**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САПР.**

отчет

**по лабораторной работе №5**

**по дисциплине «Программирование»**

Тема: «Использованием двумерных массивов и функций».

|  |  |  |
| --- | --- | --- |
| Студентка гр. 1309 |  | Ищенко Д.О. |
| Преподаватель |  | Калмычков В.А. |

Санкт-Петербург

2021

Оглавление

[Формулировка задания. 3](#_Toc88314115)

[Контрольный пример. 3](#_Toc88314116)

[Математическая постановка задачи. 3](#_Toc88314117)

[Способ внутреннего хранения данных. 3](#_Toc88314118)

[Способ реализации ввода-вывода. 3](#_Toc88314119)

[Блок-схема. 4](#_Toc88314120)

[Текст программы. 4](#_Toc88314121)

[Результаты работы программы. 4](#_Toc88314122)

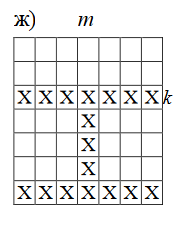
[Вывод. 4](#_Toc88314123)

# Цель работы

Освоение программирования обработки двумерных массивов с применением циклов, с организацией файлового ввода/вывода, с разбиением программы на функции.

# Формулировка задания.

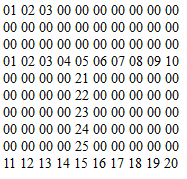
Определить наибольший элемент заштрихованной области таблицы *А* размера *N* x *N.*



# Контрольный пример.

N = 10, M = 5, K = 4

Если таблица имеет вид:



То наибольшим элементом является число 25, находящееся в 9 строке 5 столбце.

# Математическая постановка задачи.

**Дано:**

**Найти:**

**Способ решения:**

# Способ внутреннего хранения данных.

Переменные функции main.

|  |  |  |  |
| --- | --- | --- | --- |
| Имя переменной | Тип переменной | Ограничения на размер данных | Назначение |
| F\_in | double | 1,7E +/- 308 (15 знаков) |  |
| p | double |  |
| v1 | double |  |
| p | double |  |
| v1 | double |  |

# Способ реализации ввода-вывода.

Для ввода-вывода при реализации задания был задействован класс fstream из одноименной библиотеки для работы с файлами.

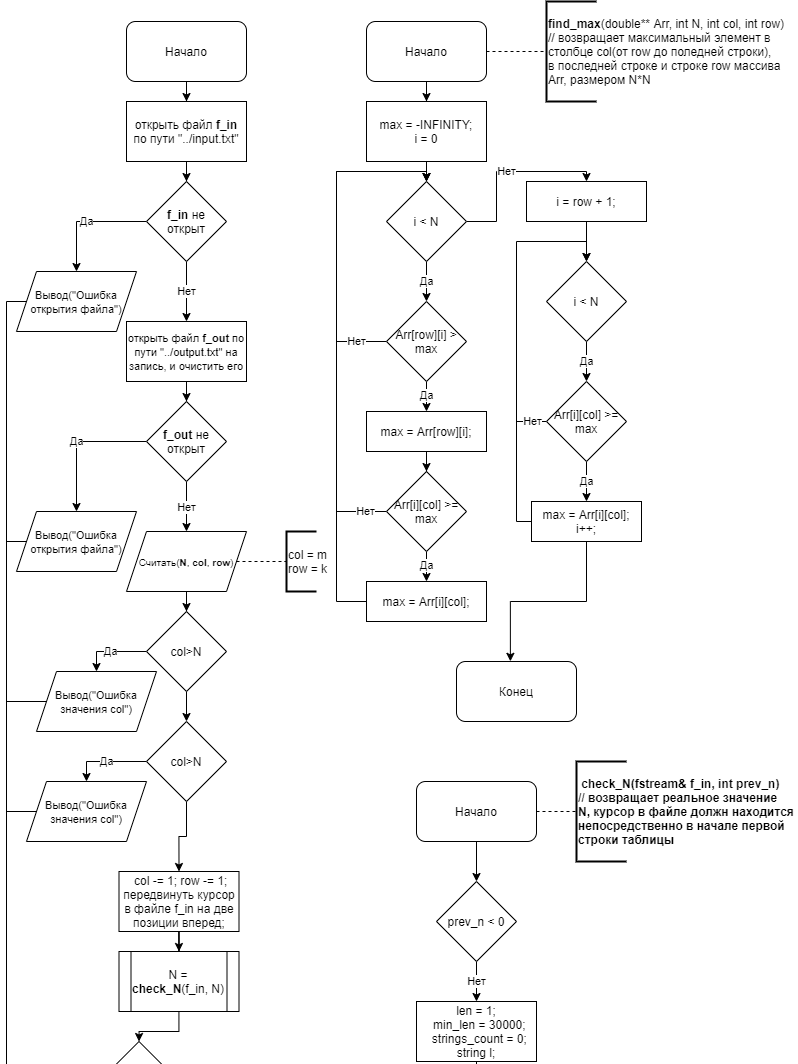
Класс fstream позволяет создать объект, с помощью которого возможно записывать и считывать данные из файла. Конструктору класса первым параметром передается расположение и название файла, вторым маркер, означающий тип работы с файлом (чтение/запись).

