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 4](#_Toc88165013)

[Макет ввода-вывода 4](#_Toc88165014)

[Определение внутреннего формата представления данных 5](#_Toc88165015)

[Процесс выполнения лабораторной работы 7](#_Toc88165016)

[Текст программы 14](#_Toc88165017)

[Контрольные тесты и примеры 14](#_Toc88165018)

[Вывод 18](#_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

Ответ:

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 8 функций: int read\_one\_int(int &y); void solve(int t); void incorrect\_test\_case(int t); void print\_matrix(int a[], int n, int type); void render\_array(int a[], int n, int k); void clockewise(int a[], int n); void skip();

1. int read\_one\_int(int &y);

Эта функция считывает 1 символ ввода input.txt.

|  |  |  |
| --- | --- | --- |
| Название переменной | Тип переменной | Пояснение |
| y | int | Если 0 то симвоола коца строки не было иначе был |
| с | char | Текущий символ |
| x | int | Встречался ли символ |
| res | int | Число ответ |

2. void solve(int t)

Эта функция обрабатывает тест под номером t

|  |  |  |
| --- | --- | --- |
| Название переменной | Тип переменной | Пояснение |
| t | int | Номер теста |

3. void incorrect\_test\_case(int t)

Эта функция выводит сообщение о том что тест под номером t неверный

|  |  |  |
| --- | --- | --- |
| Название переменной | Тип переменной | Пояснение |
| t | int | Номер теста |

4. void print\_matrix(int a[], int n, int type);

Выводит текущее состояние массива

|  |  |  |
| --- | --- | --- |
| Название переменной | Тип переменной | Пояснение |
| A[n] | int | Массив, куда записываются элементы матрицы |
| n | int | Количество строк и столбцов |
| type | int | Текущее состояние массива |

5 . void render\_array(int a[], int n, int k);

Приводит массив в ранее заданими размерами n \* n к k \* k.

|  |  |  |
| --- | --- | --- |
| Название переменной | Тип переменной | Пояснение |
| A[] | int | Массив, куда записываются элементы матрицы |
| n | int | Количество строк ранее заданых |
| k | int | Реальные размеры массива |

6. void clockewise(int a[], int n);

Поворачивает матрицу a по часовой стрелке

|  |  |  |
| --- | --- | --- |
| Название переменной | Тип переменной | Пояснение |
| A[] | int | Массив, куда записываются элементы матрицы |
| n | int | Количество строк и столбцов квадратной матрицы |

7.void skip()

Функция которая пропускает все символы до конца строки из input.txt

|  |  |  |
| --- | --- | --- |
| Название переменной | Тип переменной | Пояснение |
| c | char | Для считывания очередного символа из фала ввода |

# Процесс выполнения лабораторной работы

* int read\_one\_int(int &y);

Эта функция считывает 1 символ ввода input.txt.

* void solve(int t)

Эта функция обрабатывает тест под номером t

* .void incorrect\_test\_case(int t)

Эта функция выводит сообщение о том что тест под номером t неверный

* void print\_matrix(int a[], int n, int type)

Выводит текущее состояние массива

* void render\_array(int a[], int n, int k);

Приводит массив в ранее заданими размерами n \* n к k \* k.

* void clockewise(int a[], int n);

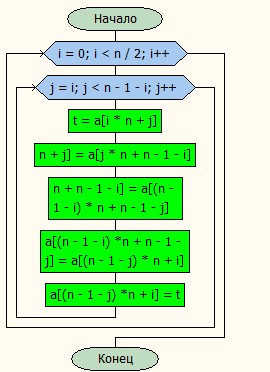
Поворачивает матрицу a по часовой стрелке

* void skip()

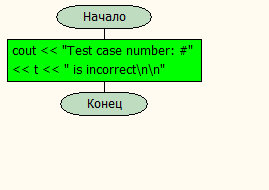
Функция которая пропускает все символы до конца строки из input.txt

