**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)

# 1 Задание

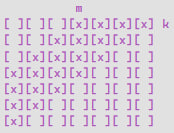


Рисунок 1.

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.

# 2 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 брать модуль значений введённых параметров и заявления о размере матрицы, а также отбрасывать дробную часть в случае, если введённое значение нецелое, с расчётом на то, что минус (отрицательное значение) и/или дробная часть — опечатка(-и) при вводе данных.

# 3 Математическая постановка

## 3.1 Данные

.

## 3.2 Заштрихованная область

Заштрихованная область представляет из себя область исходной матрицы, которая не включает треугольноподобные области, находящиеся выше линий с их пересечением, параллельных прямым, образованным парой точек на матричной плоскости и или и , и содержащих в себе соответственно точки на матричной плоскости и , или ниже линий с их пересечением, параллельных тем же прямым, но соответственно содержащих в себе точки на матричной плоскости и , а также те треугольноподобные области, что находятся левее линий с их пересечением, параллельным тем же прямым и соответственно содержащих точки на матричной плоскости и , либо правее линий с их пересечением, параллельным тем же прямым и соответственно содержащих точки на матричной плоскости и .

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

При треугольноподобная область выходит за границы матричной плоскости (по вертикали), аналогичное наблюдается и при (по горизонтали).

## 3.3 Способ решения

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

# 4 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 |

# 5 Ограничения, обусловленные вычислительным устройством

Значения .

В связи с небесконечной точностью хранимых вещественных значений различие между ними распознаётся с точностью до 30 знаков после запятой, а именно проверка вещественных значений на равенство проводится с использованием многомалого (бесконечномалого) значения , что в общем случае проводится по формуле (1):

|  |  |  |
| --- | --- | --- |
|  |  | (1) |

где , — сравниваемые на тождественность значения.

Положим за максимальный объем памяти, выделенный под (двумерный) массив обрабатываемых данных, 512 000 байт, тогда максимальный размер такого массива — квадратная матрица размером , т.е. и, следственно, .

# 6 Организация интерфейса пользователя

Макет ввода/вывода:

O1: Лабораторная работа №5

Автор: Н.А. Чехонадских

Версия: 1.0.2 Даты: начало 24.10.2024, конец 06.11.2024

Ограничения: -10^10 < a(i,j) < 10^10 с точностью до 30 знаков после запятой; N,k,m < 357.

Задание: Найти наибольший элемент заштрихованной области, зависящей от параметров m и k, квадратной матрицы A размера NxN.

m

+-+-+-+-+-+-+-+

|X|X| | | |X|X|

+-+-+-+-+-+-+-+

k|X|X|X| |X|X|X|

+-+-+-+-+-+-+-+

| |X|X|X|X|X| |

+-+-+-+-+-+-+-+

| | |X|X|X| | |

+-+-+-+-+-+-+-+

| |X|X|X|X|X| |

+-+-+-+-+-+-+-+

|X|X|X| |X|X|X|

+-+-+-+-+-+-+-+

|X|X| | | |X|X|

+-+-+-+-+-+-+-+

O2\*: Введите путь к файлу, который требуется открыть: \_

I2\*: [ ]["]t\_t.t\_t["][ ]

O2.1\*: Не удалось открыть файл "t\_t.t\_t".

O2.2\*: Превышено время ожидания входного файла. Попробуйте ещё раз позже.

O3\*\*\*: Недопустимый формат файла.

O4.1\*\*: Заявленный размер матрицы не может быть отрицательным и/или нецелым числом.

O4.2\*\*: Считанная матрица меньше того размера, который был заявлен.

O4.3\*\*: Значение \_ должно быть положительным.

O4.4\*\*: Значение \_ должно быть целым.

O5\*\*\*\*: Недостаточно памяти для продолжения работы программы.

O6\*\*\*\*\*: Не удалось открыть выходной файл для записи.