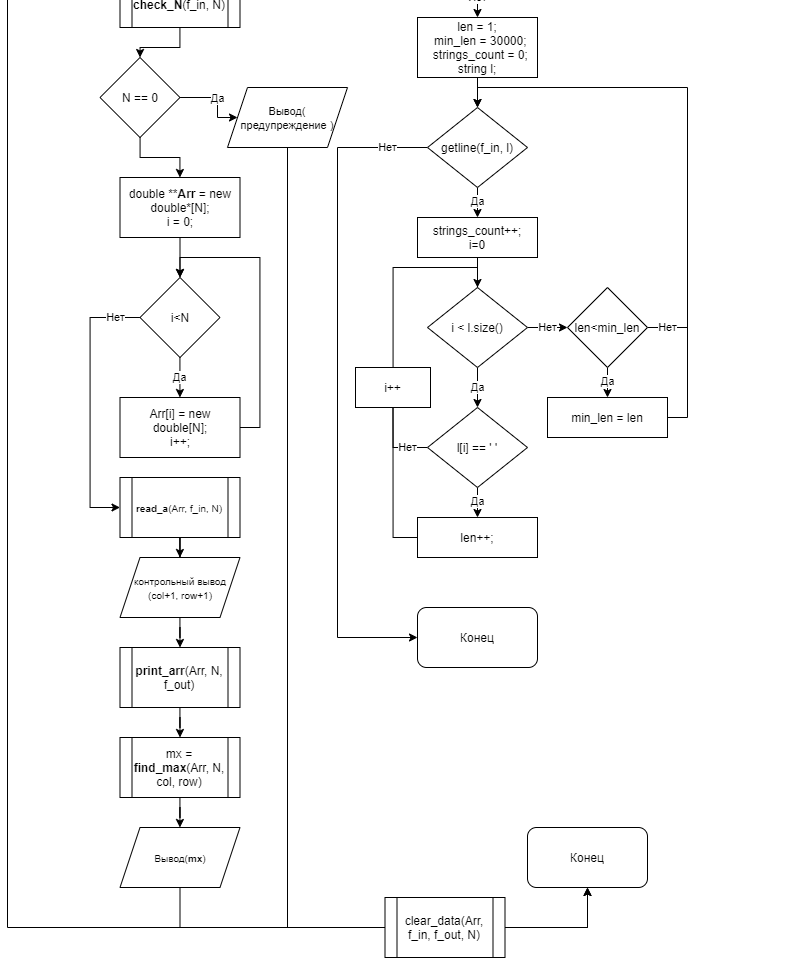
Для файлов записи (f\_out) были задействованы флаги trunc и out. Trunс очищает содержимое файла при его открытии, out указывает на то, что файл открыт только для записи.

