Федеральное государственное автономное образовательное учреждение высшего образования «Санкт-Петербургский государственный электротехнический университет «ЛЭТИ»

им. В.И. Ульянова (Ленина)»

кафедра систем автоматизированного проектирования

**ОТЧЕТ**

**по лабораторной работе**

**по дисциплине «Программирование»**

**Тема: «Обработка двумерных массивов»**

Выполнил : Игнатьев Валерий Алексеевич

Группа № 1309

Преподаватель: Калмычков Виталий Анатольевич

Санкт-Петербург

2021

Оглавление

[Цель 3](#_Toc88165009)

[Задача 3](#_Toc88165010)

[Анализ программы и контрольный пример 3](#_Toc88165011)

[Изучение необходимого материала и подготовка к выполнению лабораторной работы 3](#_Toc88165012)

[Математическая постановка задачи 3](#_Toc88165013)

[Макет ввода-вывода 3](#_Toc88165014)

[Определение внутреннего формата представления данных 3](#_Toc88165015)

[Процесс выполнения лабораторной работы 3](#_Toc88165016)

[Текст программы 3](#_Toc88165017)

[Контрольные тесты и примеры 3](#_Toc88165018)

[Вывод 3](#_Toc88165019)

# Цель

Научиться обрабатывать двумерные массивы, научиться использовать функции, закрепить умения в работе с файлами.

# Задача

Задана квадратная таблица размера N на N. Преобразовать ее, осуществив поворот элементов вокруг ее центра на 90 градусов по часовой стрелке.

# Анализ программы и контрольный пример

Для N=4

1 1 1 1

2 2 2 2

3 3 3 3

4 4 4 4

Отвтет:

4 3 2 1

4 3 2 1

4 3 2 1

4 3 2 1

# Изучение необходимого материала и подготовка к выполнению лабораторной работы

Функции в с++ бывают типа void и int. Void при вызове ничего не возвращает, int же возвращает. Функции принимают параметры через круглые скобки.

Для реализации цикла, в ходе которого вводились и обрабатывались элементы массива, использовался оператор for.

Для реализации записи в файл и из файла информации использовалась библиотека fstream.

Двумерный массив из файла вводился посимвольно.

Использование знакомых ранее библиотек iomanip и iostream.

# Математическая постановка задачи

Дано: двумерный массив N\*N в файле ввода input.

Найти: Перевернуть матрица.

Способ решения: посимвольно вводить массив в поток, после ввода делать проверку на конец строки для корректировки размера массива.

# Макет ввода-вывода

Для организации ввода вывода пользуемся операторами cin>> cout<< из библиотеки iostream.

Макеты ввода-вывода в двух версиях идентичны.

1. В случае удачного открытия файла в консоль выводится сообщение о том, что файл открыт, иначе выводится сообщения о том, что файл не открыт, программа прекращает работать;
2. Ввод t(количество тестов);
3. Считывание n и пропуск строки(n строго не отрицательно)
4. Ввод n строк чисел
5. Далее идет посимвольный ввод элементов квадратной матрицы из файла в поток;
6. Вывод в файл для вывода output.txt обрабатываемой матрицы
7. Вывод первоначальной матрицы и после обработки

# Определение внутреннего формата представления данных

Программа разбита на 6 функций: void inputstroka, void input, void output, void process, int main, int realsize.

* Void inputstroka

Эта функция посимвольно вводит элементы строки матрицы из файла ввода input.txt.

|  |  |  |
| --- | --- | --- |
| Название переменной | Тип переменной | Пояснение |
| s[] | int | Массив, куда записываются элементы строки матрицы |
| n | int | Количество элементов строки |
| j | int | Счетчик |
| b | char | Символьная переменная, которая считывает ненужные элементы матрицы |

* Void input

Эта функция построчно записывает матрицу в поток, используя функцию inputstroka.

|  |  |  |
| --- | --- | --- |
| Название переменной | Тип переменной | Пояснение |
| S[20][20] | int | Массив, куда записываются элементы матрицы |
| n | int | Количество строк |
| i | int | Счетчик |

* Void output

Эта функция выводит элементы матрицы в консоль и в файл для вывода output.txt.

