**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САПР**

отчЁт

**по лабораторной работе №5**

**по дисциплине «Программирование»**

Тема: Обработка двумерного массива с использованием функций

|  |  |  |  |
| --- | --- | --- | --- |
| Студент гр. 4352 |  |  | И.В. Чучалин |
| Преподаватель |  |  | В.А. Калмычков |

# Оглавление

[Оглавление 2](#_Toc181895887)

[1 Задание 3](#_Toc181895888)

[2 Неясности, предложения их решения и уточнения 3](#_Toc181895889)

[3 Математическая постановка 3](#_Toc181895890)

[3.1 Данные 3](#_Toc181895891)

[3.2 Заштрихованная область 3](#_Toc181895892)

[3.3 Способ решения 3](#_Toc181895893)

[4 Контрольные примеры 4](#_Toc181895894)

[5 Ограничения, обусловленные вычислительным устройством 4](#_Toc181895895)

[6 Организация интерфейса пользователя 4](#_Toc181895896)

[7 Формат файлов 5](#_Toc181895897)

[7.1 Организация входного файла 5](#_Toc181895898)

[7.2 Организация выходного файла 6](#_Toc181895899)

[8 Реализация ввода/вывода 6](#_Toc181895900)

[9 Внутреннее представление данных 6](#_Toc181895901)

[10 Описание внутренних функций 7](#_Toc181895902)

[10.1 Определение функций 7](#_Toc181895903)

[10.2 Назначение функций и им сопутствующего 7](#_Toc181895904)

[11 Описание алгоритма 8](#_Toc181895905)

[11.1 Общее описание 8](#_Toc181895906)

[11.2 Блок-схема (сделать) 8](#_Toc181895907)

[12 Текст программы 9](#_Toc181895908)

[13 Результат работы программы 11](#_Toc181895909)

[14 Вывод 12](#_Toc181895910)

# Задание

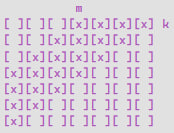


Рисунок 1.

Найти наибольший элемент заштрихованной области, зависящей от параметров и , квадратной матрицы размера (рисунок 1).

Необходимые данные: заявление о размере матрицы, параметры заштрихованной области, значения для заполнения матрицы — считать из файла.

# Неясности, предложения их решения и уточнения

Принято решение игнорировать пустые строки при вводе матрицы из файла, а также игнорировать значения, непомещающиеся в матрицу с учётом незаполненных (пустых) элементов матрицы.

Принято решение для введённых параметров начинать исчисление с единицы.

# Математическая постановка

## Данные

.

## Заштрихованная область

Заштрихованная область представляет из себя область исходной матрицы, проходящую по диагонали.

**Замечания.** Ввиду неналичия на матричной плоскости точки с индексом 0 параметры ограничены положительными значениями, в противном случае заштрихованная область будет не определена.

## Способ решения

Перебрать все элементы данной матрицы , обходя незаштрихованные области, описанные в подразделе 3.2, и найти среди них наибольшее — оно и будет ответом на задание.

# Контрольные примеры

| № | Содержание входного файла | Рассматриваемые значения в матрице | Ответ |
| --- | --- | --- | --- |
| 1 | 6 2 1  1 2 3 4 5 99 99 99  6 7 8 9 10  11 12 13 14 15 32 33 34  16 17 40 19 20 35 36 37 |  | 19 |
| 2 |  | — | Нет (недопустимый формат файла) |
| 3 | 7 1 3  78 40 50 60 50 11 44  18 80 45 11 21 23 10  17 89 45 62 32 62 28  33 22 22 22 12 19 46  48 12 21 54 63 12 49  47 89 85 12 45 30 57  82 61 33 57 90 66 45 |  | 90 |
| 4 | 2 2 -2.100  7.12 4.16 7.16  -2.100 |  | 7.12 |

# 

# Организация интерфейса пользователя

## Организация входного файла

Во входном файле первое считанное значение определяется как размерность матрицы, остальные значения представляют матрицу, в которой новая строка начинается после ‘/n’.

## Организация выходного файла

О1. Выввод общей информации.

О2. Вывод задания.

О3. Вывод указанной размерности матрицы.

О4. Вывод обработанной матрицы.

О5. Вывод реальной размерности матрицы.

О6. Вывод введенных параметров m и k.

О7. Вывод наибольшего эллемента массива в заштрихованной области.

## Организация консоли

О1. Вывод замечаний по вводу параметров.

О2. Вывод задания.

О3. Вывод размерности матрицы.

I1. Ввод параметра m.

I2. Ввод параметра k.

О4. Визуализация обработки.