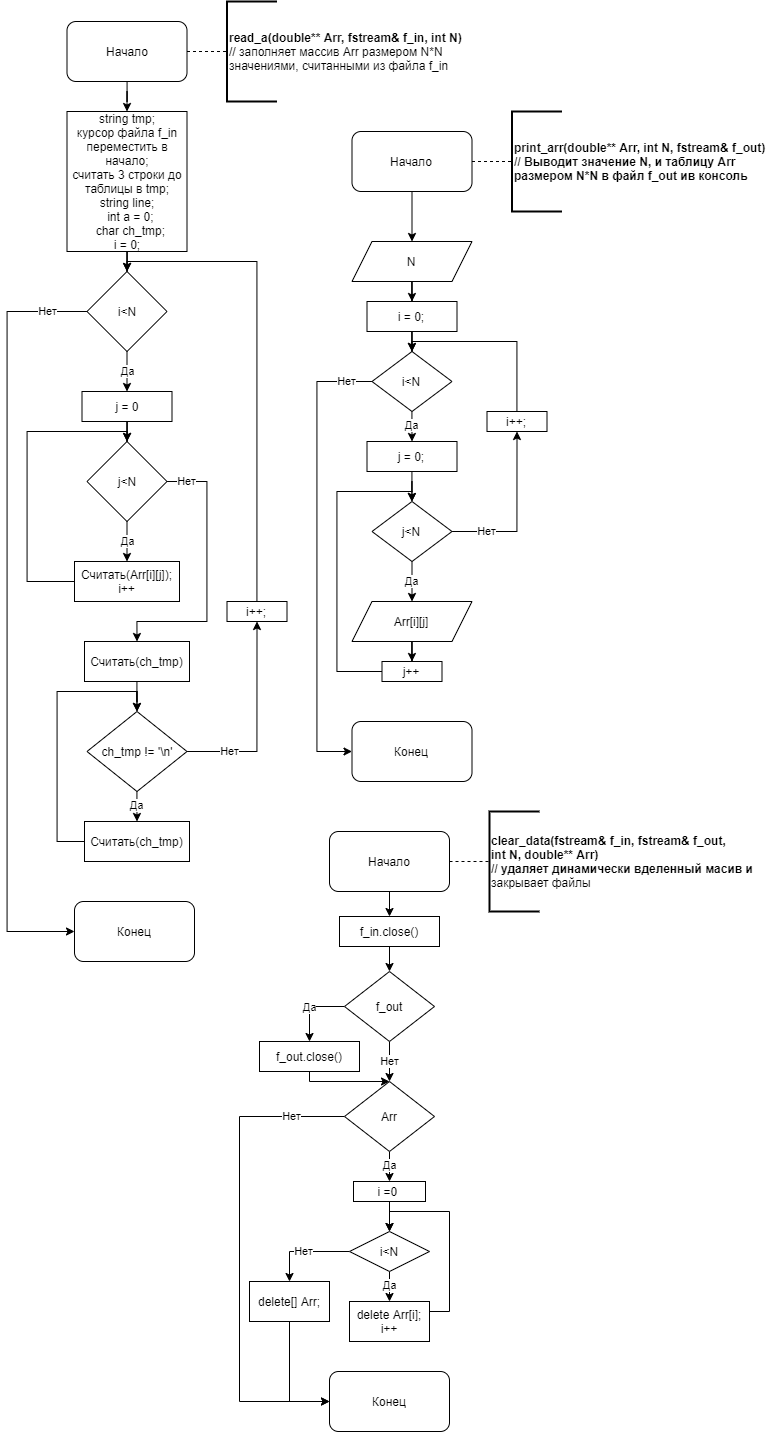
# Функции в программе.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя функции | Назначение | Параметры | | | Внешние эффекты |
| входные | выходные | модифицируемые |
| main | управляет выполнением программы | - | - | - | Создается (изменяется) файл output.txt |
| find\_max | возвращает максимальный элемент в  столбце сol(от row до поледней строки),  в последней строке и строке row массива  Arr, размером N\*N | double\*\* Arr, int N, int col, int row | double max | - |  |
| check\_N | возвращает реальное значение  N | fstream& f\_in, int prev\_n | int N | f\_in | Курсор в f\_in смещается в конец файла |
| read\_a | Заполняет массив Arr размером N\*N  значениями, считанными из файла f\_in | double\*\* Arr, fstream& f\_in, int N | - | Arr, f\_in | Курсор в f\_in смещается в конец файла |
| print\_arr | Выводит значение N, и таблицу Arr  размером N\*N в файл f\_out ив консоль | double\*\* Arr, int N, fstream& f\_out | - | f\_out | Изменяется содержимое файла f\_out |
| clear\_data | удаляет динамически выделенный массив и  закрывает файлы | fstream& f\_in, fstream& f\_out,  int N, double\*\* Arr | - | Arr, f\_in, f\_out | Закрывает переданные файлы, удаляет массив Arr |

# Блок-схема.







# Текст программы.

#include <iostream>

#include <fstream>

#include <string>

#include <cmath>

#include <stdio.h>

#include <iomanip>

#include <limits>

using namespace std;

int check\_N(fstream& f\_in, int prev\_n);

void read\_a(double\*\* Arr, fstream& f\_in, int N);

void print\_arr(double\*\* Arr, int N, fstream& f\_out);

double find\_max(double\*\* Arr, int N, int col, int row);

void clear\_data(fstream& f\_in, fstream& f\_out = numeric\_limits<fstream&>::quiet\_NaN(), int N = numeric\_limits<int>::quiet\_NaN(), double\*\* Arr = numeric\_limits<double\*\*>::quiet\_NaN());

int main()

