

**МИНОБРНАУКИ РОССИИ**  
**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ**  
**УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ**  
**УНИВЕРСИТЕТ им. В.Г. ШУХОВА»**  
**(БГТУ им. В.Г. Шухова)**

Институт ЭИТУС  
Кафедра информационных технологий

**Отчет о прохождении учебной практики**

База практики: кафедра информационных технологий

Выполнил:  
студент группы ИТ-192  
Жулега И.Д.

Проверил:  
старший преподаватель  
Имайкина Л.Е.

Белгород 2020

## Оглавление

1. Общая характеристика организации .....	2
2. Задачи практики .....	3
Работа с системой контроля версий GIT.....	3
Работа с двумерными массивами, работа с файлами, применение рекурсивных функций на языке C/C++ .....	8
Обработка динамических массивов на языке C/C++. ....	17
3. Заключение .....	26

## 1. Общая характеристика организации

### **Информация о кафедре Информационных технологий:**

Кафедра информационных технологий входит в состав института информационных технологий и управляющих систем, организована приказом ректора от 29.06.1999 г. Первоначальной задачей кафедры являлось проведение занятий по информатике и смежным дисциплинам у неспециалистов.

С 2003 года кафедра становится выпускающей и ведет подготовку по направлению:

- 09.03.02 - Информационные системы и технологии.

С 2014 года открыта подготовка по направлению:

- 09.03.03 - Прикладная информатика (в бизнесе).

С 2016 года началась подготовка магистров по направлению:

- 09.04.02 - Информационные системы и технологии.

Будущие специалисты по информационным технологиям обучаются с применением современных информационных технологий. Компьютерные залы оборудованы новейшими компьютерами, видеопроекторными средствами. Широко применяются автоматизированное тестирование, электронные учебники, выполнение виртуальных лабораторных работ. Посредством сети Интернет студенты могут из своего дома или общежития получать удаленный доступ к учебным материалам, расписанию занятий. При кафедре функционирует учебно-методический центр информационно-технического обеспечения образования, который проводит работу по созданию и обеспечению применения в университете прогрессивных информационно-технических средств обучения (ИТСО), компьютерных информационных технологий, мультимедийных и других технических средств, способствующих совершенствованию учебного процесса.

### **Информация о ПК:**

ОС – Windows 10

Процессор: Intel Core i5-3470 3.20 GHz

ОЗУ: 8 Гб

Тип системы: 64 – разрядная

Монитор: ASUS (старый) формат 5:4, 1280x1024 точек

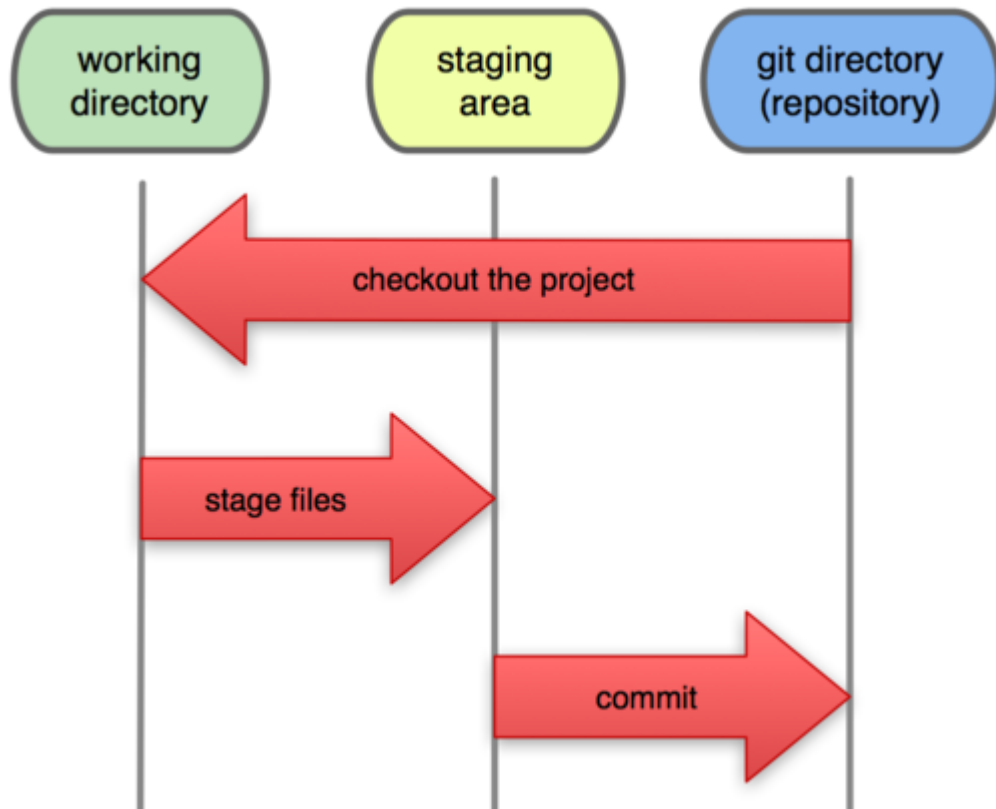
ПО: MS Office Word 2016, MS Office Visio 2016, Visual Studio 2019

## 2. Задачи практики

### Работа с системой контроля версий GIT

#### 2.1.1. Что такое GIT?

GIT – система контроля версий (СКВ). Эта система, регистрирующая изменения в одном или нескольких файлах с тем, чтобы в дальнейшем была возможность вернуться к определённым старым версиям этих файлов.



**Рис.1 Б-с того как работает GIT**

**GIT directory** - каталог GIT - это место, где GIT хранит метаданные и базу данных объектов вашего проекта. Это наиболее важная часть GIT, и именно она копируется, когда вы клонируете репозиторий с другого компьютера.

**Working directory** - рабочий каталог - это извлечённая из базы копия определённой версии проекта. Эти файлы достаются из сжатой базы данных в каталоге GIT и помещаются на диск для того, чтобы вы их просматривали и редактировали.

**Staging area** - область подготовленных файлов — это обычный файл, обычно хранящийся в каталоге GIT, который содержит информацию о том, что должно войти в следующий коммит. Иногда его называют индексом (*index*), но в последнее время становится стандартом называть его областью подготовленных файлов.

### 2.1.2. Что нам даёт СКВ?

СКВ даёт множество полезных функций. В первую очередь эта система позволит нам сделать «откат», привести отдельные файлы к предыдущему виду или даже весь проект. В следующую же очередь, в ней есть возможность просматривать происходящие изменения файлов – проекта, и так же, кто и когда внёс ошибку в код.

### 2.1.3. Начало работы с GIT

Для того чтобы начать работать с GIT я скачал само приложение с официального сайта, открыл удобную рабочую область GIT (ПКМ → GIT Bush Here) либо в самой рабочей папке либо в другом месте компьютера открыть рабочую область и при помощи `$ cd <Диск>:\<Путь до папки>\<Папка>`. Пробовал оба варианта, первый конечно же проще.

После того как оказался в исходной папке я создал сам проект, для этого написал команду `$ git init` (после этого в исходной папке создаётся новая папка (.git) в которой и будет храниться ваш коммит и все данные что вы делали).

Для того что бы не забывать операнды GIT, есть удобный способ увидеть их все, для этого написал `$ git help` после чего появится текст с операндом и что он делает (правда всё на английском к сожалению).

После того как я инициализировал рабочую область - репозиторий я смог работать с GIT полноценно.

### 2.1.4. Работа с GIT

После того как я добавил файлы в репозиторий, отредактировал или же, даже просто удалил файлы нужно добавить изменения в гит-проект, для этого написал команду `$ git add <название папки или файла>` или `< . >` (при написании (.) добавляются все новые или изменённые файлы). Мне удобней пользоваться `$ git add .` т.к. добавление отдельных файлов не только занимает время, но и если вы хотите не добавлять некоторые файлы в репозиторий то просто «за игнорируйте» его, об этом я написал дальше. Проверил работоспособность этой функции. Его (файл с игнорированием) можете посмотреть в моём проекте ссылка в конце темы про GIT.

