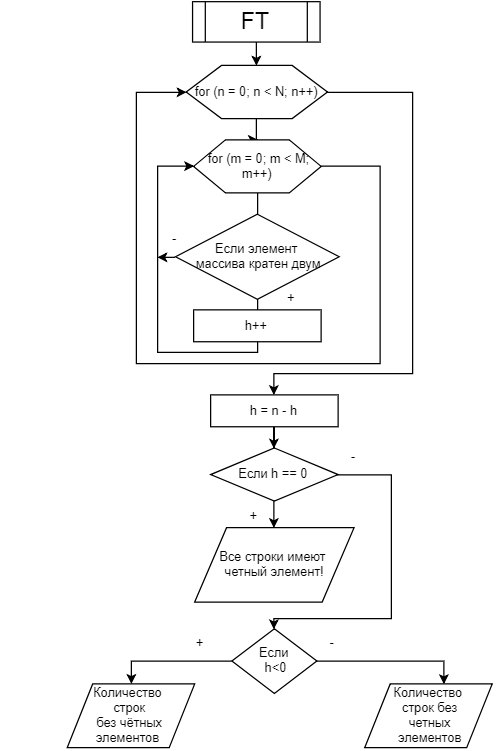
int &n - строковый индекс,

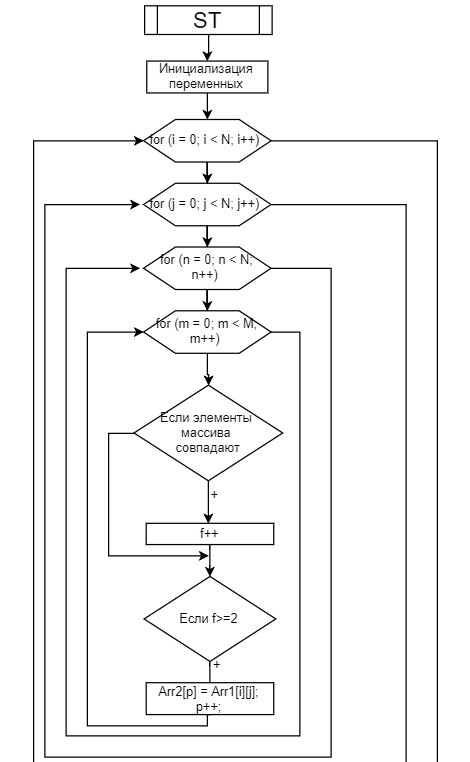
int &m - столбцовый индекс,

int &h - счетчик строк с четными элементами

int Arr[][] - матрица

* 1. Блок-схема

****

1. ST
   1. Назначение: вывод наибольшего из значений, повторяющихся более одного раза
   2. Прототип: void ST
   3. Обращение: ST (…)
   4. Описание параметров:

int xN - число строк,

int xM - число столбцов,

int xn - строковый индекс,

int xm - столбцовый индекс,

int xi - строковый индекс,

int xj - столбцовый индекс,

int xf - счетчик одинаковых элементов,

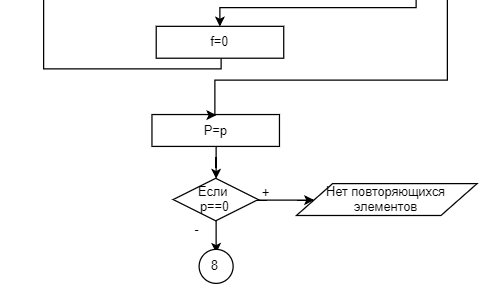
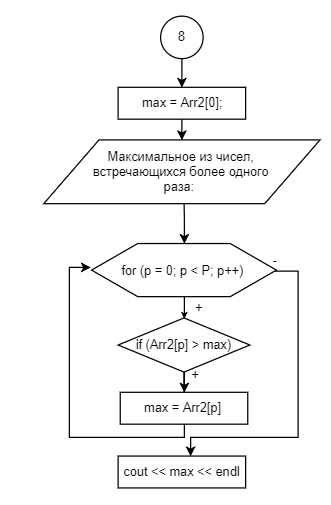
int xp - индекс для строки,