Блок-схемы

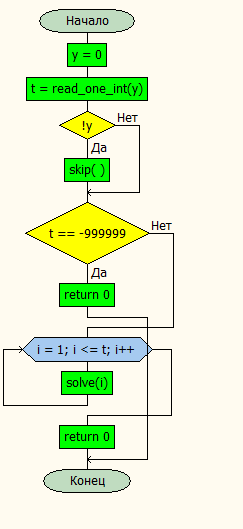
* Void clockewise



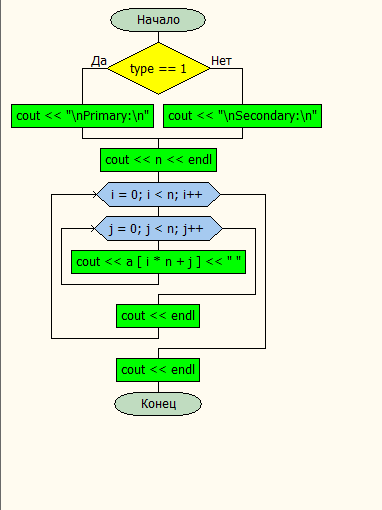
* Void incorrect\_test\_case



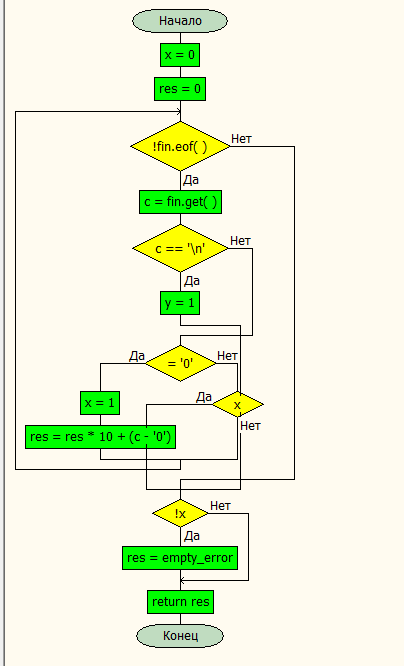
* Int main



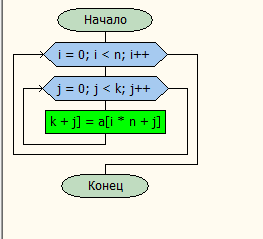
* Void print\_matrix



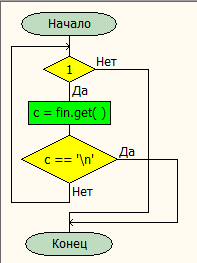
* int read\_one\_int



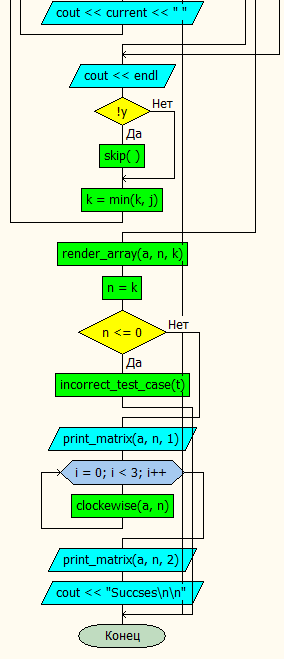
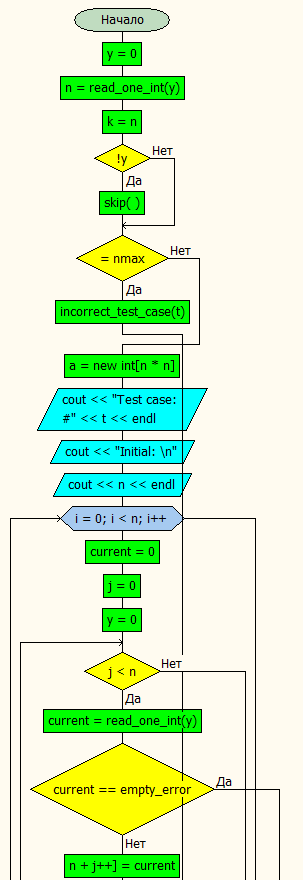
* void render\_array