Для того, чтобы удалить файл нужно написать `git rm --cached <название файла>`. Но я использовал для игнорирования некоторых файлов текстовый документ, для этого зашёл в него записываем что игнорируем (см. пример Рис.2) и перевёл в тип .gitignore. (Знак # игнорирует что находится в строке после него, т.е. можно сделать пояснения почему вы хотите игнорировать данный файл).

```
# Исключить все файлы с расширение .a
*.a

# Но отслеживать файл lib.a даже если он подпадает под исключение выше
!lib.a

# Исключить файл TODO в корневой директории, но не файл в subdir/TODD
/TODD

# Игнорировать все файлы в директории build/
build/

# Игнорировать файл doc/notes.txt, но не файл doc/server/arch.txt
doc/*.txt

# Игнорировать все .txt файлы в директории doc/
doc/**/*.txt
```

**Рис.2 Пример**

Написал `git status` – для просмотра статуса репозитория.

Зарегистрировал себя в репозитории при помощи команды `git config - -global user.<name>или<@mail>` для привязки акк. с сервер GIT (без этого вы не сможете коммитить )

```
$ git config --global user.name "John Doe"
$ git config --global user.email johndoe@example.com
```

**Рис.3**

Связал аккаунт и репозиторий github при помощи команды `$ git remote add origin <ссылка на проект в github>`.

Загрузил проект на github при помощи команды `$ git push -u origin master`; `<origin master>` можно будет в дальнейшем писать только `$ git push -u` это будет загружать ваш коммит в сам сервер GitHub.

Зашёл в другую папку и попробовал клонировать (копировать) свой проект с сайта на компьютер при помощи команды `$ git clone <ссылка на репозиторий с сайта>` - скачивает.

Доказательство работы с GIT'ом (см. Рис.4, Рис.5, Рис.6)

```
MINGW64/d/project
320\270\320\272 \321\203\321\207\320\265\320\261\320\275\320\276\320\271 \320\27
7\321\200\320\260\320\272\321\202\320\270\320\272\320\270 09.03.02.docx"
new file: "Adeptus Practicus\320\237\321\200\320\260\320\272\321\202\
320\270\320\272\320\260.DOCX"

User@DESKTOP-15Q50C4 MINGW64 /d/project (master)
$ git commit -m 'Добавил заполненный дневник по практике'
[master 95d529e] 2 files changed, 0 insertions(+), 0 deletions(-)
 create mode 100644 "Adeptus Practicus\320\224\320\275\320\265\320\262\320\275\
320\270\320\272 \321\203\321\207\320\265\320\261\320\275\320\276\320\271 \320\27
7\321\200\320\260\320\272\321\202\320\270\320\272\320\270 09.03.02.docx"
 create mode 100644 "Adeptus Practicus\320\237\321\200\320\260\320\272\321\202\
320\270\320\272\320\260.DOCX"

User@DESKTOP-15Q50C4 MINGW64 /d/project (master)
$ git log
commit 95d529e0325b17505d44e5f365bbe3c58cab191e (HEAD -> master)
Author: Zhulega Igor <zhulega2001@mail.ru>
Date: Wed Jul 1 21:18:50 2020 +0300

    Добавил заполненный дневник по практике

commit eb79aaf2a2658480b75bdeb3f2b24be1df679a17 (origin/master)
Author: Zhulega Igor <zhulega2001@mail.ru>
Date: Tue Jun 30 21:44:48 2020 +0300

    Добавил лабораторные работы нового варианта (9-ю вместе с отчётом), обе с б-
    с

commit 680e506bfdcbe6fb57b176436c01df8655cc94ab
Author: Zhulega Igor <zhulega2001@mail.ru>
Date: Wed Jun 24 16:00:33 2020 +0300

    Изменил пустую папку pr на proshlie rab и ввместил туда прошлые свои лаборато
    рные работы по инф. 8-варианта

commit 3741a602eec54e83442f585e75bbf237bbb04603
Author: Zhulega Igor <zhulega2001@mail.ru>
Date: Tue Jun 23 19:30:56 2020 +0300
:...skipping...
commit 95d529e0325b17505d44e5f365bbe3c58cab191e (HEAD -> master)
Author: Zhulega Igor <zhulega2001@mail.ru>
Date: Wed Jul 1 21:18:50 2020 +0300

    Добавил заполненный дневник по практике

commit eb79aaf2a2658480b75bdeb3f2b24be1df679a17 (origin/master)
Author: Zhulega Igor <zhulega2001@mail.ru>
Date: Tue Jun 30 21:44:48 2020 +0300

    Добавил лабораторные работы нового варианта (9-ю вместе с отчётом), обе с б-с

commit 680e506bfdcbe6fb57b176436c01df8655cc94ab
Author: Zhulega Igor <zhulega2001@mail.ru>
Date: Wed Jun 24 16:00:33 2020 +0300

    Изменил пустую папку pr на proshlie rab и ввместил туда прошлые свои лаборато
    рные работы по инф. 8-варианта

commit 3741a602eec54e83442f585e75bbf237bbb04603
Author: Zhulega Igor <zhulega2001@mail.ru>
Date: Tue Jun 23 19:30:56 2020 +0300

    Проба добавление репозитория, и манипуляции связанные с файлами
~
```

Рис.4

## History for ProjectOne / Adeptus Practicus

Commits on Jul 6, 2020

- Отредактировал отчёт и дневник  
Zhulegal committed 8 days ago [a2110b7](#) <>
- Добавил  
Zhulegal committed 8 days ago [6e36c0a](#) <>

Commits on Jul 2, 2020

- Отредактировал помарки в дневнике  
Zhulegal committed 12 days ago [cce574f](#) <>
- Добавил отчёт по практике и дописал недостающие эл. в дневнике практи...  
Zhulegal committed 12 days ago [43ca4c6](#) <>

Commits on Jul 1, 2020

- Добавил заполненный дневник по практике  
Zhulegal committed 13 days ago [95d529e](#) <>

**Рис.5 История одной из папок проекта на сайте (комиты).**

Отредактировал отчёт и дневник [Browse files](#)

master

Zhulegal committed 8 days ago 1 parent 20238ec commit a2110b78ebce122ada0016de2bc1d5fd8e9d7279

Showing 6 changed files with 0 additions and 0 deletions. [Unified](#) [Split](#)

- BIN -162 Bytes Adeptus Practicus/~\$актика.DOCX [Binary file not shown.](#)
- BIN -162 Bytes Adeptus Practicus/~\$евник учебной практики 09.03.02.docx [Binary file not shown.](#)
- BIN +391 Bytes (100%) Adeptus Practicus/Дневник учебной практики 09.03.02.docx [Binary file not shown.](#)
- BIN +214 KB (130%) Adeptus Practicus/Практика.DOCX [Binary file not shown.](#)
- BIN +49 KB lab7/Документ Microsoft Visio.vsdх [Binary file not shown.](#)
- BIN -440 Bytes lab7/Этот компьютер - Ярлык.lnk [Binary file not shown.](#)

0 comments on commit [a2110b7](#) [Lock conversation](#)

[Write](#) [Preview](#) H B I [≡](#) < > [🔗](#) [⋮](#) [⋮](#) [☑](#) [@](#) [🗨](#) [↶](#)

**Рис.6 последнее обновление**

Ссылка на GIT проект (<https://github.com/ZhulegalI/ProjectOne.git>)



## **Работа с двумерными массивами, работа с файлами, применение рекурсивных функций на языке C/C++ .**

### **Постановка задания:**

Выбрать алгоритм, составить его блок-схему и программу для решения выбранного варианта задания. Программа должна по выбору пользователя осуществлять ввод исходной матрицы с клавиатуры или из файла. Для этого программа должна содержать две соответствующие функции, указатель на одну из которых необходимо передавать в функцию для вычисления элементов массива  $X$ . Данная функция должна вызывать через указатель одну из функций ввода элементов матрицы, производить вычисление элементов массива  $X$  в соответствии с заданием и возвращать указатель на этот массив. Кроме того, программа должна содержать функцию для вывода на экран и в файл исходной матрицы и результирующего массива, а также рекурсивную функцию определения в соответствии с заданием величины  $Y$ . В программе не должно быть глобальных переменных.

Вариант 9: Дана матрица  $A(5 \times 5)$ . Определить массив  $X$  из 5 элементов, каждый из которых равен 1, если элементы упорядочены по возрастанию или по убыванию и  $-1$  в противном случае. Определить величину  $Y$ , как среднее арифметическое наибольшего и наименьшего элемента главной диагонали матрицы  $A$ .

