**МИНОБРНАУКИ РОССИИ**

**Санкт-Петербургский государственный**

**электротехнический университет**

**«ЛЭТИ» им. В.И. Ульянова (Ленина)**

**Кафедра САПР**

отчет

**по лабораторной работе №5**

**по дисциплине «Программирование»**

Тема: Программа на С++ поворачивающая таблицу

|  |  |  |
| --- | --- | --- |
| Студент гр. 4352 |  | Редкокаша Р. Ю. |
| Преподаватель |  | Калмычков В. А. |

Санкт-Петербург

2024

**Оглавление**

[1 Исходная формулировка задания 3](#_Toc183064633)

[2 Определение неясностей 3](#_Toc183064634)

[3 Математическая постановка 3](#_Toc183064635)

[4 Устройство файла 3](#_Toc183064636)

[5 Контрольный пример 3](#_Toc183064637)

[6 Ограничения 3](#_Toc183064638)

[7 Организация интерфейса пользователя 3](#_Toc183064639)

[8 Реал изация ввода-вывода 4](#_Toc183064640)

[9 Внутреннее представление данных 4](#_Toc183064641)

[10 Описание алгоритма 4](#_Toc183064642)

[11 Тексты программ 6](#_Toc183064643)

[12 Результаты работы программ 9](#_Toc183064644)

[Выводы 9](#_Toc183064645)

**Цель работы.**

Написать программу, поворачивающая элементы таблицы по часовой стрелке вокруг центра на 90 градусов. Реализовать вывод таблицы в файл и консоль.

# Исходная формулировка задания

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

# Определение неясностей

в начале таблицы указывается её размер. Отрицательный размер принять за 0. Размер больший размера файла принять равным размеру файла. реальная ширина таблицы определяется минимальной длинной строки или столбца.

# Математическая постановка

Дано: таблица с вещественными значениями.

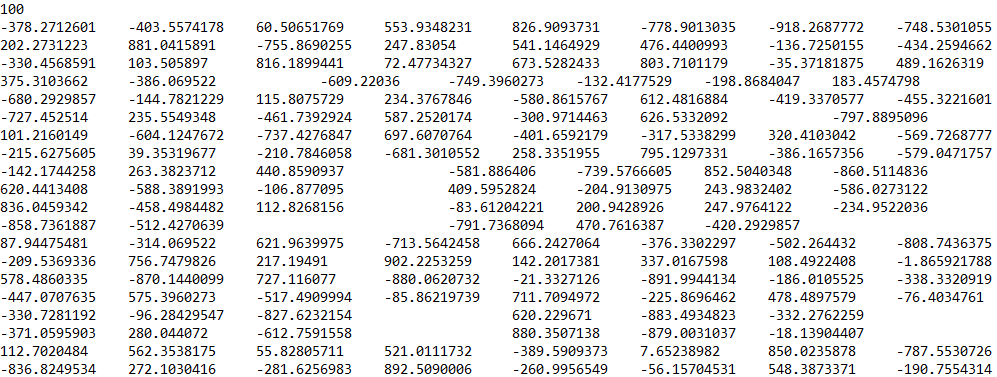
Нужно повернуть элементы по часовой стрелке на 90 градусов.

Математических преобразований производить не требуется.

# Устройство файла

В начале файла указывается размер. Дальше в файле находятся числа и символы разделители.

# Контрольный пример



# Ограничения

Вводимые данные ограничены ёмкостью double.

# Организация интерфейса пользователя

Данные отображаются в программной консоли и файле в формате текста и знаков. Для чисел отображается 7 знаков после запятой.

|  |  |  |
| --- | --- | --- |
| **id** | **Файл** | **Текст** |
| E1 | Log, Error | Файл не помещён в главную директорию. |
| E2 | Файл пуст. |
| E3 | Не указан размер. |
| E4 | Указанный размер меньше или равен нулю. |
| E5 | Размер больше реального. Было приравнено к реальному. |
| O1 | Log | Ввод: |
| O2 | Log | Итог: |
| O3 | Log, FileOut |  |