* void skip



* void solve



# Текст программы

#include <iostream>

#include <fstream>

using namespace std;

ifstream fin("input.txt");

ofstream fout("output.txt");

const int empty\_error = -999999;

const int nmax = 100;

int read\_one\_int(int& y);

void solve(int t);

void incorrect\_test\_case(int t);

void print\_matrix(int a[], int n, int type);

void render\_array(int a[], int n, int k);

void clockewise(int a[], int n);

void skip();

int main() {

int y = 0;

int t = read\_one\_int(y);

if (!y) skip();

if (t == -999999) {

fout << "Empty input!\n";

return 0;

}

for (int i = 1; i <= t; i++) {

solve(i);

}

return 0;

}

void solve(int t) {

int y = 0;

int n = read\_one\_int(y);

int k = n;

if (!y) skip();

if (n <= 0 or n >= nmax) {

incorrect\_test\_case(t);

return;

}

int\* a = new int[n \* n];

fout << "Test case: #" << t << endl;

fout << "Initial: \n";

fout << n << endl;

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

int current = 0;

int j = 0;

y = 0;

while (j < n) {

current = read\_one\_int(y);

if (current == empty\_error) break;

a[i \* n + j++] = current;

fout << current << " ";

}

fout << endl;

if (!y) skip();

k = min(k, j);

}

render\_array(a, n, k);

n = k;

if (n <= 0) {

incorrect\_test\_case(t);

return;

}

print\_matrix(a, n, 1);

for (int i = 0; i < 3; i++)clockewise(a, n);

print\_matrix(a, n, 2);

fout << "Succses\n\n";

return;

}

// 0 0 -> 0 2

// 0 2 -> 2 2

// 2 2 -> 2 0

// 2 0 -> 0 0

void clockewise(int a[], int n) {

for (int i = 0; i < n / 2; i++) {

for (int j = i; j < n - 1 - i; j++) {

int t = a[i \* n + j];

a[i \* n + j] = a[j \* n + n - 1 - i];

a[j \* n + n - 1 - i] = a[(n - 1 - i) \* n + n - 1 - j];

a[(n - 1 - i) \* n + n - 1 - j] = a[(n - 1 - j) \* n + i];

a[(n - 1 - j) \* n + i] = t;

}

}

return;

}

void render\_array(int a[], int n, int k) {

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

for (int j = 0; j < k; j++) a[i \* k + j] = a[i \* n + j];

}

}

int read\_one\_int(int& y) {

/// -999999 = end of line

char c;

int x = 0;

int res = 0;

while (!fin.eof()) {

c = fin.get();

if (c == '\n') {

y = 1;

break;

}

if (c <= '9' and c >= '0') {

x = 1;

res = res \* 10 + (c - '0');

}

else if (x) break;

}

if (!x) res = empty\_error;

return res;

}

void incorrect\_test\_case(int t) {

fout << "Test case number: #" << t << " is incorrect\n\n";

}

void print\_matrix(int a[], int n, int type) {

if (type == 1)fout << "\nPrimary:\n";

else fout << "\nSecondary:\n";

fout << n << endl;

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

for (int j = 0; j < n; j++) fout << a[i \* n + j] << " ";

fout << endl;

}

fout << endl;

}

void skip() {

char c;

while (1) {

c = fin.get();

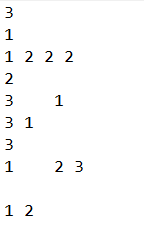
if (c == '\n') break;

}

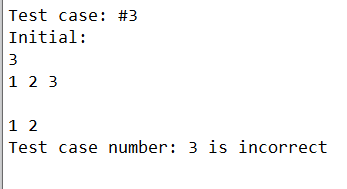
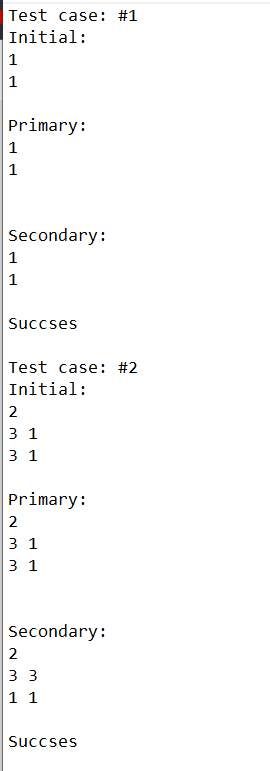
# }

# Контрольные тесты и примеры

Проведем 3 теста



Результаты тестов:



# 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.