### Блок-схемы функций и основного алгоритма:

Функция sfile:

Назначение: заполнение матрицы данными из файла. (см. Рис.7)

Входные параметры: матрица  $mas[n][n]$ .

Выходные параметры: нет.

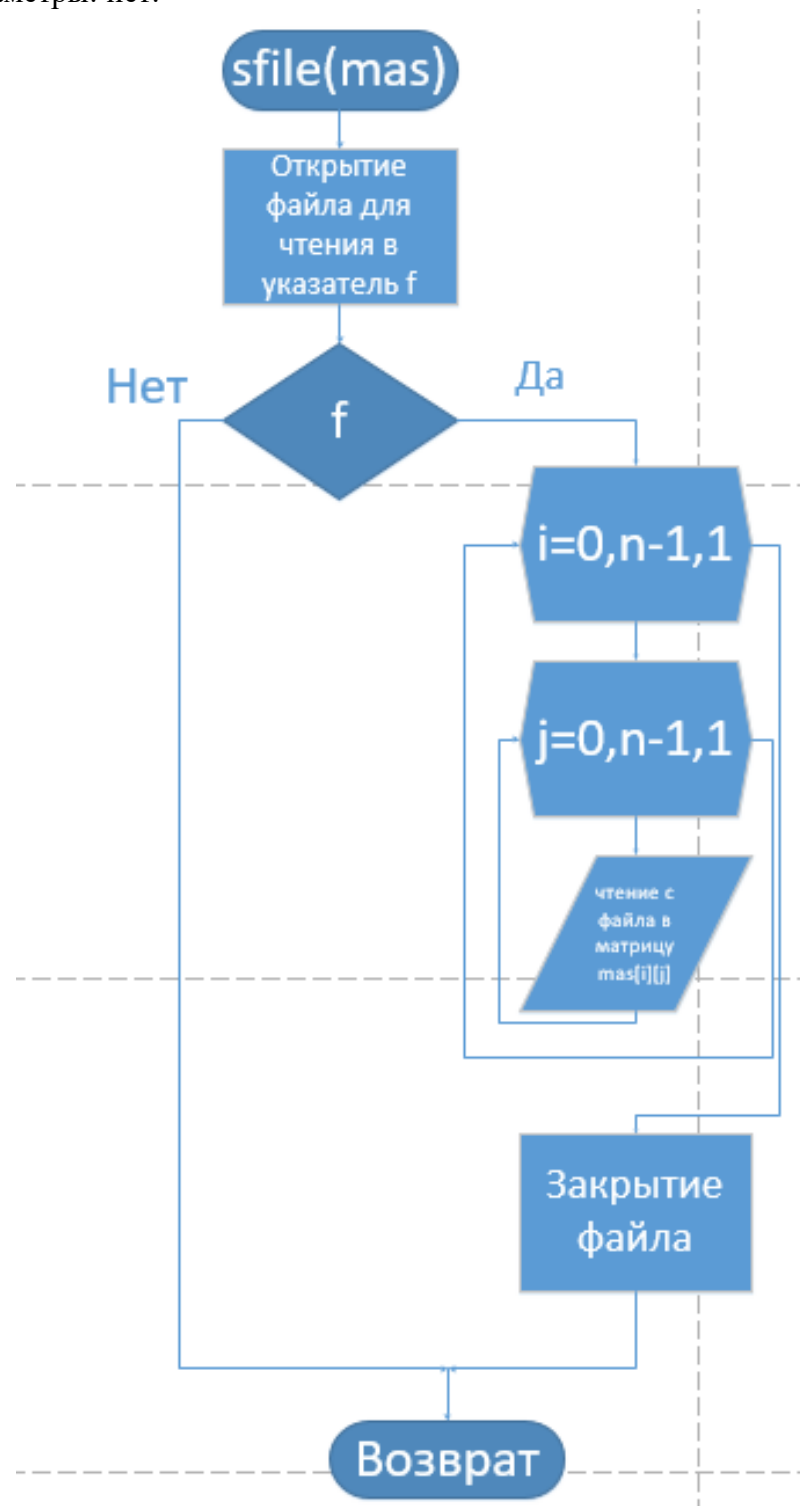


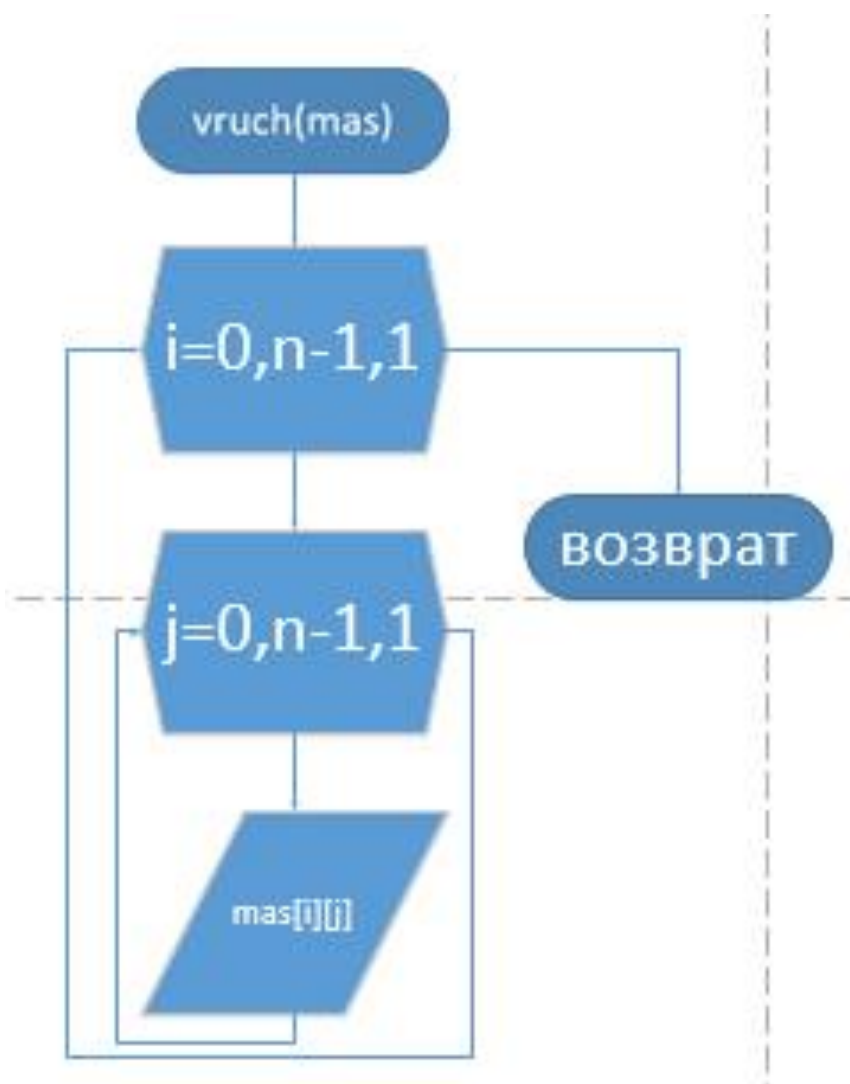
Рис.7 функция заполнения массива с файла

Функция vruch:

Назначение: ввод элементов массива с клавиатуры (см. Рис.8).

Входные параметры: матрица mas[n][n].

Выходные параметры: нет.