int xP - число элементов в строке,

int xmax - максимальное из чисел,

const char \*XFNAME – файловая переменная

* 1. Блок-схема

****

1. Task
   1. Назначение: вызывает функции для выполнения заданий
   2. Прототип: void Task(int &N, int &M, int &n, int &m, int &h, int &\*\*Arr,

const char \*FNAME, int ReadFile(int n, int m, int &N, int &M,

const char\* FNAME, int &\*\*Arr)))

* 1. Обращение: Task (…)
  2. Описание параметров:

int &N - число строк,

int &M - число столбцов,

int &n - строковый индекс,

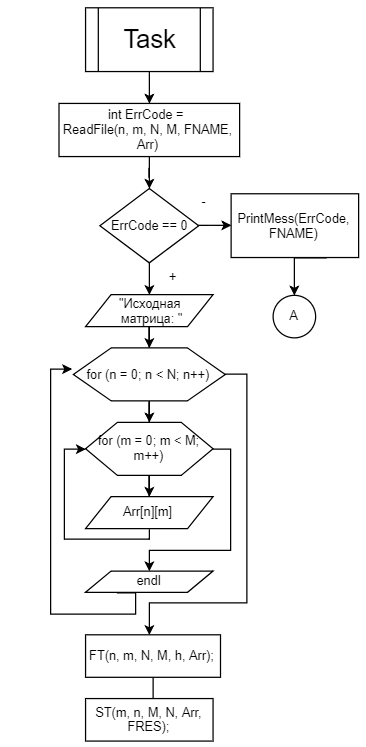
int &m - столбцовый индекс,

int &h - счетчик строк с четными элементами

int Arr[][] – матрица

const char \*FNAME – файловая переменная

int ReadFile (…) – функция для чтение матрицы из файла

* 1. Блок-схема

**Псевдокод**

*Основная функция*

НЧ

Инициализация переменных

Функция Task(N, M, n, m,h,Arr1,FNAME1,FRES1, ReadFile1)

Функция Task(N, M, n, m,h,Arr2,FNAME2,FRES2, ReadFile2)

КЦ

Если произошла ошибка

То

Конец

*Функция ReadFile1*

Если файл не найден

То

Произошла ошибка

Иначе

Чтение N

Чтение M

Выделение памяти для нового массива

От n=0 до n<N; n++

Arr[n] = new int[M];

Если файл пустой

То

Произошла ошибка

Иначе

Если в файле не число

То

Произошла ошибка

От n = 0 до n < N ; n++

От m=0 до m<N ; m++

Чтение массива

Если массив пустой

То

Произошла ошибка

Иначе

Если в массиве не числа

То

Произошла ошибка

Иначе

Если массив переполнен

То

Произошла ошибка

*Функция FT*

От n=0 до n<N; n++

От m=0 до m<N; n++

Если элемент массива кратен двум

То

h++

h=n-h

Если h==0

Вывод на экран :”Все строки имеют чётный элемент”

Иначе

Если h<0

Вывод на экран: "Количество строк без чётных элементов"<<n

Иначе

"Количество строк без четных элементов"<<m

*Функция ST*

Инициализация переменных

От i=0 до i<N; i++

От j=0 до j<N; j++

От n=0 до n<N; n++

От m=0 до m<N; m++

Если элементы массива совпадают

То

f++

Иначе

Если f>=2

То

Arr2[p] = Arr1[i][j]

p++

f=0

P=p

Если p==0

То

Вывод на экран : «Нет повторяющихся элементов»

Иначе

От p=0 до p<P; p++

Запись Arr[p] в файл

max=Arr2[0]

От p=0 до p<P; p++

Если Arr2[p] > max

max = Arr2[p]