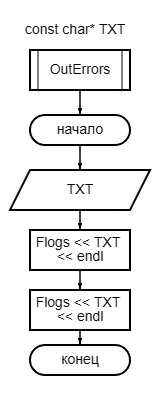
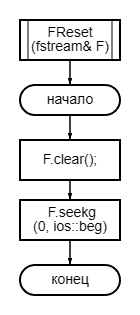
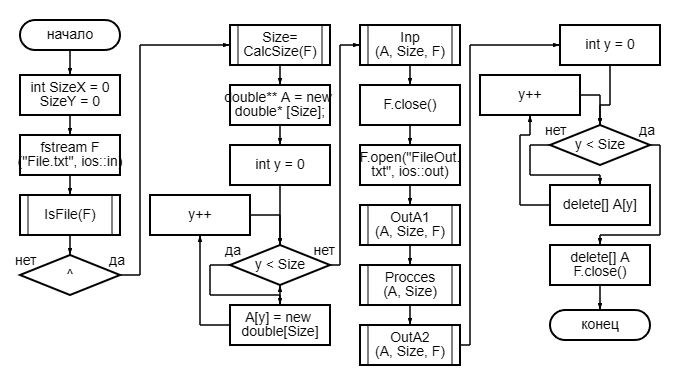
# Реализация ввода-вывода

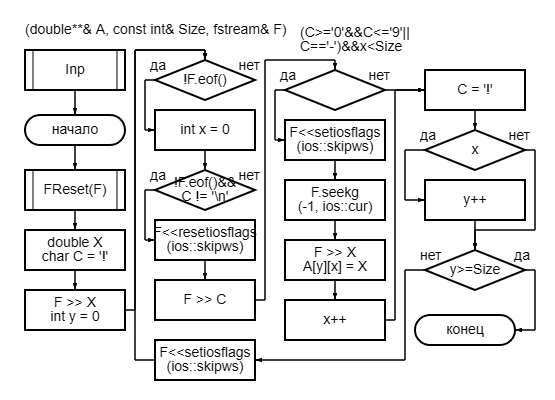
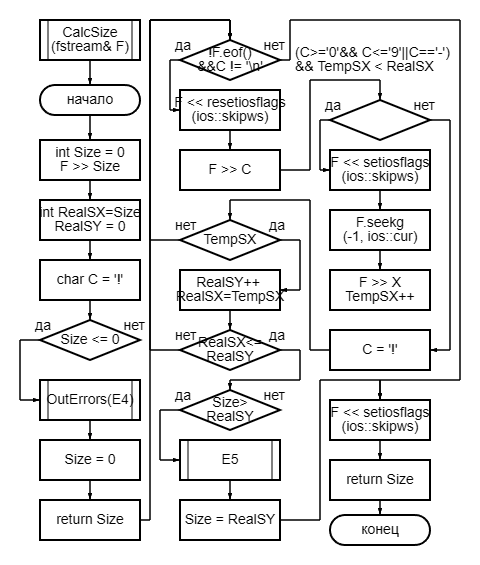
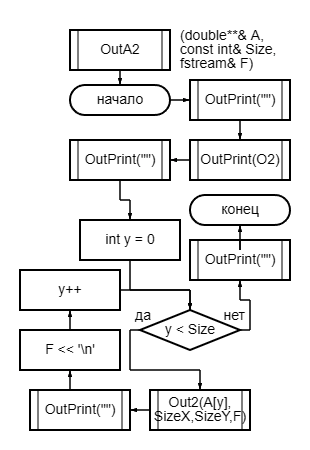
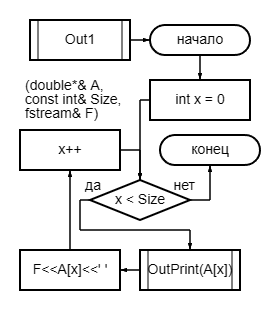
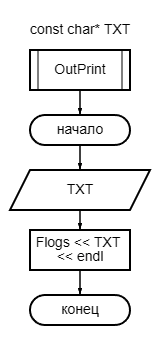
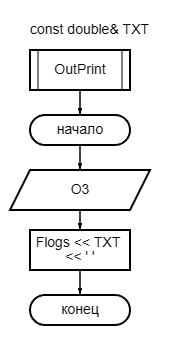
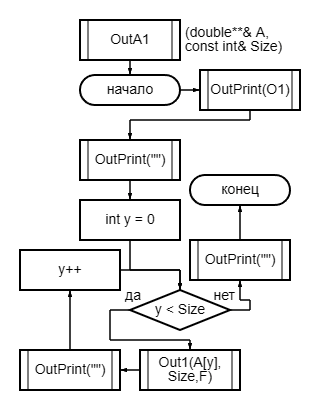
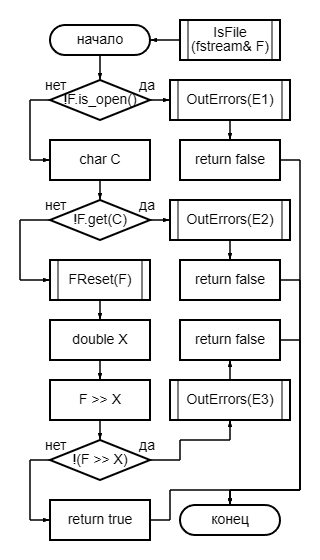
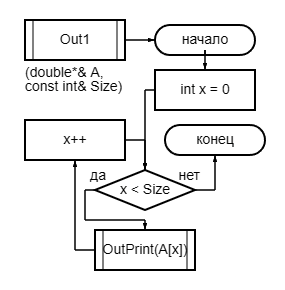
|  |  |  |  |
| --- | --- | --- | --- |
| Библиотека | Ввод | Вывод | Дробная часть |
| iostream | cin | cout |  |
| iomanip |  |  | cout.setf(ios::fixed);  cout.precision(7); |
| fstream | ios::in >> | ios::out << |  |