|  |  |  |
| --- | --- | --- |
| Название переменной | Тип переменной | Пояснение |
| Z[20][20] | int | Массив, куда записываются элементы матрицы |
| n | int | Количество строк |
| i | int | Счетчик строк |
| j | int | Счетчик столбцов |

* Void process

Эта функция ходит по массиву, записывает в новый одномерный массив заштрихованные элементы и выбирает среди них наибольший.

|  |  |  |
| --- | --- | --- |
| Название переменной | Тип переменной | Пояснение |
| Z[20][20] | int | Массив, куда записываются элементы матрицы |
| n | int | Количество строк и столбцов квадратной матрицы |
| i | int | Счетчик строк |
| j | int | Счетчик столбцов |
| с | int | Счетчик элементов нового массива |
| А[] | int | Новый массив для записи в него нужных элементов |
| k, m | int | Координаты по столбцам и строкам центрального заштрихованного элемента |

* Int main

Главная функция открывает файлы для ввода и вывода, вызывает функции, также она содержит приветствие и условие задачи.

|  |  |  |
| --- | --- | --- |
| Название переменной | Тип переменной | Пояснение |
| A[N][N] | int | Массив, куда записываются элементы матрицы |
| N | const int | Максимальный размер матрицы |
| n | int | Количество строк и столбцов квадратной матрицы |
| i | int | Счетчик |
| Z[100] | int | Новый массив для записи в него нужных элементов |
| k, m | int | Координаты по столбцам и строкам центрального заштрихованного элемента |
| s | string | Строка матрицы для расчета ее реальной длины |

* Int realsize

Эта функция определяет реальный размер строки матрицы. Функция возвращает размер строки.

|  |  |  |
| --- | --- | --- |
| Название переменной | Тип переменной | Пояснение |
| s | string | Строка матрицы для расчета ее реальной длины |
| i | int | Счетчик |
| size | int | Размер строки |
| chislo | int | Маркер начала числа в строке |

# Процесс выполнения лабораторной работы

* Void inputstroka

Функция считывает из строки матрицы n элементов, оставшиеся элементы строки, ненужные для программы, считывает символьная переменная b до тех пор, пока строка не закончится.

* Void input

Эта функция вводит в поток строки массива, используя функцию inputstroka

* Void output

Функция вывода матрицы введенного размера в файл output.txt и консоль.

* Void process

Функция выполняется сравнение необходимых элементов матрицы сначала по диагонали, горизонтали, и затем по вертикали. В конце сравнения выводятся сравниваемые элементы матрицы, записанные в новый одномерный массив, наибольшее значение, среди всех значений нового одномерного массива.

* Void realsize

Эта функция определяет реальный размер строки, чтоб избежать случаев, когда программа не может обработать количество n строк, которое превышает размер строки матрицы в файле, поэтому эта функция обрабатывает размер квадратной матрицы n, вводимый пользователем.

* Int main

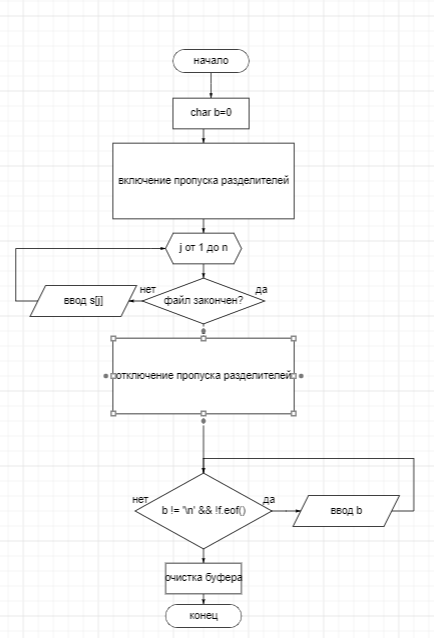
C самого начала открывается файл, в случае удачного открытия файла пользователь ввода вводит желаемый размер матрицы. Если пользователь ввел отрицательное значение, n становится равным единице. После этого программа проверяет, есть ли введенное количество строк и столбцов в файле. Для этого в цикле от 0 до n с помощью команды getline() берется строка, размер которой определяется с помощью функции int realsize, после в n передается минимальное значение между n и size, значение которого возвращается из функции int realsize. После обработки n выводится ее новое значение.