*Функция Task*

int ErrCode = ReadFile(n, m, N, M, FNAME, Arr)

Если ErrCode == 0

То

PrintMess(ErrCode, FNAME)

Иначе

От n=0 до n<N; n++

От m=0 до m<M; m++

Вывод на экран Arr[n][m]

FT(n, m, N, M, h, Arr)

ST(m, n, M, N, Arr, FRES);

Ф*ункция PrintMess*

Если case 1

То

Вывод на экран : "Файл <" « FNAME«"> не найден"

КЦ

Иначе

Если case 2

То

Вывод на экран : "Файл <"« FNAME«"> пуст"

КЦ

Иначе

Если case 3

То

Вывод на экран : "Ошибка при чтении "« FNAME

КЦ

Иначе

Если case 4

То

Вывод на экран : "Матрица " « FNAME « " заполнена не полностью\n"

КЦ

Иначе

Если case 5

То

Вывод на экран : "Матрица " « FNAME « " заполнена не полностью\n"

КЦ

Иначе

Если case 6

То

Вывод на экран : "Ошибка при чтении " « FNAME « endl

КЦ

Иначе

Если case 7

То

Вывод на экран : "Матрица переполнена "

КЦ

Иначе

Если default

То

Вывод на экран : "Файл <" « FNAME « ">"«endl«"Неизвестный код ошибки "

КЦ

**Текст программы**

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

\* Информатика и Вычислительная техника \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* Language: cpp, MSVS 2010 and above \*

\* Programmers: M3O-107Б-18 \*

\* Кривонос Александр Александрович \*

\* Created: 20.04.2019 Last revision: 26.04.2019 \*

\* Comment: Двумерные массивы \*

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

#include <iostream>

#include <fstream>

#include <string.h>

using namespace std;

const char \*FNAME1 = "Array1.txt"; //имя файла с исходными данными

const char \*FNAME2 = "Array2.txt"; //имя файла с исходными данными

ifstream fin(FNAME1);

ifstream gin(FNAME2);

int ReadFile1(int n, int m, int &N, int &M, const char\* FNAME, int \*\*&Arr)

{

/\*Если файл не найден\*/

if (!fin)

return 1; //выход по ошибке

fin >> N; //чтение числа строк

fin >> M; //чтение числа столбцов

Arr = new int\*[N]; //выделение памяти для матрицы

for (n = 0; n < N; n++)

{

Arr[n] = new int[M];

}

/\*Если файл пуст\*/

if (fin.eof())

return 2; //выход по ошибке

/\*Если в файле не число\*/

if (fin.fail())

return 3; //выход по ошибке

for (n = 0; n < N; n++)

{

for (m = 0; m < M; m++)

{

fin >> \*(\*(Arr + n) + m);

if ((fin.eof()) && (n < N - 1)) //если не полностью заполнен

return 4; //выход по ошибке

if ((fin.eof()) && (m < M - 1)) //если не полностью заполнен

return 5; //выход по ошибке

if (fin.fail()) //если в матрице не числа

return 6; //выход по ошибке

}//end for m

} //end for n

if (!fin.eof())

return 7; //выход по ошибке

return 0;

}

int ReadFile2(int n, int m, int &N, int &M, const char\* FNAME, int \*\*&Arr)

{

/\*Если файл не найден\*/

if (!gin)

return 1; //выход по ошибке

gin >> N; //чтение числа строк

gin >> M; //чтение числа столбцов

Arr = new int\*[N]; //выделение памяти для матрицы

for (n = 0; n < N; n++)

{

Arr[n] = new int[M];

}

/\*Если файл пуст\*/

if (gin.eof())

return 2; //выход по ошибке

/\*Если в файле не число\*/

if (gin.fail())

return 3; //выход по ошибке

for (n = 0; n < N; n++)

{

for (m = 0; m < M; m++)