# Внутреннее представление данных

|  |  |  |
| --- | --- | --- |
| **Наименование** | **Тип** | **Назначение** |
| Size | int | Хранение размера |
| RealSX, RealSY | Хранение настоящего размера (временно) |
| TempSX | Хранение временного размера для оси X |
| X | Счёт значений из файла |
| С | char | Счёт символа из файла |
| A | double\*\* | Массив из файла |
| FLog, FOut, FError | fstream | Доступ к файлам для взаимодействия с пользователем |

# Описание алгоритма

Программа открывает файл. Считает размерность максимально возможной квадратной матрицы и сравнивает её с заявленной. Записывает требуемую или возможную матрицу в массив, выводит его. Поворачивает матрицу на 90 градусов. Выводит итог. Вывод производится в файлы и консоль.



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | параметры | | | Внешние изменения |
| Входные | Выходные | Модифицируемые |  |
| FReset |  |  | fstream& F |  |
| OutPrint | const char\* TXT или  const double& TXT |  |  | Вывод TXT в файл |
| OutErrors |  |  | То же, но в файл Errors |
| IsFile | fstream& F |  |  |  |
| CalcSize |  | Size | fstream& F |  |
| Inp | const int&  Size |  | double\*\*& A  fstream& F |  |
| OutA1 |  |  |
| OutA2 |  | Вывод в файл с таблицей |
| Out2 |  | double\*& A  fstream& F |
| Out1 |  |  |