О5. Вывод результата обработки.

# Реализация ввода/вывода

Использованы потоки ввода (cin>>) и вывода (cout<<) библиотеки «iostream», а также функции объекта cin: getline, clear, eof.

Использованы потоки чтения из файла (>>) и записи в файл (<<) библиотеки «fstream», а также функции объекта типа fstream: open, is\_open, fail, seekg, eof, close. Для настройки ввода были использованы функции setiosflags, resetiosflags библиотеки «iomanip».

# 9 Внутреннее представление данных

| Имя переменной | Тип | Назначение |
| --- | --- | --- |
| g | fstream | Выхадной файл |
| f | Входной файл |
| A[][] | float | Массив с матрицей |
| B[] | Массив заштрихованных значений |
| result | Результат |
| tmp | Вспомогательная переменная |
| temp |
| a | unsigned | Размерность матрицы |
| b | Вспомогательная переменная |
| i |
| j |
| h |
| size |
| s | char | Вспомогательная переменная |
| m | int | Вводимый параметр обработки |
| k |
| raz | Вспомогательная переменная |
| t |
| i |
| j |
| z |
| y |
| x |
| temp |
| N, M | const unsigned | Размер матрицы |

# Описание внутренних функций

## Определение функций

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя функции | Аргументы | | | | Возвращаемое значение |
| Входные | Выходные | Модифи-цируемые | Транзитные |
| CalcSize | unsigned &a, unsigned \*b, fstream& g, int& raz |  |  |  | bool |
| InpF | unsigned a, unsigned b, float A[N][M], fstream &g |  |  |  | bool |
| Out | float A[N][M], unsigned a, unsigned b, fstream &g |  |  |  | void |
| OutStroke | unsigned b, fstream &g, float A[M] |  |  |  | void |
| Process | float A[N][M], unsigned a, unsigned b, int m, int k, float& result |  |  |  | float |
| InM | int& m, unsigned a, fstream& g |  |  |  | bool |
| InK | int& k, unsigned a, fstream& g |  |  |  | bool |
| main | void |  |  |  | void |

## Назначение функций и описание им сопутствующего

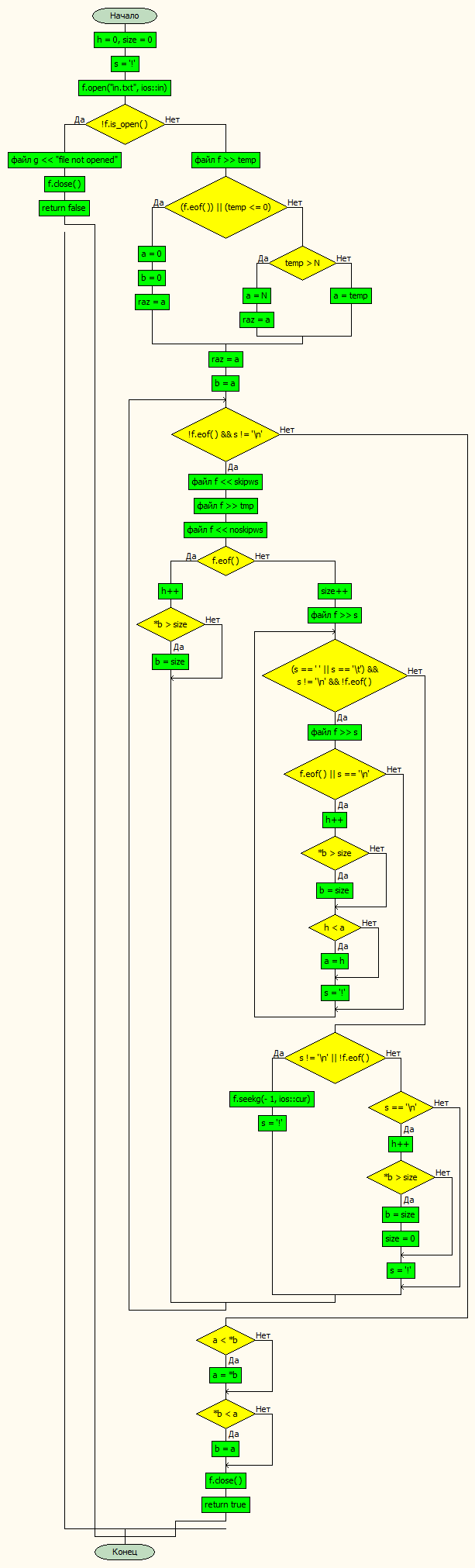
# Описание алгоритма

## Общее описание

Проанализировать файл, определить размер матрицы для чтения. Считать матрицу в массив. Вывести считанные данные в выходной файл. Перебрать числа в заштрихованной области в поиске максимального значения. Вывести ответ на задание в файл.