{

gin >> \*(\*(Arr + n) + m);

if ((gin.eof()) && (n < N - 1)) //если не полностью заполнен

return 4; //выход по ошибке

if ((gin.eof()) && (m < M - 1)) //если не полностью заполнен

return 5; //выход по ошибке

if (gin.fail()) //если в матрице не числа

return 6; //выход по ошибке

}//end for m

} //end for n

if (!gin.eof())

return 7; //выход по ошибке

return 0;

}

void PrintMess(int ErrCode, const char\* FNAME)

{

switch (ErrCode)

{

case 1: cout << "Файл <" << FNAME << "> не найден" << endl; break;

case 2: cout << "Файл <" << FNAME << "> пуст" << endl; break;

case 3: cout << "Неверно введена размерность в файле " << FNAME << endl; break;

case 4: cout << "Матрица " << FNAME << " заполнена не полностью\n"; break;

case 5: cout << "Матрица " << FNAME << " заполнена не полностью\n"; break;

case 6: cout << "Неверно введены значения элементов в файле " << FNAME << endl; break;

case 7: cout << "Матрица " << FNAME << " переполнена " << endl; break;

default: cout << "Файл <" << FNAME << ">" << endl << "Неизвестный код ошибки " << ErrCode

<< endl; break;

}

}

void FT(int &n, int &m, int &N, int &M, int &h, int \*\*Arr)

{

for (n = 0; n < N; n++)

{

for (m = 0; m < M; m++)

{

if (\*(\*(Arr+n)+m) % 2 == 0)

{

h++;

break;

}

}

}

h = n - h;

if (h == 0)

{

cout << "Все строки имеют четный элемент!" << endl;

}

if (h < 0)

{

cout << "Количество строк без четных элементов: " << n << endl;

}

else

cout << "Количество строк без четных элементов: " << h << endl;

}

void ST(int &m, int &n, int &M, int &N, int \*\*Arr)

{

int i = 0; // строковый индекс

int j = 0; // столбцовый индекс

int f = 0; //счетчик одинаковых элементов

int h = 0; //счетчик строк с четными элементами

int p = 0; //индекс для строки

int P = 0; //число элементов в строке для матрицы

int \*Arr3 = new int[P];//строка из элементов, встречающихся более одного раза

int max = 0; //максимальное из чисел

for (i = 0; i < N; i++)

{

for (j = 0; j < N; j++)

{

for (n = 0; n < N; n++)

{

for (m = 0; m < M; m++)

{

if (\*(\*(Arr + i) + j) == \*(\*(Arr + n) + m))

{

f++;

if (f >= 2)

{

\*(Arr3+p) = \*(\*(Arr + i) + j);

p++;

}//if f

}//if ==

}//for m

}//for n

f = 0;

}//for j

}//for i

P = p;

if (p == 0)

cout << "Нет повторяющихся элементов.\n";

else

{

max = \*(Arr3);

cout << "Максимальное из чисел, встречающихся более одного раза: ";

for (p = 0; p < P; p++)

{

if (\*(Arr3+p) > max)

max = \*(Arr3 + p);

}

cout << max << endl;

}//else

delete[]Arr3; //удаление строки

}

void Task(int &N, int &M, int &n, int &m, int &h, int \*\*&Arr, const char \*FNAME, int ReadFile(int n, int m, int &N, int &M, const char\* FNAME, int \*\*&Arr))

{

int ErrCode = ReadFile(n, m, N, M, FNAME, Arr);

if (ErrCode == 0)

{

cout << "Исходная матрица:\n";

for (n = 0; n < N; n++)

{

for (m = 0; m < M; m++)

{

cout << \*(\*(Arr + n) + m) << " ";

}//for m

cout << endl;

}//for n

FT(n, m, N, M, h, Arr);

ST(m, n, M, N, Arr);

}//if

else

PrintMess(ErrCode, FNAME);

}

int main()