После этого файл для ввода закрывается, открывается вновь. Пользователь вводит значение k, m, потом эти значение обрабатываются, принимая минимальное значение между n и k, n и m. Потом вызывается функция input, в которую передаются двумерный массив A, файл для ввода f, значение n. После файл для ввода закрывается окончательно, открывается файл для вывода fout, в него выводится значение n, вызывается функция output, в которую передаются двумерный массив A, файл для вывода fout, значение n.

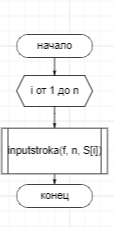
В самом конце запускается функция process, в которую передаются двумерный массив A, одномерный массив Z, файл для вывода fout, значение n, k, m. В файл и консоль выводятся элементы нового массива, наибольший из них.

Блок-схемы

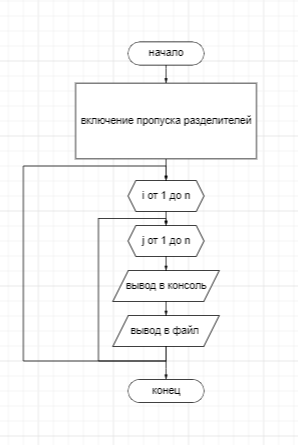
* Void inputstroka



* Void input



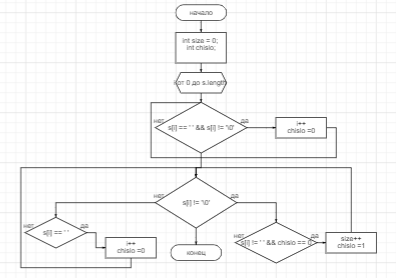
* Void output



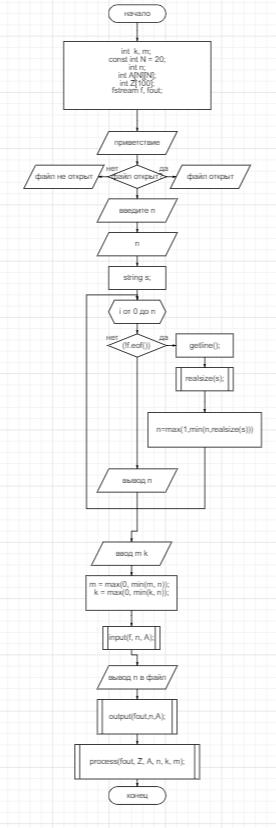
* Void process



* Void realsize



* Int main



# Текст программы

#include <iostream>

#include <iomanip>

#include <fstream>

#include <math.h>

#include <string>

using namespace std;

/\*

Автор: Бобков Владислав Дмитриевич, гр. 1309

Начало: 04.11.2021

Конец:14.11.2021

Задание 20(о):

0 0 0 0 0 X 0

0 0 0 0 X 0 0

0 0 0 X X X X

0 0 X X 0 0 0

0 X 0 X 0 0 0

X 0 0 X 0 0 0

0 0 0 X 0 0 0

\*/

void inputstroka(fstream& f, int& n, int s[]) {

char b=0;

f << setiosflags(ios::skipws);

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

if (!f.eof())

{

f >> s[j];

}

}

f << resetiosflags(ios::skipws);

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

f << resetiosflags(ios::skipws);

f >> b;

}

cout << endl;

f.flush();

}

int realsize(string s) {

int size = 0;

int chislo;

for (int i = 0; i < s.length(); ) {

while (s[i] == ' ' && s[i] != '\0')

i++;

chislo = 0;

while (s[i] != '\0') {

if (s[i] != ' ' && chislo == 0)

{

chislo = 1;

size++;

}

else if (s[i] == ' ')

chislo = 0;

i++;

}

}

return size;

}

void input(fstream& f, int& n, int S[20][20]) {

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

inputstroka(f, n, S[i]);

}

}

void output(fstream& f, int& n, int Z[20][20]) {

f << setiosflags(ios::skipws);

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

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

cout << Z[i][j]<<'\t';

f << Z[i][j] <<'\t';

}

cout << endl;

f << endl;

}

}

void process(fstream& f, int A[], int Z[20][20], int& n, int& k, int& m) {

int c = 0;

int max = A[1];

for (int i = k - 1, j = m - 1; i > -1 && j > -1 && i <= n - 1 && j <= n - 1; i--, j++)

{

A[c] = Z[i][j];

if (A[c] >= max) {

max = A[c];