## Блок-схема

А. Функция CalcSize



Б. InK

A diagram of a computer program

Description automatically generated

В. InM

A diagram of a computer program

Description automatically generated

Г. InpF

A diagram of a computer program

Description automatically generated

Е. Out

A diagram of a algorithm

Description automatically generated

Ж. OutStroke

A diagram of a algorithm

Description automatically generated

Д. Main

A screenshot of a computer

Description automatically generated

З. Process

A diagram of a flowchart

Description automatically generated

# 12 Текст программы

/\*

Автор: Чучалин Иван Валентинович Группа: 4354 Версия программы: 5

Дата начала: 07.11.24 Завершения: 17.11.24

Задание: Найти наибольший элемент заштрихованной области таблицы вида:

m

[ ][ ][ ][x][x][x][x] k

[ ][ ][x][x][x][x][ ]

[ ][x][x][x][x][ ][ ]

[x][x][x][x][ ][ ][ ]

[x][x][x][ ][ ][ ][ ]

[x][x][ ][ ][ ][ ][ ]

[x][ ][ ][ ][ ][ ][ ]

\*/

#include <iostream>

#include <iomanip>

#include <fstream>

#include <locale>

#include <math.h>

using namespace std;

const unsigned N = 20, M = N;

bool CalcSize(unsigned &a, unsigned \*b, fstream& g, int& raz);

bool InpF(unsigned a, unsigned b, float A[N][M], fstream &g);

void Out(float A[N][M], unsigned a, unsigned b, fstream &g);

void OutStroke(unsigned b, fstream &g, float A[M]);

float Process(float A[N][M], unsigned a, unsigned b, int m, int k, float& result);

bool InM(int& m, unsigned a, fstream& g);

bool InK(int& k, unsigned a, fstream& g);

void main(void)

{

setlocale(LC\_ALL, "russian");

fstream g; float A[N][M], result = 0; unsigned a, b;int m, k, raz = 0;

g.open("out.txt", ios::out);

if (!g.is\_open()) cout << "File is not opened\n";

else

{

if (CalcSize( a, &b, g, raz) == false)

{

cout << "Не удалось считать размер\n";

g << "Не удалось считать размер\n";

}

else

{

if (InpF( a, b, A, g) == false)

{

cout << "Ошибка обработки\n";

g << "Ошибка обработки\n";

}

else

{

cout << "При вводе параметров m и k учитывайте, что исчисление столбцов и строк начинается с единицы.\n";

cout << "Также не забывайте, что вводимые значения должны быть целыми.\n\n";

cout << " m\n";

cout << " [ ][ ][ ][x][x][x][x] k\n";

cout << " [ ][ ][x][x][x][x][ ]\n";

cout << " [ ][x][x][x][x][ ][ ]\n";

cout << " [x][x][x][x][ ][ ][ ]\n";

cout << " [x][x][x][ ][ ][ ][ ]\n";

cout << " [x][x][ ][ ][ ][ ][ ]\n";

cout << " [x][ ][ ][ ][ ][ ][ ]\n\n";

g << "Автор: Чучалин Иван Валентинович\tГруппа : 4354\tВерсия программы : 5\nДата начала : 07.11.24\tЗавершения : 17.11.24\n";

g << "Задание: Найти наибольший элемент заштрихованной области таблицы вида:\n\n";

g << " m\n";

g << " [ ][ ][ ][x][x][x][x] k\n";

g << " [ ][ ][x][x][x][x][ ]\n";

g << " [ ][x][x][x][x][ ][ ]\n";

g << " [x][x][x][x][ ][ ][ ]\n";

g << " [x][x][x][ ][ ][ ][ ]\n";

g << " [x][x][ ][ ][ ][ ][ ]\n";

g << " [x][ ][ ][ ][ ][ ][ ]\n\n";

g << "Указанная размерность матрицы: " << raz << "\*" << raz << "\n";

g << "Ниже представлена обработанная матрица:\n";

Out(A, a, b, g);

g << "Реальная размерность матрицы: " << a << "\*" << a << ".\n";

cout << "Введите m\n";

cin >> m;

if (InM(m, a, g) == false)

{

cout << "Введены неправильные параметры обработки. Перезапустите программу.\n";

g << "Введены неправильные параметры обработки. Перезапустите программу.\n";

}

else

{

cout << "Введите k\n";

cin >> k;

if (InK(k, a, g) == false)