{

setlocale(LC\_ALL, "Russian");

int n = 0; //строковый индекс

int m = 0; //столбцовый индекс

int p = 0;

int N; //число строк

int M; //число столбцов

int h = 0; //счетчик строк с четными элементами

int \*\*Arr1; //объявление матрицы

int \*\*Arr2; //объявление матрицы

Task(N, M, n, m, h, Arr1, FNAME1, ReadFile1);

//удаление первой матрицы

for (n = 0; n < N; n++)

{

delete[]Arr1[n];

}

delete[]Arr1;

Task(N, M, n, m, h, Arr2, FNAME2, ReadFile2);

//удаление второй матрицы

for (n = 0; n < N; n++)

{

delete[]Arr2[n];

}

delete[]Arr2;

fin.close();

gin.close();

system("pause");

return 0;

} //end main

**Тесты (корректные и некорректные)**

**1.Некорректные**

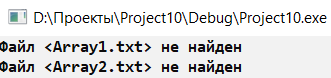
**Тесты (корректные и некорректные)**

**1.Некорректные**

1. Цель: проверить работу программы без файлов с матрицей

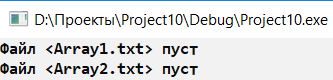
Ожидаемый результат: Файл Array1.txt не найден.

Файл Array2.txt не найден.



1. Цель: проверить работу программы с пустыми файлами.

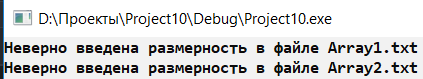
Ожидаемый результат: Файл Array1.txt пустой.

 Файл Array2.txt пустой.

1. Цель: проверить работу программы с некорректно введенными значениями размерности.

Ожидаемый результат: Неверно введена размерность в файле Array1.txt

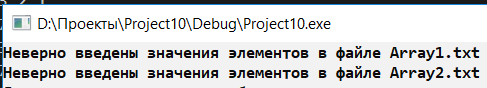
Неверно введена размерность в файле Array2.txt



1. Цель: проверить работу программы с некорректно введенными значениями элементов.

Ожидаемый результат: Неверно введены значения элементов в файле Array1.txt

Неверно введены значения элементов в файле Array2.txt

****

1. Цель: проверить работу программы с матрицами, заполненными неполностью.

Исходные данные:

Для файла Array1.txt:

5 8

1 2 3 4 5 6 7 8

9 8 7 6 5 4 3 2

5 4 3 2 1 6 7 8

6 7 8 9 5 4 3 2

1 2 3 4 5 6 7

Для файла Array2.txt:

6 9

1 2 3 4 5 6 7 8 9

9 8 7 6 5 4 3 2 1

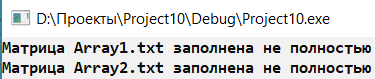
5 4 3 2 1 6 7 8 9

6 7 8 9 5 4 3 2 1

1 2 3 4 5 6 7 8 9

9 8 7 6 5 4 3 2

Ожидаемый результат: Матрица Array1.txt заполнена не полностью

****Матрица Array2.txt заполнена не полностью

1. Цель: проверить работу программы с переполненными матрицами.

Исходные данные:

Для файла Array1.txt:

5 8

1 2 3 4 5 6 7 8

9 8 7 6 5 4 3 2

5 4 3 2 1 6 7 8

6 7 8 9 5 4 3 2

1 2 3 4 5 6 7 8 9

Для файла Array2.txt:

6 9

1 2 3 4 5 6 7 8 9

9 8 7 6 5 4 3 2 1

5 4 3 2 1 6 7 8 9

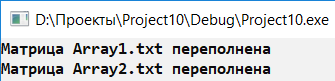
6 7 8 9 5 4 3 2 1

1 2 3 4 5 6 7 8 9

9 8 7 6 5 4 3 2 7 3

Ожидаемый результат: Матрица Array1.txt переполнена

Матрица Array2.txt переполнена

****

**2. Корректные**

1. Цель: проверить работу программы с корректно введенными данными (есть повторяющиеся числа и во всех строках присутствуют четные числа)