**Рис.8** заполнение матрицы с клавиатуры

### Функция vvod\_X:

Назначение: заполнение массива  $X[n]$  (массив  $X$  из 5 элементов, каждый из которых равен 1, если элементы упорядочены по возрастанию или по убыванию и  $-1$  в противном случае) и матрицы  $mas[n][n]$ , матрица заполняется посредством передачи в функцию указателя на функцию, которую выбирает пользователь в основной программе. (см. Рис.9)

Входные параметры: массив  $X[n]$ , матрица  $mas[n][n]$ , указатель  $rfunc$

Выходные параметры: указатель на  $X$

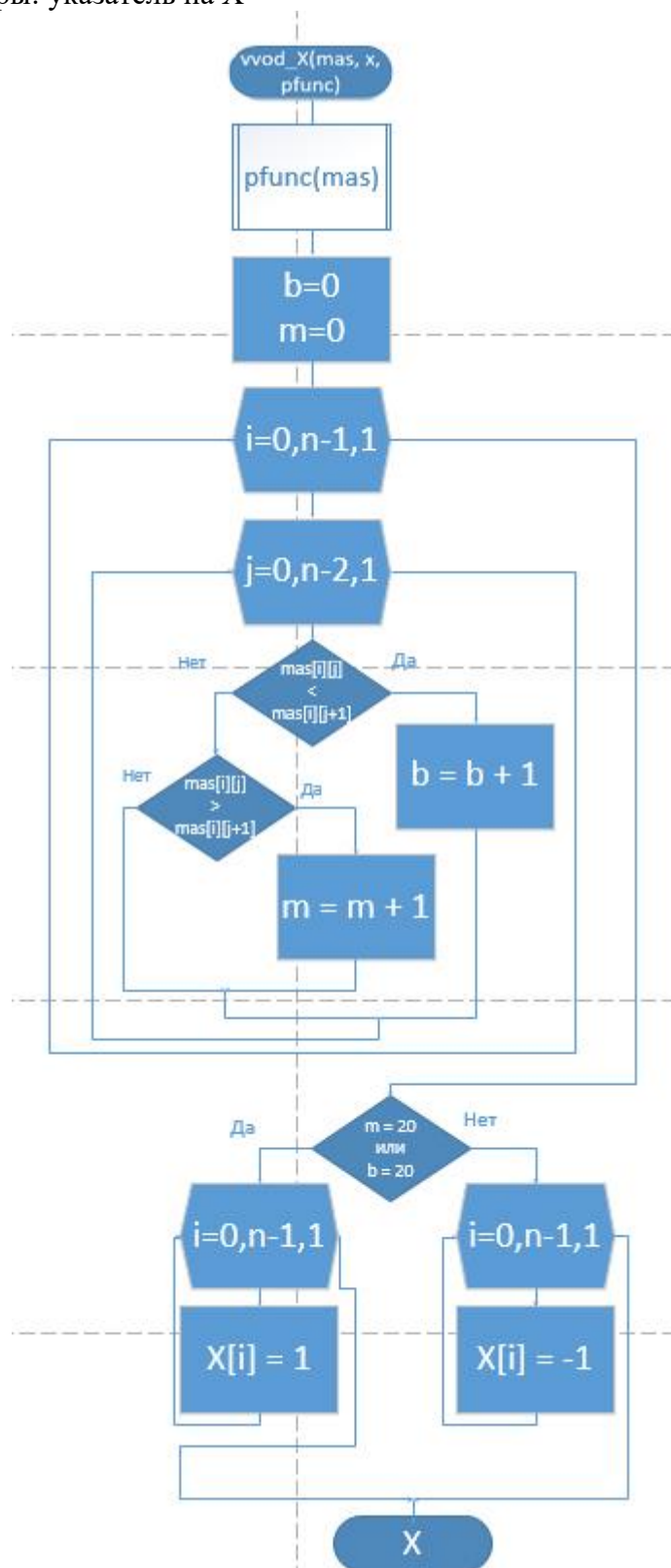


Рис.9 заполнение массива  $X$

Функция out:

Назначение: вывод элементов массива и матрицы, а так же подсчёт  $Y$  и вывод его на экран (см. Рис.10).

Входные параметры: массив  $X[n]$ , матрица  $mas[n][n]$ .

Выходные параметры: нет.

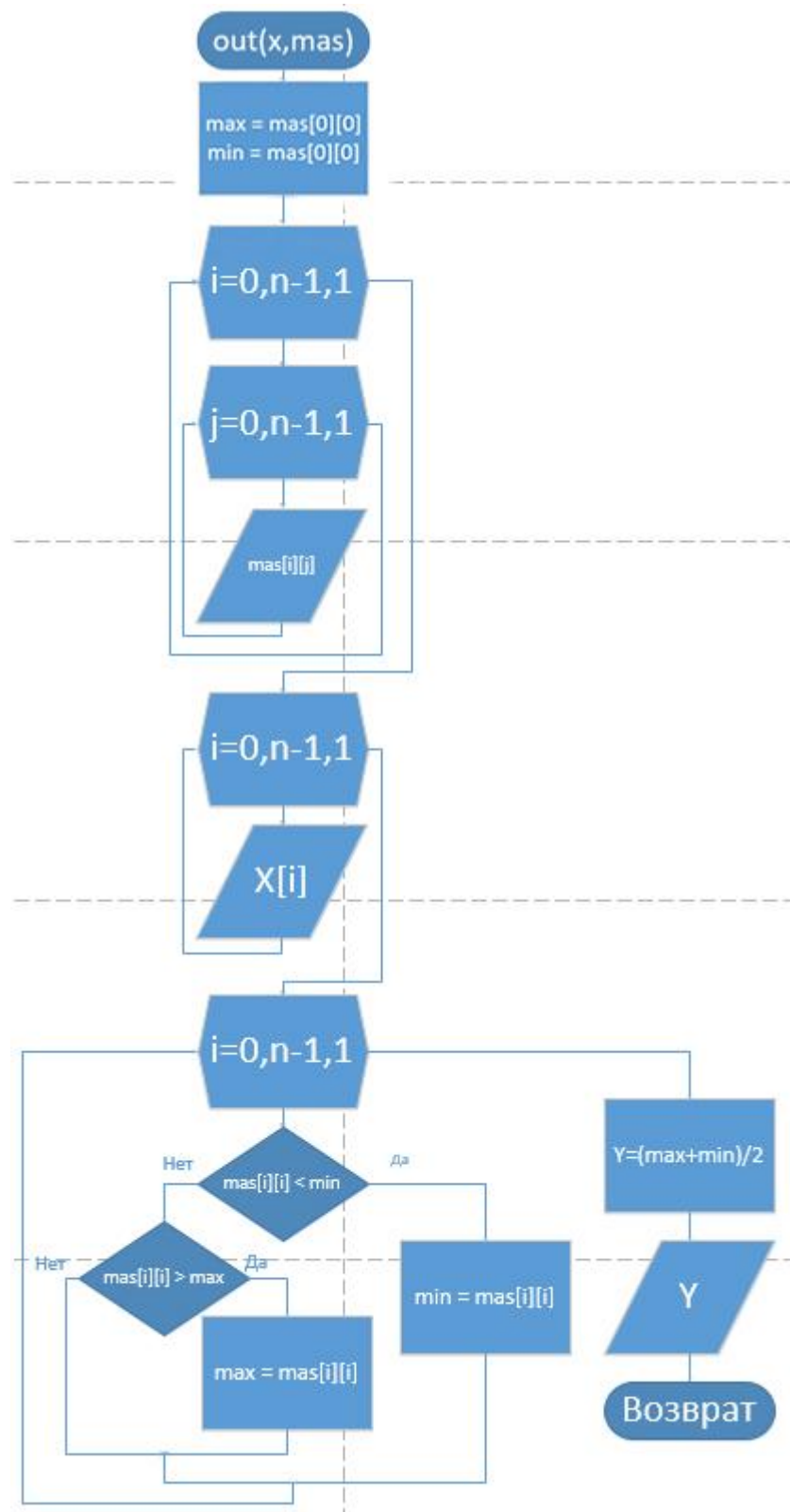


Рис.10 вывод на экран

Основная программа. (см. Рис.11)

Алгоритм: даётся выбор пользователю как заполнить массив. После того как он сделал выбор в указатель на функцию передаётся адрес функции и далее в функции out за место параметра X[n] предаётся функция возвращающая указатель на этот массив (в функции vvod\_X же, в свою очередь идёт заполнение как массива, так и матрицы. В неё передаётся как раз таки указатель на функцию Point\_X) и матрица mas.

Всё выводится на экран массив, матрица и Y.