{

fstream f\_in("../input.txt");

if (!f\_in.is\_open())

{

cout << "input file open error.";

return -1;

}

fstream f\_out("../output.txt", fstream::trunc | fstream::out);

if (!f\_out.is\_open()) {

cout << "output file open error.";

return -1;

}

int N, col, row;

f\_in >> N >> col >> row;

if (col > N) {

cout << "column value more than N";

clear\_data(f\_in, f\_out);

return -1;

}

else if (row > N) {

cout << "row value more than N";

clear\_data(f\_in, f\_out);

return -1;

}

col -= 1; row -= 1;

f\_in.seekg(2, ios::cur);

N = check\_N(f\_in, N);

if (col+1 > N) {

cout << "New N value = "<<N<<". column value now more than N";

clear\_data(f\_in, f\_out);

return -1;

}

else if (row+1 > N) {

cout << "New N value = " << N << ". row value now more than N";

clear\_data(f\_in, f\_out);

return -1;

}

if (N == 0) {

cout << "N is null";

f\_out << "N is null";

clear\_data(f\_in, f\_out);

return 0;

}

//init arr

double \*\*Arr = new double\*[N];

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

{

Arr[i] = new double[N];

}

read\_a(Arr, f\_in, N);

f\_out << "m = " << col + 1 << "\nk = " << row+1 << endl;

print\_arr(Arr, N, f\_out);

double mx = find\_max(Arr, N, col, row);

cout << mx;

f\_out << "max item = " << mx;

clear\_data(f\_in, f\_out, N, Arr);

}

double find\_max(double\*\* Arr, int N, int col, int row) {

double max = -INFINITY;

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

if (Arr[row][i] > max)

max = Arr[row][i];

if (Arr[N - 1][i] > max)

max = Arr[N - 1][i];

}

for (int i = row + 1; i < N; i++) {

if (Arr[i][col] >= max)

max = Arr[i][col];

}

return max;

}

int check\_N(fstream& f\_in, int prev\_n) {

// курсор в файле должн находится непосредственно в начале первой строки таблицы

// строки таблицы не могут оканчиваться пробелом

if (prev\_n < 0)

return 0;

int min\_len = 30000;

int strings\_count = 0;

string l;

while (getline(f\_in, l)) {

int len = 1;

strings\_count++;

for (int i = 0; i < l.size(); i++) {

if (l[i] == ' ')

len++;

}

if (len < min\_len)

min\_len = len;

}

int mn = min(min(min\_len, strings\_count), prev\_n);

return mn;

}

void read\_a(double\*\* Arr, fstream& f\_in, int N) {

string tmp;

f\_in.clear();

f\_in.seekg(ios::beg);

for (int i = 0; i < 3; i++)

getline(f\_in, tmp);

string line;

int a = 0;

char ch\_tmp;

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

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

f\_in >> setiosflags(ios::skipws) >> Arr[i][j];

}

f\_in >> resetiosflags(ios::skipws) >> ch\_tmp;

while (ch\_tmp != '\n')

f\_in >> ch\_tmp;

}

}

void print\_arr(double\*\* Arr, int N, fstream& f\_out) {

f\_out << "N = " << N << endl;

f\_out << "table = \n";

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

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

f\_out << Arr[i][j] << " ";

cout << Arr[i][j] << " ";

}

f\_out << endl;

cout << endl;

}

}

void clear\_data(fstream& f\_in, fstream& f\_out, int N, double\*\* Arr) {

f\_in.close();

if (f\_out) {

f\_out.close();

}

if (Arr) {

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

delete Arr[i];

}

delete[] Arr;

}

# }

# Результаты работы программы.

m = 5

k = 4

N = 10

table =

0 2 3 0 0 0 0 0 0 0

20 0 0 0 0 0 0 0 0 0

0 0 0 0 0 0 0 0 0 0

1 2 3 4 5 6 7 8 9 11

0 0 0 0 21 0 0 0 0 0

0 0 0 0 22.4 0 0 0 0 0

0 0 0 0 23 0 0 0 0 0

0 0 0 0 24 0 0 0 0 0

0 0 0 0 25 0 0 0 0 0

11 12 13 14 15 16 17 18 19 20

max item = 25

# Вывод.

В ходе выполнения лабораторной работы была написана программа, решающая поставленную задачу. Во время её разработки были получены практические навыки работы с различными конструкциями языка с++, такими как двумерные массивы, функции, разные типы передачи аргументов в функцию, а так же закреплен навык работы с файлами.