\* Вывод O2 и ввод I2 используются не более 4 раз, при этом каждый новый раз предоставляется пользователю по прошествию неудачной попытки открытия файла, заданного пользователем, о чём выводится сообщение об ошибке O2.1. По прошествии 4 неудачных попыток открытия заданных пользователем файлов выводится сообщение O2.2 и программа завершает свою работу.

\*\* При обнаружении недопустимых данных выводится предупреждение: при вводе отрицательного и/или нецелого числа для — O4.1, при невозможности считать матрицу заявленного размера — O4.2, при вводе отрицательного числа для — O4.3 с наименованием параметра заместо нижнего подчёркивания, нецелого числа для этих параметров — O4.4 с наименованием параметра заместо нижнего подчёркивания.

\*\*\* При невозможности считать заявление о размере матрицы и значений для параметров выводится сообщение об ошибке O3 и программа завершает свою работу.

\*\*\*\* По прошествии неудачной попытки выделить память под считанные из файла данные выводится сообщение об ошибке O5 и программа завершает свою работу.

\*\*\*\*\* По прошествии неудачной попытки открыть выходной файл для записи выводится сообщение об ошибке O6 и программа завершает свою работу.

# 7 Формат файлов

## 7.1 Организация входного файла

Макет организации файла (… — один или несколько пробелов и/или знаков табуляции, N — , K — , M — , A — ):

...N...K...M...

[...A][...A][...][

...A][...A][...]

Макет чисел N, K, M:

d\_d

Макет чисел A (вывод производится с минимальной шириной в 7 символов):

sd\_d[.d\_d][esd\_d]

## 7.2 Организация выходного файла

Макет организации файла:

Введённая матрица (d\_dxd\_d):

[sd\_d[.d\_d][esd\_d]][ sd\_d[.d\_d][esd\_d]][

sd\_d[.d\_d][esd\_d]][ sd\_d[.d\_d][esd\_d]]

Введённые параметры заштрихованной области: k=d\_d, m=d\_d.

[Максимальное значение в заштрихованной области: sd\_d[.d\_d][esd\_d].]

[Максимального значения нет. Заштрихованная область представляет из себя пустое множество или не является определённой.]

# 8 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».

Была использована библиотека «Windows.h» с предшествующим определением макроса WIN32\_LEAN\_AND\_MEAN для использования функций SetConsoleCP и SetConsoleOutputCP, настраивающих соответственно язык ввода и вывода в консоль (русский).

# 9 Внутреннее представление данных

| Имя переменной | Тип | Назначение |
| --- | --- | --- |
| f | fstream | Интерфейс ввода в файл и вывода из файла. |
| a | float\*\* | Считанная матрица . |
| res | float | Ответ на задание. |
| hr | int | Результат работы функции (успешность выполнения). |
| n | unsigned int | Значение (фактического) . |
| k | Значение параметра . |
| m | Значение параметра . |

Значения элементов считанной матрицы хранятся внутри программы следующим образом:

Изображение выглядит как текст, рукописный текст, Шрифт, письмо

Автоматически созданное описание

# 10 Описание внутренних функций

## 10.1 Определение функций

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Имя функции | Аргументы | | | | Возвращаемое значение |
| Входные | Выходные | Модифи-цируемые | Транзитные |
| MAX | a, b |  |  |  | float |
| MAX |  |  |  | a, b, c, d | float |
| InputFile |  | f |  |  | int |
| AnalyzeFile |  | size\_container, k\_container, m\_container | f |  | int |
| ReadArray | size |  | f |  | float\*\* |
| PrintArrayLine | line, size |  | f |  | void |
| PrintArray | array, size |  | f |  | void |
| MainOutput | size, k, m |  | f | array | void |
| AnalyzeArray | array, k, size, m | result\_container |  |  | int |
| PrintResult | result |  | f |  | void |
| EndArray | array, size |  |  |  | void |

## 10.2 Назначение функций и описание им сопутствующего

В данном разделе под словом «параметры» («параметр») подразумеваются «параметры заштрихованной области» (подраздел 3.2).