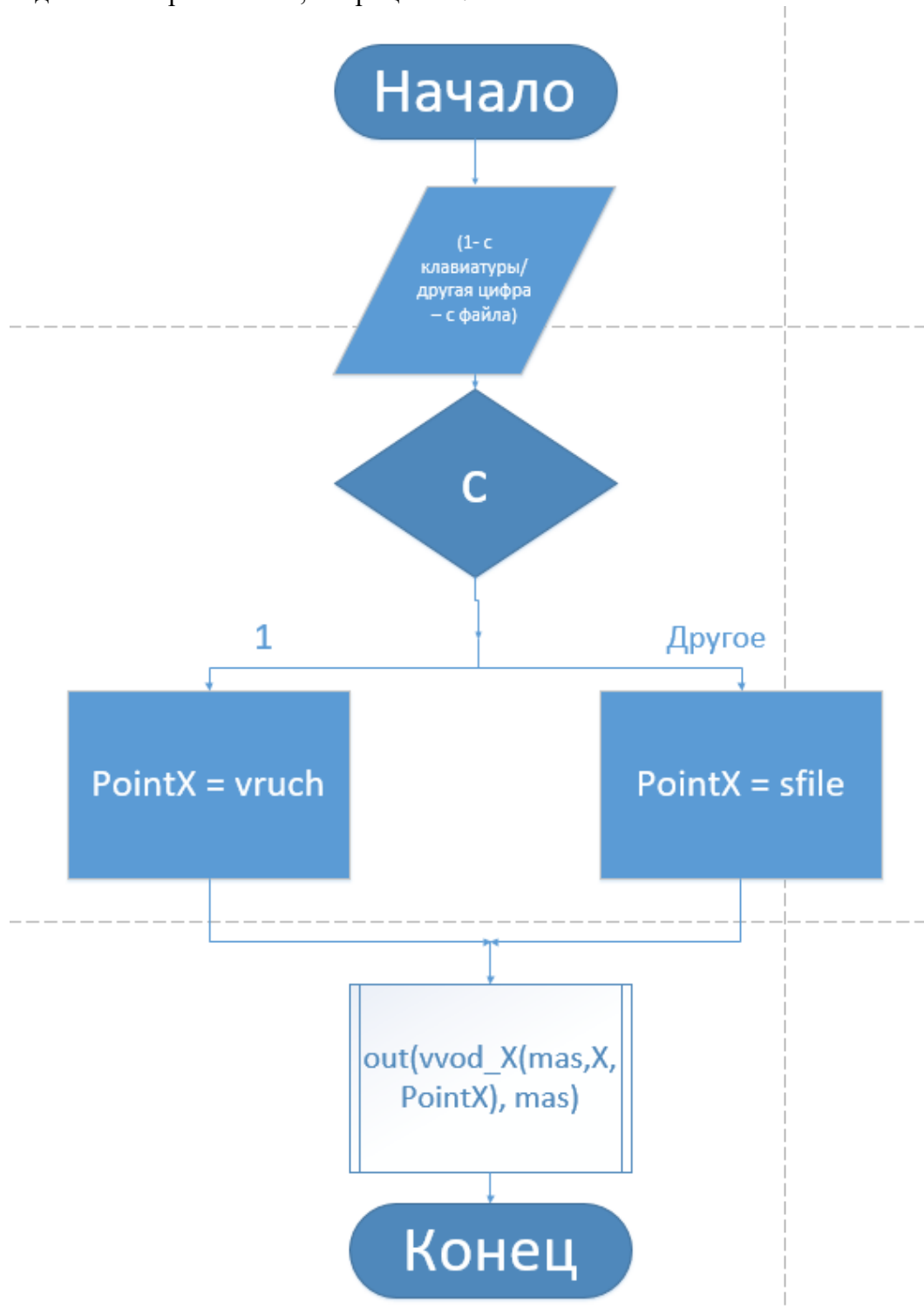


Рис.11 Основная программа

Текст программы:

```

#include <iostream>
#include <locale.h>
#include <stdio.h>
# define N 5

using namespace std;

// Вариант с считыванием из файла
void sfile(int mas[][N]) {
    FILE* f;
    fopen_s(&f, "1.txt", "r");

    if (f) {
        for (int i = 0; i < N; i++) {
            for (int j = 0; j < N; j++) {
                fscanf_s(f, "%d", &mas[i][j]);
            }
        }
        fclose(f);
    }
}

void vruch(int mas[][N]) {
    cout << "Введите эл. матрицы A(5 x 5):" << endl;

    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N; j++) {
            cout << "A[" << i << "][" << j << "]:\t";
            cin >> mas[i][j];
        }
    }
}

int *vvod_X(int mas[][N], int X[N], void (*pfunc) (int masp[][N])) {
    pfunc(mas);

    int b=0, m=0;
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N - 1; j++) {
            if (mas[i][j] < mas[i][j + 1]) b++; // по возрастанию
            else if (mas[i][j] > mas[i][j + 1]) m++; //по убыванию
        }
    }
    if ((m == 20) or (b == 20))
        for (int i = 0; i < N; i++) X[i] = 1;
    else for (int i = 0; i < N; i++) X[i] = -1;

    return X;
}

void out(int X[], int mas[][N]) {
    int i, j, max = -100000, min = 100000 ;

    cout << "Массив A(5 x 5): " << endl;
    for (i = 0; i < N; i++) {

```

```

        for (j = 0; j < N; j++) {
            cout << mas[i][j] << "\t";
        }
        cout << endl;
    }

    cout << "Массив X[5]: " << endl;

    for (i = 0; i < N; i++) cout << X[i] << "\t";

    for (i = 0; i < N; i++) {
        if (mas[i][i] < min) min = mas[i][i];
        else if (mas[i][i] > max) max = mas[i][i];
    }

    cout << endl << "У являющийся средним арифм. главной диагонали равен: " << (max +
min) / 2 << endl;
}

void main() {
    setlocale(LC_CTYPE, "rus");

    int c;

    int mas[N][N];
    int X[N];

    void (*PointX)(int mas[][N]);
    cout << "Вы хотите ввести матрицу в ручную (1-да\\другая цифра - нет)" << endl;

    cin >> c;

    switch(c) {
        case 1: PointX = &vruch; break;

        default: PointX = &sfile;
    }

    out(vvod_X(mas, X, PointX), mas);
}