{

cout << "Введены неправильные параметры обработки. Перезапустите программу.\n";

g << "Введены неправильные параметры обработки. Перезапустите программу.\n";

}

else

{

g << "Введенное m: " << m << "\tВведенное k: " << k << "\n";

cout << "Визуализация обработки:\n";

Process(A, a, b, m, k, result);

cout << "Наибольший эллемент массива в заданной области: " << result << "\n";

g << "Наибольший эллемент массива в заданной области: " << result << "\n";

}

}

}

}

}

g.close();

}

void OutStroke(unsigned b, fstream& g, float A[M])

{

unsigned i;

for (i = 0; i < b; i++) g << A[i] << " ";

g << "\n";

}

void Out(float A[N][M], unsigned a, unsigned b, fstream& g)

{

unsigned i;

for (i = 0; i < a; i++) OutStroke(b, g, A[i]);

}

bool CalcSize(unsigned& a, unsigned\* b, fstream& g, int& raz)

{

fstream f; unsigned h = 0, size = 0; char s = '!'; float tmp; int temp;

f.open("in.txt", ios::in);

if (!f.is\_open())

{

g << "file not opened";

f.close();

return false;

}

else

{

f >> temp;

if ((f.eof()) || (temp <= 0))

{

a = 0;

\*b = 0;

raz = a;

}

else if (temp > N)

{

a = N;

raz = a;

}

else a = temp;

raz = a;

\*b = a;

while (!f.eof() && s != '\n')

{

f << skipws;

f >> tmp;

f << noskipws;

if (f.eof())

{

h++;

if (\*b > size) \*b = size;

}

else

{

size++;

f >> s;

while ((s == ' ' || s == '\t') && s != '\n' && !f.eof())

{

f >> s;

if (f.eof() || s == '\n')

{

h++;

if (\*b > size) \*b = size;

if (h < a)a = h;

s = '!';

}

}

if (s != '\n' || !f.eof())

{

f.seekg(-1, ios::cur);

s = '!';

}

else if (s == '\n')

{

h++;

if (\*b > size)

{

\*b = size;

size = 0;

}

s = '!';

}

}

}

if (a < \*b)a = \*b;

if (\*b < a)\*b = a;

f.close();

return true;

}

}

bool InpF(unsigned a, unsigned b, float A[N][M], fstream& g)

{

fstream f; unsigned i, j; char s = '!'; float tmp;

f.open("in.txt", ios::in);

if (f.eof()) return false;

else

{

f >> tmp;

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

{

f << skipws;

for (j = 0; j < b; j++) f >> A[i][j];

f << noskipws;

s = '!';

while (!f.eof() && s != '\n') f >> s;

}

return true;

}

f.close();

}

bool InM(int& m,unsigned a, fstream& g)

{

if (m-1 < 0)

{

cout << "Значение m не может быть меньше или равно нулю.\n";

g << "Введенное m: " << m << "\tЗначение m не может быть меньше или равно нулю.\n";

return false;

}

else if (m > a)

{

cout << "Значение m не может быть больше колличества столбцов.\n";

g << "Введенное m: " << m << "\tЗначение m не может быть больше колличества столбцов.\n";

return false;

}

else

{

return true;

}

}

bool InK(int& k, unsigned a, fstream& g)

{

if (k - 1 < 0)

{

cout << "Значение k не может быть меньше или равно нулю.\n";

g << "Введенное k: " << k << "\tЗначение k не может быть меньше или равно нулю.\n";

return false;

}

else if (k > a)

{

cout << "Значение k не может быть больше колличества строк.\n";

g << "Введенное k: " << k << "\tЗначение k не может быть больше колличества строк.\n";

return false;

}

else

{

return true;

}

}

float Process(float A[N][M], unsigned a, unsigned b, int m, int k, float& result)

{

int t, i = 0, j = m, z, y; float B[N\*N], temp; int x = -1;

y = k;

while (i!=a)

{

m = m - 1;

if (m < 0) m = 0;

while (i < k)

{

j = m;

while (j < b)

{

x++;

B[x] = A[i][j];

cout << B[x] << ' ';

z = x;

j++;

}

i++;

cout << '\n';

}

b--;

for (t = 0; t < y; t++) k++;

if (k > a)

{

k = a;

}

}

x = 0;

temp = B[x];

while (x < z)

{

x++;

if (temp < B[x]) temp = B[x];

else result = temp;

}

return result;

}

# Результат работы программы

# Вывод

Были изучены аспекты использования ссылок, указателей, двумерных массивов и функций в языке программирования C++, а также способы их задания и объявления, и была пройдена практика по использованию функций и двумерных массивов при обработке матрицы данных.