**Функция MAX** с двумя аргументами a и b типа float, передающихся по значению, определяет наибольшее среди них и возвращает его значение.

**Функция MAX** с четырьмя аргументами a, b, c и d типа float, передающихся по значению, определяет наибольшее среди них и возвращает это (максимальное) значение.

**Функция InputFile** отвечает за ввод пути ко входному файлу пользователем и открывает его с помощью переданного по ссылке объекта f типа fstream, при этом в случае 4 неудачных попыток открытия входного файла возвращается ненулевое значение, сигнализирующее о неудачном вводе файла пользователем, в ином случае — нулевое значение, сигнализирующее об успешном выполнении функции.

**Функция AnalyzeFile** анализирует данные во входном файле, открытом с помощью передаваемого по ссылке через аргумент f объекта типа fstream, и определяет размер матрицы, которую необходимо считать, в переданную по ссылке через аргумент size\_container переменную типа unsigned int с учётом заявления о размере вводимой матрицы и фактического наполнения файла, а также считывает скорректированные значения введённых параметров в переданные по ссылке через аргументы k\_container и m\_container переменные типа unsigned int. В случае обнаружения неверного формата файла, чем является отсутствие заявления о размере вводимой матрицы и значений для каждого из двух параметров, 0-ой бит (отсчёт битов ведётся с индекса 0 справа налево) возвращаемого значения будет иметь нулевое значение; при допустимом формате файла 0-ой бит возвращаемого значения будет иметь ненулевое значение. При невозможности считать матрицу заявленного размера за размер матрицы будет взят размер наибольшей квадратной матрицы, доступной для чтения, а также 27-ой бит в возвращаемом значении будет иметь ненулевое значение. Корректировка считанных значений заявления о размере вводимой матрицы и параметров следующая:

* + при отрицательном и/или нецелом значении в заявлении о размере вводимой матрицы будет использовано абсолютное значение без дробной части, а также 30-ый бит возвращаемого значения будет иметь ненулевое значение, в ином случае — нулевое;
  + при вводе отрицательного значения для параметра или будет использовано абсолютное значение, а также 29-ый или 28-ой соответственно бит возвращаемого значения будет иметь ненулевое значение, в ином случае — нулевое;
  + при вводе нецелого значения для параметра или дробная часть этого значения будет отброшена, а также 26-ой или 25-ый соответственно бит возвращаемого значения будет иметь ненулевое значение, в ином случае — нулевое.

**Функция ReadArray** выделяет память под матрицу необходимого размера, переданного по значению через аргумент size типа unsigned int, и возвращает (ненулевой) указатель на заполненную считанными из файла, открытого с помощью переданного по ссылке через аргумент f объекта типа fstream, данными матрицу (подробнее об организации хранения элементов матрицы в разделе 9), в случае невозможности выделения требуемой памяти возвращается нулевое значение.

**Функция PrintArrayLine** выводит строку из матрицы, указатель на которую передан по значению через аргумент line типа float\* (подробнее об организации хранения элементов матрицы в разделе 9), размера, переданного по значению через аргумент size типа unsigned int, в файл, открытый с помощью переданного по ссылке через аргумент f объекта типа fstream.

**Функция PrintArray** выводит матрицу, указатель на которую передан по значению через аргумент array типа float\*\* (подробнее об организации хранения элементов матрицы в разделе 9), размера, переданного по значению через аргумент size типа unsigned int, в файл, открытый с помощью переданного по ссылке через аргумент f объекта типа fstream.

**Функция MainOutput** выводит в файл, открытый с помощью переданного по ссылке через аргумент f объекта типа fstream, заголовок с обозначением размера считанной матрицы, переданного по значению через аргумент size типа unsigned int, и введённых параметров и , значения которых переданы по значению типа unsigned int через аргументы k и m соответственно, и считанную матрицу, указатель на которую передан по значению через аргумент array типа float\*\* (подробнее об организации хранения элементов матрицы в разделе 9).