```



## Тестирование программы:

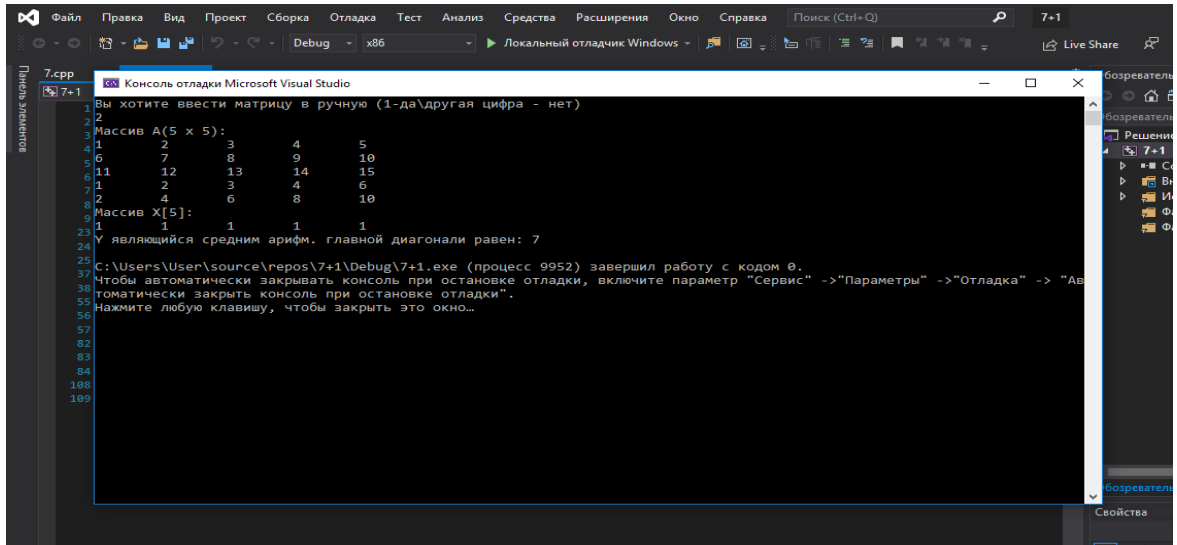


Рис.12 с файла

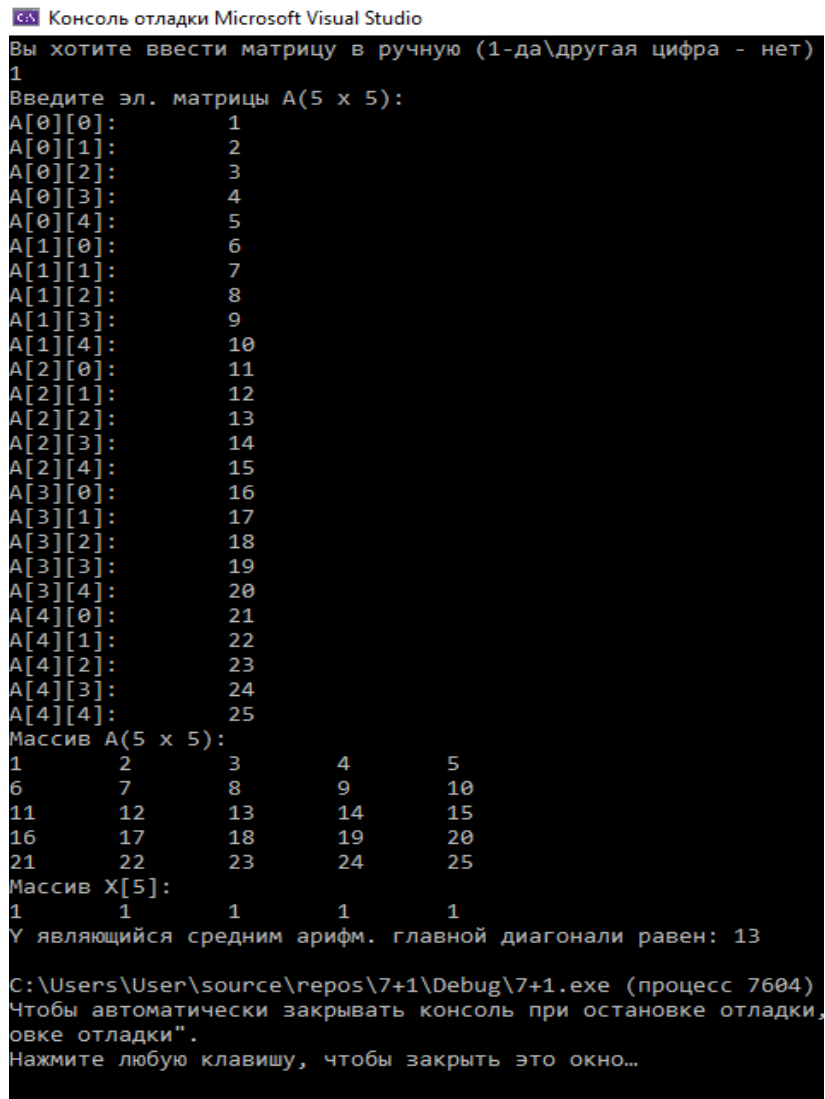


Рис.13 с клавиатуры

## **Обработка динамических массивов на языке C/C++.**

Выбрать алгоритм, составить его блок-схему и программу, выполняющую создание и обработку двумерного динамического массива, в соответствии со своим вариантом задания. Во всех вариантах предполагается, что размерность массива задается на этапе выполнения пользователем. Элементы матрицы вводятся с клавиатуры. На экран выводится исходная матрица и результаты работы программы.

Вариант 9: Дана матрица  $A(n \times n)$ , состоящая из ненулевых элементов. Упорядочить по возрастанию элементы каждого из столбцов. Если имеются несколько столбцов, содержащих одинаковый набор элементов, то оставить без изменений только столбец с младшим индексом, заполнив остальные нулями.

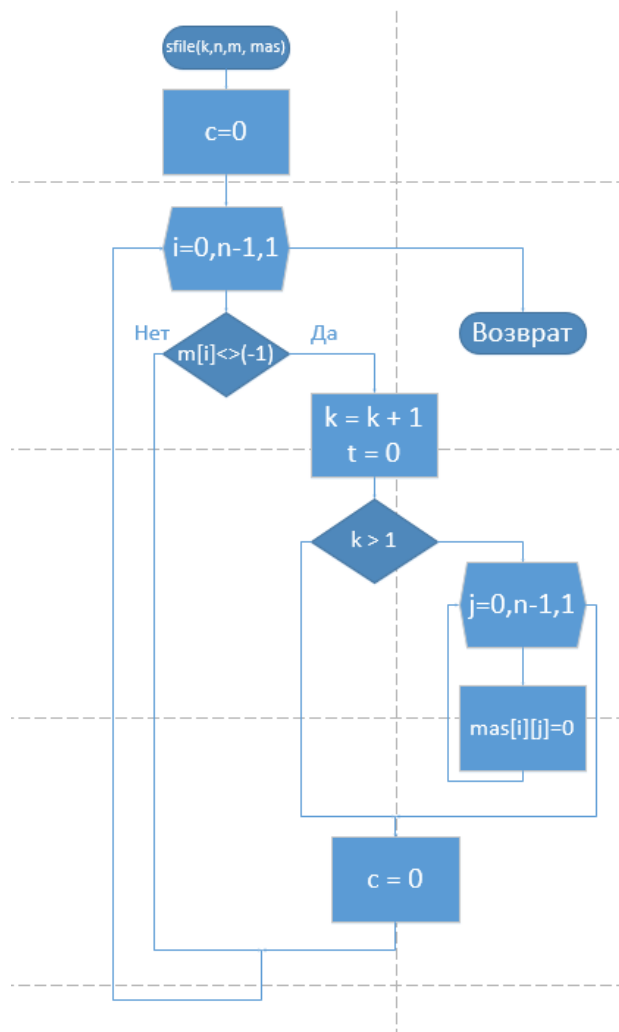
### **Блок-схемы используемых функций и основного алгоритма:**

#### Функция sfile:

Назначение: обнуление столбцов матрицы mas. (см. Рис.14)

Входные данные: k – счётчик столбцов с повт. эл., n - константа, m – динамический массив, mas - матрица

Выходные данные: нет



**Рис.14 Функция обнуления столбцов**

Функция raz:

Назначение: упорядочивание элементов (в столбцах) матрицы mas по возрастанию. (см.