Исходные данные:

Для файла Array1.txt:

5 8

1 2 3 4 5 6 7 8

9 8 7 6 5 4 3 2

5 4 3 2 1 6 7 8

6 7 8 9 5 4 3 2

1 2 3 4 5 6 7 8

Для файла Array2.txt:

6 9

1 2 3 4 5 6 7 8 9

9 8 7 6 5 4 3 2 1

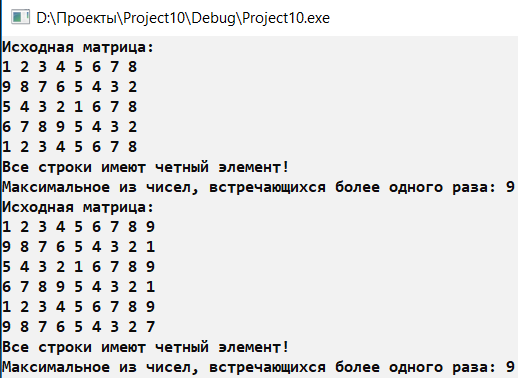
5 4 3 2 1 6 7 8 9

6 7 8 9 5 4 3 2 1

1 2 3 4 5 6 7 8 9

9 8 7 6 5 4 3 2 7

Ожидаемый результат: «Все строки имеют четный элемент! Максимальное из чисел, встречающихся более одного раза: 9»



1. Цель: проверить работу программы с корректно введенными данными (нет повторяющихся чисел и в одной строке отсутствуют четные числа)

Исходные данные:

Для файла Array5.txt:

3 3

7 8 99

2 44 3

55 37 9

Для файла Array6.txt:

3 4

1 3 5 7

9 11 13 15

17 19 21 4

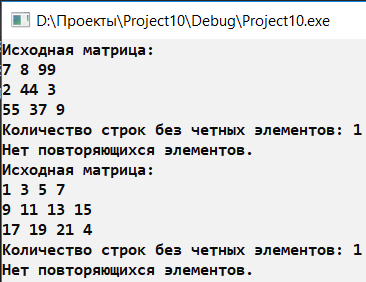
Ожидаемый результат:

Для файла Array5.txt: «Количество строк без четных элементов: 1

Нет повторяющихся элементов.»

Для файла Array6.txt: «Количество строк без четных элементов: 1

Нет повторяющихся элементов.»



1. Цель: проверить работу программы с корректно введенными данными (Для Array3.txt: нет повторяющихся чисел и во всех строках отсутствуют четные числа; Для Array4.txt: есть повторяющиеся числа и во всех строках отсутствуют четные числа)

Исходные данные:

Для Array3.txt:

5 4

1 3 5 7

9 11 13 15

17 19 21 23

25 27 29 31

33 35 37 39

Для Array4.txt:

4 4

1 1 1 1

1 1 1 1

1 1 1 1

1 1 1 1

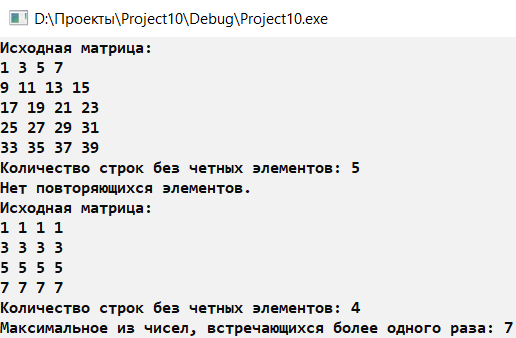
Ожидаемый результат:

Для файла Array3.txt: «Количество строк без четных элементов: 5

Нет повторяющихся элементов.»

Для файла Array4.txt: «Количество строк без четных элементов: 4

Максимальное из чисел, встречающихся более одного раза: 7.»



**Вывод**

Разработка программы завершена на том основании, что

1) Полученные результаты совпали с ожидаемыми.

2) Считаем, что набор тестов полный.