|  |  |
| --- | --- |
|  | Описание |
| FReset | Получает ссылку на файл (*F*), переносить курсор в начало и очищает состояние файла |
| OutPrint | Получает на вход указатель типа char (*TXT*) и выводит его в консоль, файл “*log.txt*” |
| OutErrors | Получает на вход указатель типа char (*TXT*) и выводит его в консоль, файл “*log.txt*” и “*errors.txt*” |
| IsFile | Получает на вход ссылку на файл (*F*) и проверяет его «правильность» |
| CalcSize | Получает на вход ссылку на файл (*F*) и считает размер таблицы |
| Inp | Получает на вход ссылку на указатель на указатель типа *double* (*A*) – 2d динамический массив, постоянную ссылку на размер (*Size*), ссылку на файл (*F*) и заносит в динамический массив данные из файла. |
| OutA1 | Получает на вход ссылку на указатель на указатель типа *double* (*A*) – 2d динамический массив, постоянную ссылку на размер (*Size*) и выводит данные через функцию *OutPrint* |
| OutA2 | Получает на вход ссылку на указатель типа *double* (*A*) – 1d динамический массив, постоянную ссылку на размер (*Size*) и выводит его значения через функцию *OutPrint* |
| Out1 | Получает на вход ссылку на указатель типа *double* (*A*) – 2d динамический массив, постоянную ссылку на размер (*Size*), ссылку на файл (*F*) и выводит данные через функцию *OutPrint* и в файл “*FileOut.txt*” |
| Out2 | Получает на вход ссылку на указатель типа *double* (*A*) – 1d динамический массив, постоянную ссылку на размер (*Size*), ссылку на файл (*F*) и выводит его значения через функцию *OutPrint* и в файл “*FileOut.txt*” |

# Тексты программ

/\*я—»√ћјя –”“ќ…я—»√ћј

\* Задание:

\* Задана квадратная таблица размера N N. Преобразовать ее, осуществив поворот

\* элементов вокруг ее центра на 90 по часовой стрелке.

\* Автор: Редкокаша Роман Юрьевич Версия: 4.1

\* Группа: 4352

\* Даты: начало: 07.11.2024 - конец: --.11.2024

\*/

#include "iostream"

#include "fstream"

#include "iomanip"

using namespace std;

fstream Flogs("Logs.txt", ios::out), Fe("Errors.txt", ios::out);

void FReset(fstream& F) {

F.clear();

F.seekg(0, ios::beg);

return;}

void OutPrint(const char\* TXT) {

cout << TXT << endl;

Flogs << TXT << endl;}

void OutPrint(const double& TXT) {

cout << TXT << ' ';

Flogs << TXT << ' ';}

void OutErrors(const char\* TXT) {

cout << TXT << endl;

Flogs << TXT << endl;

Fe << TXT << endl;}

bool IsFile(fstream& F) {

if (!F.is\_open()) {

OutErrors("Файл не помещён в главную директорию."); //E1

return false; }

char C;

if (!F.get(C)) {

OutErrors("Файл пуст."); //E2

return false; }

FReset(F);

double X;

F >> X;

if (!(F >> X)) {

OutErrors("Не указан размер."); //E3

return false; }

return true;}

int CalcSize(fstream& F) {

FReset(F);

int Size = 0;

F >> Size;

int RealSX = Size, RealSY = 0;

char C = '!';

if (Size <= 0) {

OutErrors("Указанный размер меньше или равен нулю."); // E4

Size = 0;

return Size; }

while (!F.eof()) {

double X;

int TempSX = 0;

while (!F.eof() && C != '\n') {

F << resetiosflags(ios::skipws);

F >> C;

if ((C >= '0' && C <= '9' || C == '-') && TempSX < RealSX) {

F >> setiosflags(ios::skipws);

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

F >> X;

TempSX++; } }

C = '!';

if (TempSX) {

RealSY++;

RealSX = TempSX;

if (RealSX <= RealSY) break; } }

if (Size > RealSY) {

OutErrors("Размер больше реального. Было приравнено к реальному."); //E5

Size = RealSY; }

F >> setiosflags(ios::skipws);

return Size;}

void Inp(double\*\*& A, const int& Size, fstream& F) {

FReset(F);

double X;

char C = '!';

F >> X;

int y = 0;

while (!F.eof()) {

int x = 0;

while (!F.eof() && C != '\n') {

F << resetiosflags(ios::skipws);

F >> C;

if ((C >= '0' && C <= '9' || C == '-') && x < Size) {

F >> setiosflags(ios::skipws);

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

F >> X;