**Ошибка! Источник ссылки не найден.**)

Входные данные: n - константа, mas - матрица

Выходные данные: нет

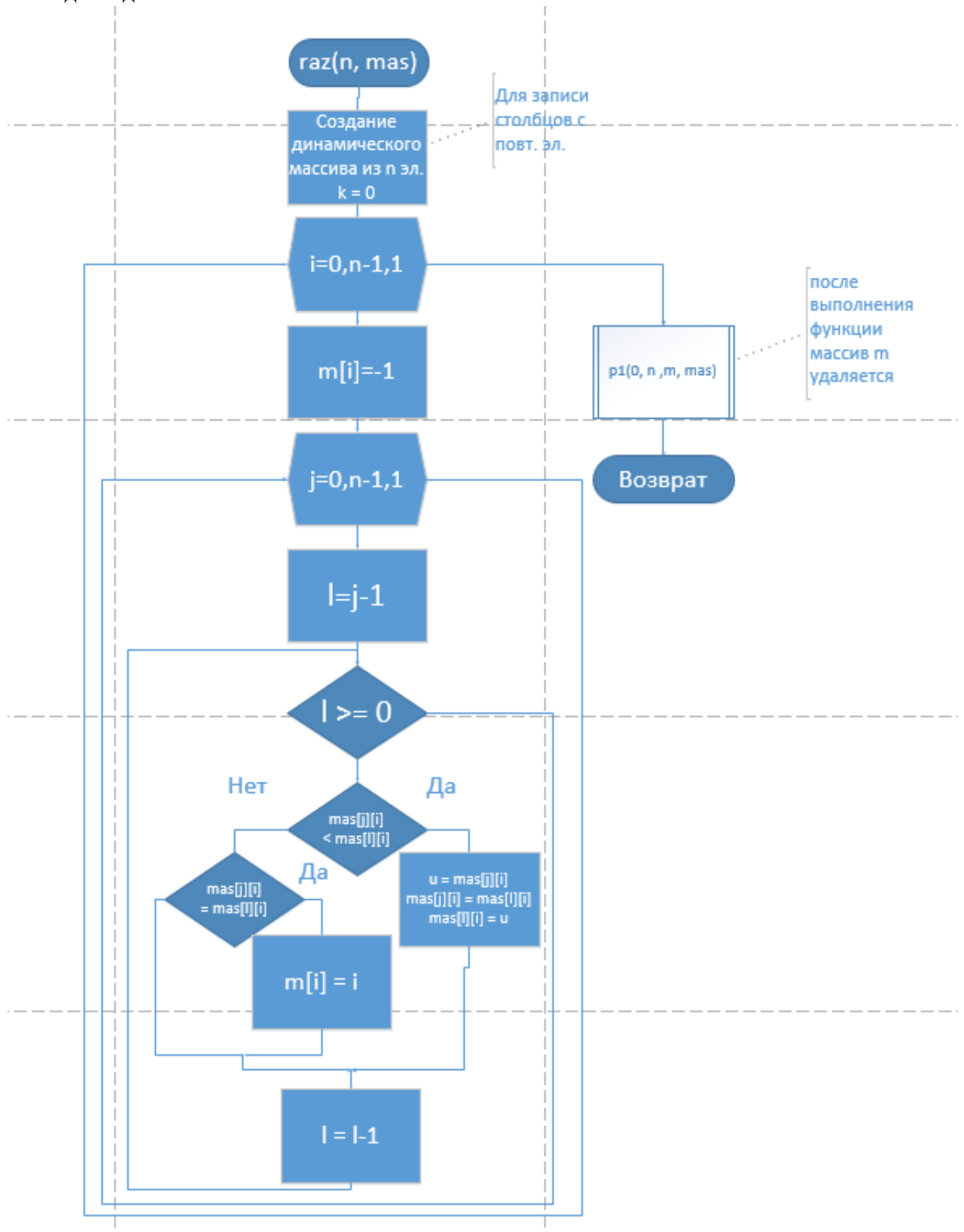


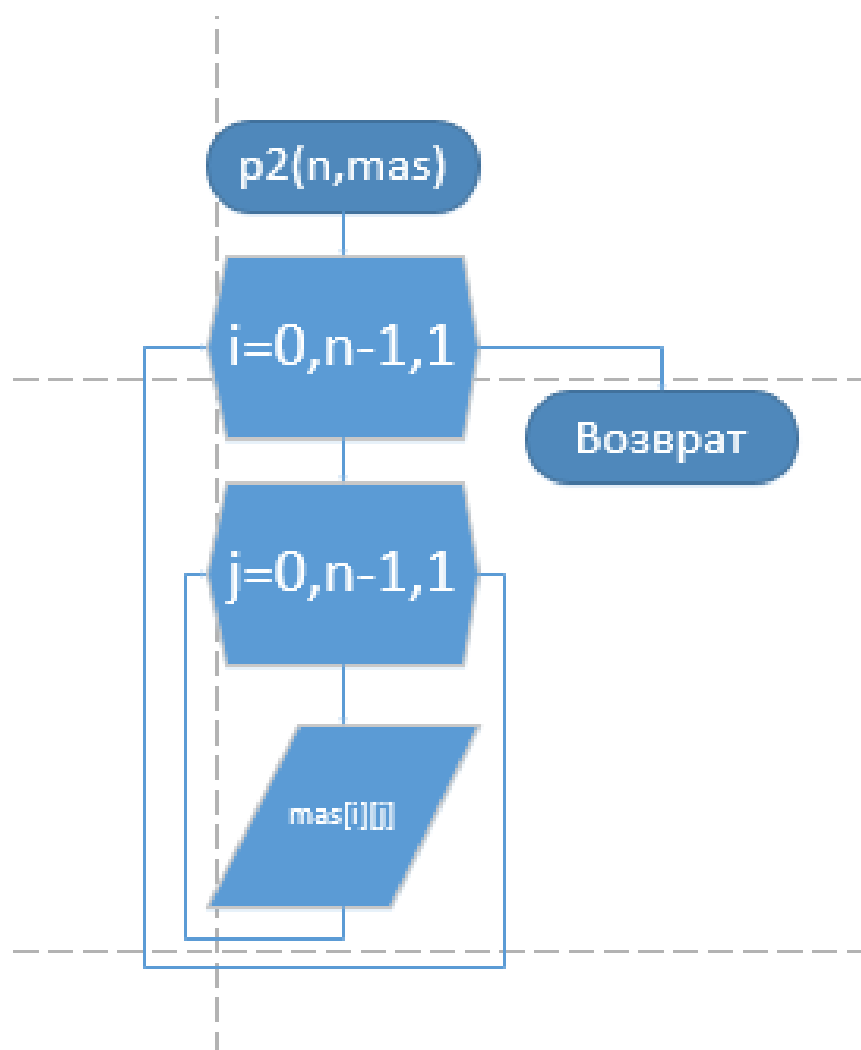
Рис.15

Функция p2:

Назначение: вывод на экран матрицы mas (см. Рис.16)

Входные данные: n - константа, mas - матрица

Выходные данные: нет



**Рис.16 Функция вывода на экран эл.**

Смысл программы: пользователь вводит кол-во эл. квадратной матрицы с клавиатуры, после чего создаётся динамический массив из указателей на массивы. После чего идёт заполнение матрицы и проверка на ненулевые элементы (если найдёт, то пользователю выйдет сообщение об ошибке, не соответствии эл. и заданного условия и запись в err). (см. Рис.17, Рис.18)