**Функция AnalyzeArray** просматривает матрицу, указатель на которую передан по значению через аргумент array типа float\*\* (подробнее об организации хранения элементов матрицы в разделе 9), размера, переданного по значению через аргумент size типа unsigned int, и выполняет задание — ищет наибольшее значение в заштрихованной области с учётом параметров и , переданных по значению типа unsigned int через аргументы k и m соответственно. Найденное наибольшее значение записывается в переменную, адрес которой передан по указателю через аргумент result\_container типа float\*, и возвращается нулевое значение; в случае, если заштрихованная область представляет из себя пустое множество или не определена, возвращается ненулевое значение.

**Функция PrintResult** выводит в файл, открытый с помощью переданного по ссылке через аргумент f объекта типа fstream, результат выполнения задания (поиска наибольшего значения в заштрихованной области при её определённости), переданного с помощью указателя на переменную через аргумент result типа float\*, если аргумент result имеет ненулевое значение, в ином случае — сообщение о том, что заштрихованная область представляет из себя пустое множество или не определена.

**Функция EndArray** освобождает память, выделенную под матрицу, указатель на которую передан через аргумент array типа float\*\* (подробнее об организации хранения элементов матрицы в разделе 9), размера, переданного по значению через аргумент size типа unsigned int, если аргумент array имеет ненулевое значение, в ином случае не делает ничего.

# 11 Описание алгоритма

## 11.1 Общее описание

Получить путь к открываемому файлу от пользователя. Проанализировать файл, определить размер матрицы для чтения. Считать матрицу в динамически выделенный буфер. Вывести считанные данные в выходной файл. Перебрать числа в заштрихованной области в поиске максимального значения. Вывести ответ на задание в файл.

## 11.2 Блок-схема (сделать)

# 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;

}

# 13 Результат работы программы

Результат работы программы (далее — Результат) для контрольного примера №1 (раздел 4). Вывод в консоль:

Лабораторная работа №5

Автор: Н.А. Чехонадских

Версия: 1.0.2 Даты: начало 24.10.2024, конец 06.11.2024

Ограничения: -10^10 < a(i,j) < 10^10 с точностью до 30 знаков после запятой; N,k,m < 357.

Задание: Найти наибольший элемент заштрихованной области, зависящей от параметров m и k, квадратной матрицы A размера NxN.

m

+-+-+-+-+-+-+-+

|X|X| | | |X|X|

+-+-+-+-+-+-+-+

k|X|X|X| |X|X|X|

+-+-+-+-+-+-+-+

| |X|X|X|X|X| |

+-+-+-+-+-+-+-+

| | |X|X|X| | |

+-+-+-+-+-+-+-+

| |X|X|X|X|X| |

+-+-+-+-+-+-+-+

|X|X|X| |X|X|X|

+-+-+-+-+-+-+-+

|X|X| | | |X|X|

+-+-+-+-+-+-+-+

Введите путь к файлу, который требуется открыть: 1.txt

Считанная матрица меньше того размера, который был заявлен.

Вывод в файл:

Введённая матрица (4x4):

1 2 3 4

6 7 8 9

11 12 13 14

16 17 40 19

Введённые параметры заштрихованной области: k=2, m=1.

Максимальное значение в заштрихованной области: 19.

Результат для контрольного примера №2 (раздел 4). Вывод в консоль (в сокращении):

...

Введите путь к файлу, который требуется открыть: 2.txt

Недопустимый формат файла.

Вывод в файл не проводился.

Результат для контрольного примера №4 (раздел 4). Вывод в консоль (в сокращении):

...

Введите путь к файлу, который требуется открыть: 4.txt

Считанная матрица меньше того размера, который был заявлен.

Значение m должно быть положительным.

Значение m должно быть целым.

Вывод в файл:

Введённая матрица (1x1):

7.12

Введённые параметры заштрихованной области: k=2, m=2.

Максимальное значение в заштрихованной области: 7.12.

# 14 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.