A[y][x] = X;

x++; } }

C = '!';

if (x) y++;

if (y >= Size) break; }

F >> setiosflags(ios::skipws);

return;}

void Out1(double\*& A, const int& Size) {

for (int x = 0; x < Size; x++) {

OutPrint(A[x]); }

return;}

void OutA1(double\*\*& A, const int& Size) {

OutPrint("Ввод:");

OutPrint("");

for (int y = 0; y < Size; y++) {

Out1(A[y], Size);

OutPrint(""); }

return;}

void Out2(double\*& A, const int& Size, fstream& F) {

for (int x = 0; x < Size; x++) {

OutPrint(A[x]);

F << A[x] << ' '; }

return;}

void OutA2(double\*\*& A, const int& Size, fstream& F) {

OutPrint("");

OutPrint("Итог:");

OutPrint("");

for (int y = 0; y < Size; y++) {

Out2(A[y], Size, F);

OutPrint("");

F << '\n'; }

return;}

void Procces(double\*\*& A, int& Size) {

fstream F("FileOut.txt", ios::out);

int n = Size - 1, n2 = Size / 2 + Size % 2;

double Temp;

for (int y = 0; y < n2; y++)

for (int x = y; x < n - y; x++) {

Temp = A[y][x];

A[y][x] = A[n - x][y];

A[n - x][y] = A[n - y][n - x];

A[n - y][n - x] = A[x][n - y];

A[x][n - y] = Temp; }

return;}

int main() {

setlocale(LC\_ALL, "RUSSIAN");

cout.precision(7);

cout.setf(ios::fixed);

int Size;

fstream F("File.txt", ios::in);

if (IsFile(F)) {

Size = CalcSize(F);

double\*\* A = new double\* [Size];

for (int y = 0; y < Size; y++)

A[y] = new double[Size];

Inp(A, Size, F);

F.close();

F.open("FileOut.txt", ios::out);

OutA1(A, Size);

Procces(A, Size);

OutA2(A, Size, F);

for (int y = 0; y < Size; y++)

delete[] A[y];

delete[] A;

F.close(); }

return 0;}

# Результаты работы программ

|  |  |
| --- | --- |
| Заявленный размер больше (максимальной ширины):  Размер больше реального. Было приравнино к реальному.  Ввод:  -378.271 -403.557 60.5065 553.935  202.273 881.042 -755.869 247.831  -330.457 103.506 816.19 72.4773  375.31 -386.07 -609.22 -749.396  Итог:  375.31 -330.457 202.273 -378.271  -386.07 103.506 881.042 -403.557  -609.22 816.19 -755.869 60.5065  -749.396 72.4773 247.831 553.935 | Больше (максимальной высоты):  Размер больше реального. Было приравнино к реальному.  Ввод:  -378.271 -403.557 60.5065 553.935 826.909  202.273 881.042 -755.869 247.831 541.146  -330.457 103.506 816.19 72.4773 673.528  375.31 -386.07 -609.22 -749.396 -132.418  -680.293 -144.782 115.808 234.377 -580.862  Итог:  -680.293 375.31 -330.457 202.273 -378.271  -144.782 -386.07 103.506 881.042 -403.557  115.808 -609.22 816.19 -755.869 60.5065  234.377 -749.396 72.4773 247.831 553.935  -580.862 -132.418 673.528 541.146 826.909 |
| Меньше:  Ввод:  -378.271 -403.557 60.5065 553.935 826.909  202.273 881.042 -755.869 247.831 541.146  -330.457 103.506 816.19 72.4773 673.528  375.31 -386.07 -609.22 -749.396 -132.418  -680.293 -144.782 115.808 234.377 -580.862  Итог:  -680.293 375.31 -330.457 202.273 -378.271  -144.782 -386.07 103.506 881.042 -403.557  115.808 -609.22 816.19 -755.869 60.5065  234.377 -749.396 72.4773 247.831 553.935  -580.862 -132.418 673.528 541.146 826.909 | Size = 1  Ввод:  -378.271  Итог:  -378.271 |

Выводы.

Написал программу, считывающую максимально возможную квадратную таблицу из файла и поворачивающую её на 90 градусов по часовой стрелке. Научился пользоваться двухмерными массивами и функциями.