}

c++;

}

for (int i = k, j = m - 2; i > -1 && j > -1 && i <= n - 1 && j <= n - 1; i++, j--)

{

A[c] = Z[i][j];

if (A[c] >= max) {

max = A[c];

}

c++;

}

for (int i = k, j = m - 1; i < n && i <= n - 1 && j <= n - 1; i++)

{

A[c] = Z[i][j];

if (A[c] >= max) {

max = A[c];

}

c++;

}

for (int i = k - 1, j = m; j < n && i <= n - 1 && j <= n - 1; j++)

{

A[c] = Z[i][j];

if (A[c] >= max) {

max = A[c];

}

c++;

}

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

cout << A[i] << endl;

f << A[i] << endl;

}

cout << "Наибольшее из этих значений: " << max;

f << "Наибольшее из этих значений: " << max;

}

int main()

{

setlocale(LC\_ALL, "Russian");

int k, m;

const int N = 20;

int n;

int A[N][N];

int Z[100];

fstream f, fout;

cout << "Бобков Владислав Дмитриевич, гр. 1309\n";

cout << "Версия: 5.1.1 Дата: начало(04.11.2021) конец(14.11.2021)\n\n";

cout << "Формулировка задачи: Найти наибольший элемент заштрихованной области таблицы A размера N \* N\n\n";

cout << "0 0 0 0 0 X 0\n";

cout << "0 0 0 0 X 0 0\n";

cout << "0 0 0 X X X X\n";

cout << "0 0 X X 0 0 0\n";

cout << "0 X 0 X 0 0 0\n";

cout << "X 0 0 X 0 0 0\n";

cout << "0 0 0 X 0 0 0\n\n";

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

if (f.is\_open())

cout << "файл открыт\n";

else {

cout << "файл открыть не удалось\n";

}

cout << "Введите желаемую длину и ширину массива(<=10)\n ";

cin >> n;

string s;

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

if (!f.eof()) {

getline(f, s);

n = max(1, min(n, realsize(s)));

}

}

cout << "Доступный размер массива: " << n<<'\n';

f.close();

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

cout << "Введите характеризующие параметры m и k\n";

cin >> m >> k;

m = max(0, min(m, n));

k = max(0, min(k, n));

input(f, n, A);

f.close();

fout.open("output.txt", ios::out);

fout << "Для матрицы размером "<<n<<endl;

output(fout,n,A);

cout << endl << endl;

process(fout, Z, A, n, k, m);

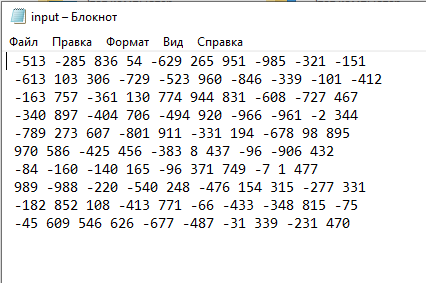
return 0;

}

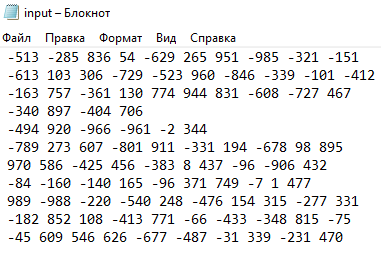
# Контрольные тесты и примеры

Проведем 3 теста для разных вариаций входного файла:

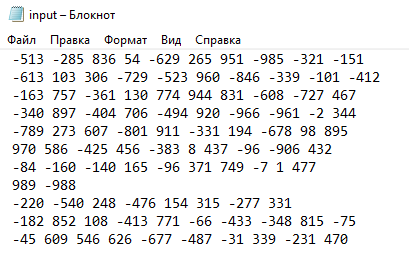
1. N=10, m=5, k=5



1. N=7, m=3, k=3

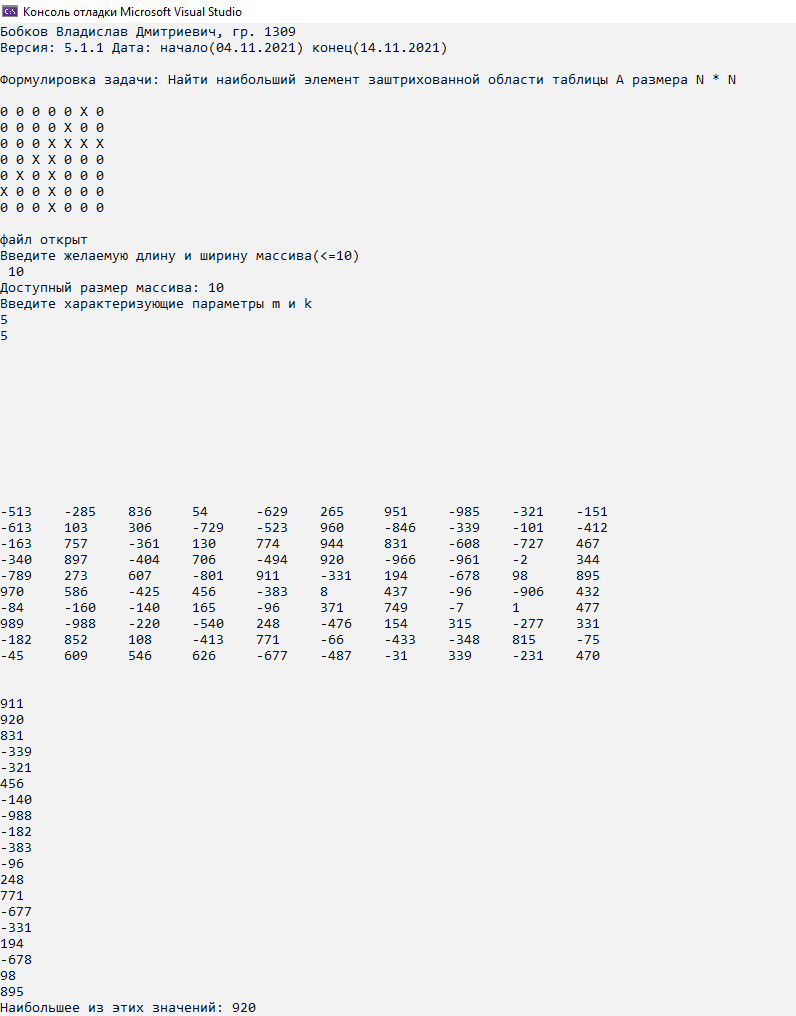


1. N=11, m=3, k=5

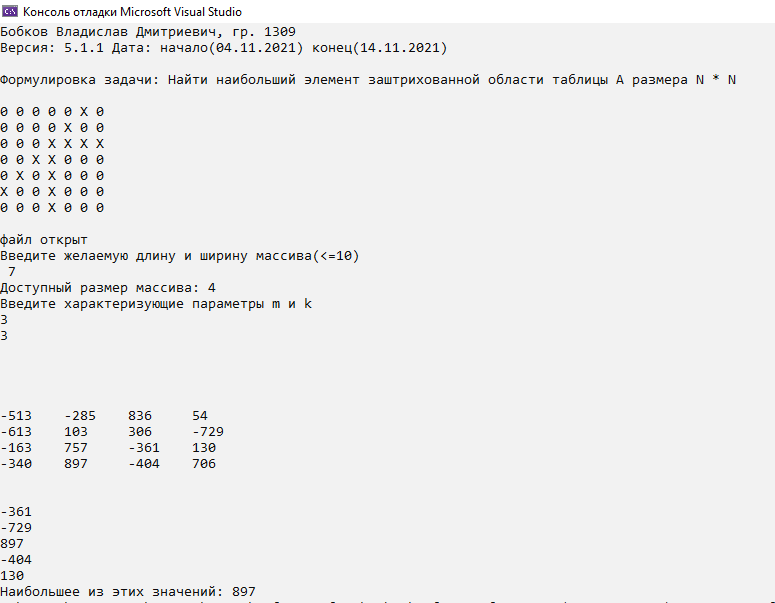


Результаты тестов:

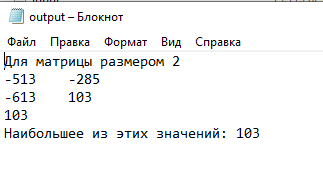
Первый тест:



Второй тест:



Третий тест:



# Вывод

Я научился работать с двумерными массивами: вводить их из файла, обрабатывать и выводить в файл и в консоль, разбивать программу на функции для решения задач.