**Рис.17 Начало блок-схемы основного алгоритма**

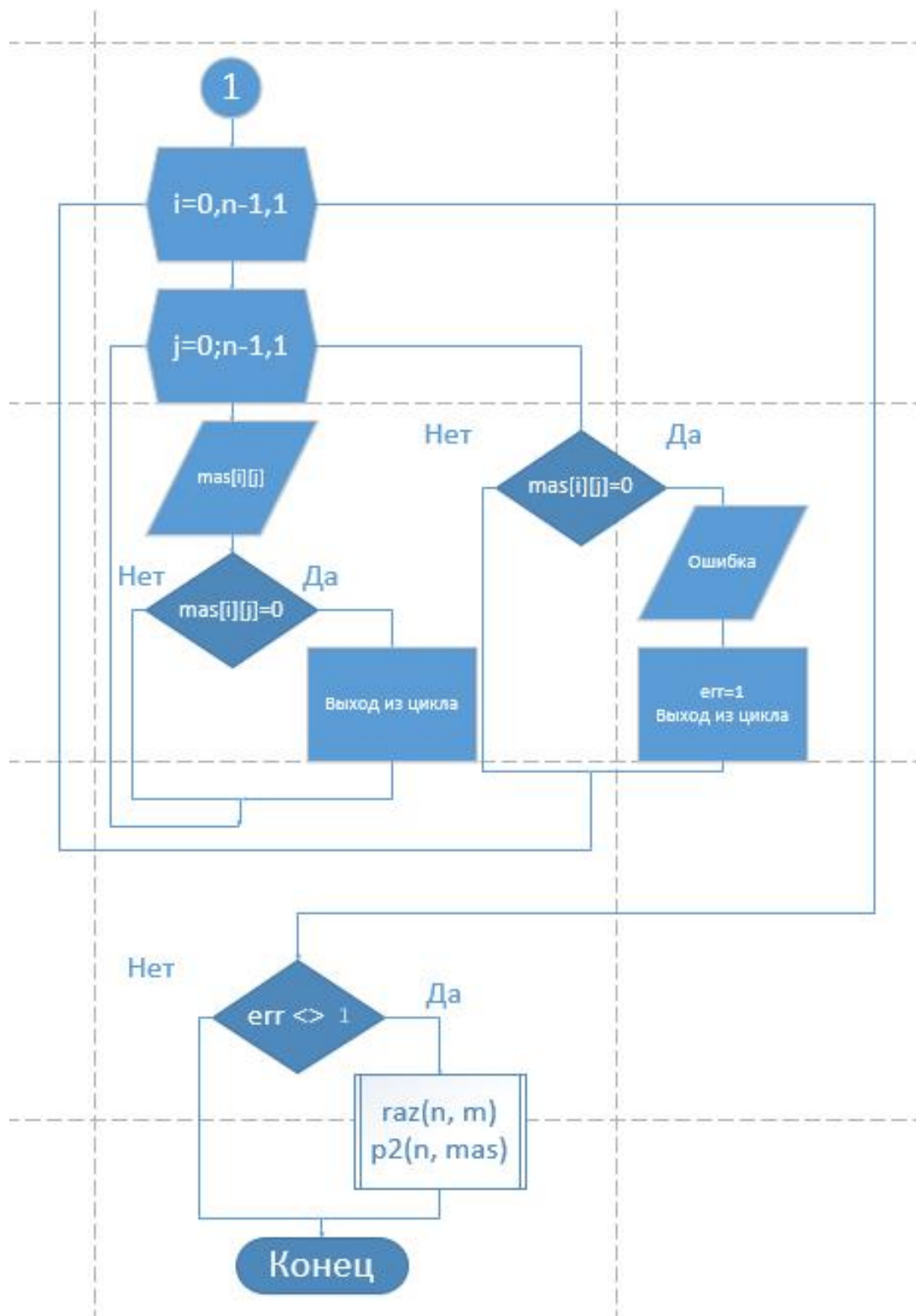


Рис.18 Продолжение блок-схемы основного алгоритма

## Текст программы:

```
#include <iostream>
#include <stdio.h>
#include <locale>

using namespace std;

void polozhit(const int n, int* mas[]) {
    int t = 0;
    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            if (mas[i][j] > 0) t++;
        }
    }

    if (t == 0) cout << "Нет положительных эл." << endl;
}

void p1(int k, const int n, int m[], int* mas[]) {
    int l, t = 0, c = 0;
    for (int i = 0; i < n; i++) {
        if (m[i] != (-1)) {
            k++;

            t = 0;
            if (k > 1) {
                if (c == 0) {
                    cout << "Меняем столбец " << (i + 1) << " обнуляя" <<
endl;

                    c++;
                }
                for (int j = 0; j < n; j++) {
                    mas[j][i] = 0;
                }
            }
            //Обнуление повторяющихся (см. на условие варианта)
            c = 0;
        }
    }
}

void kol_vo(int &i, const int n, int m[], int* mas[]) {
    int l = n - 1, kol = 0;

    for (int j = 0; j < n; j++) {
        while ((l >= 0) && (j!=l)) {
            if (mas[j][i] == mas[l][i]) {
                m[i] = i; //запомнил столбец с повт. эл.

                kol++;
            }
            l--;
        }
    }

    cout << "Одинаковое кол-во эл. в столбце " << (i + 1) << " равняется: " <<
kol << endl;
    kol = 0;
}

void raz(const int n, int* mas[]) {
    int l, u;
    int* m = new int[n];
```



```

for (int i = 0; i < n; i++) {
    polozhit(n, mas);
    m[i] = -1;
    int c = 0;

    kol_vo(i, n, m, mas);

    for (int j = n - 1; j >= 0; j--) {
        l = j - 1;
        while (l >= 0) {
            if (mas[j][i] < mas[l][i]) {
                if (c == 0) {
                    cout << "Меняем столбец " << (i+1) << " по
возрастанию" << endl;
                    c++;
                }
                u = mas[j][i];
                mas[j][i] = mas[l][i];
                mas[l][i] = u;
            }
            l--;
        }
    }

    c = 0;
} // Распределение по возрастанию эл. и запоминание столбцов с повт.

p1(0, n, m, mas);

delete[]m;
}

void p2(const int n, int* mas[]) {
    cout << "После выполнения:" << endl;

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            cout << mas[i][j] << "\t";
        }

        cout << endl;
    } //Вывод массива
}

void main() {
    setlocale(LC_CTYPE, "rus");
    cout << "Введите кол-во элементов n для A(n x n): " << endl;

    int n, i, j, err = 0;

    cin >> n;
    int** mas = new int* [n];

    for (i = 0; i < n; i++) {
        mas[i] = new int[n];
    }
}

```

```

////////////////////////////////////
cout << "Введите элементы не равные 0:,\n";
for (i = 0; i < n; i++) {
    for (j = 0; j < n; j++) {
        cin >> mas[i][j];
        if (mas[i][j] == 0) break;
    }
    if (mas[i][j] == 0) {
        cout << "Ошибка, найден элемент не удовлетворяющий условию!" << endl;
        err = 1;
        break; }
} // заполнение двумерного массива

if (err != 1) {

    raz(n, mas);

    p2(n, mas);

}

////////////////////////////////////
for (i = 0; i < n; i++) {
    delete[] mas[i];
} // Удалил одномерные массивы

delete[] mas; // Удалил динамический массив
}

```

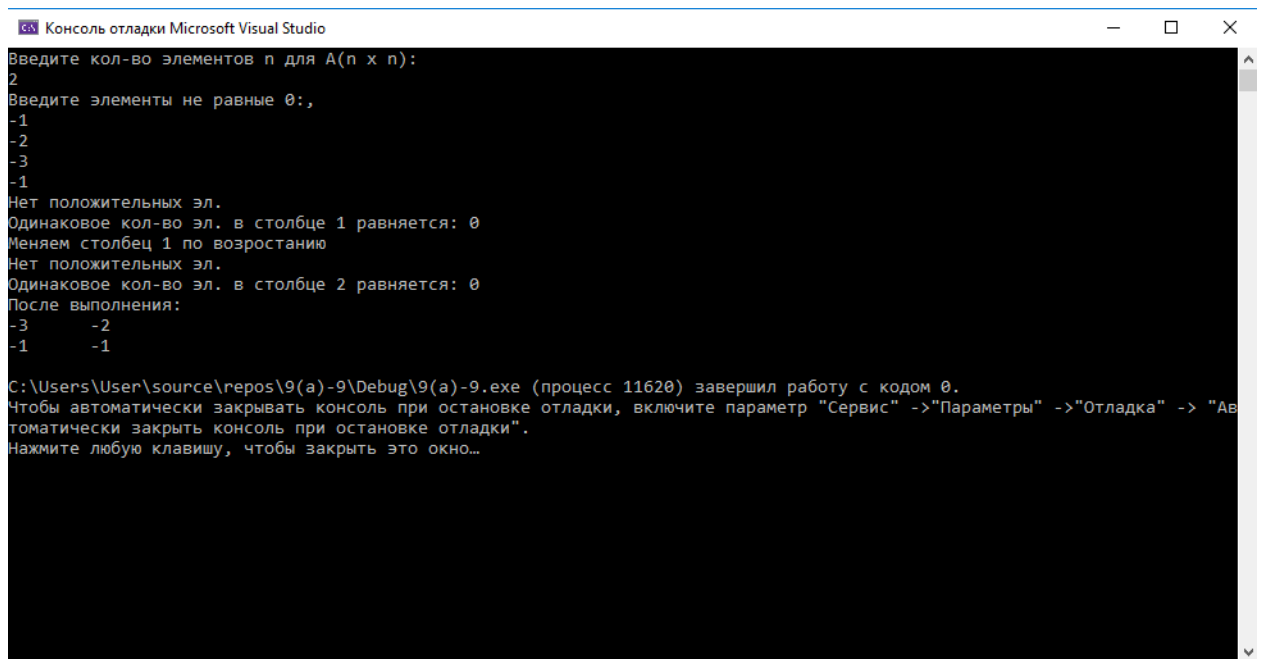
### Тестирование программы:

```

Консоль отладки Microsoft Visual Studio
Введите кол-во элементов n для A(n x n):
3
Введите элементы не равные 0:,
1
2
4
1
2
3
1
3
2
Одинаковое кол-во эл. в столбце 1 равняется: 3
Одинаковое кол-во эл. в столбце 2 равняется: 2
Одинаковое кол-во эл. в столбце 3 равняется: 0
Меняем столбец 3 по возрастанию
Меняем столбец 2 обнуляя
После выполнения:
1      0      2
1      0      3
1      0      4
C:\Users\User\source\repos\9(a)-9\Debug\9(a)-9.exe (процесс 16192) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" -> "Параметры" -> "Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...

```

Рис.19 Подсчёт столбцов и сообщение о том что меняется



```
Консоль отладки Microsoft Visual Studio
Введите кол-во элементов n для A(n x n):
2
Введите элементы не равные 0:,
-1
-2
-3
-1
Нет положительных эл.
Одинаковое кол-во эл. в столбце 1 равняется: 0
Меняем столбец 1 по возрастанию
Нет положительных эл.
Одинаковое кол-во эл. в столбце 2 равняется: 0
После выполнения:
-3      -2
-1      -1

C:\Users\User\source\repos\9(a)-9\Debug\9(a)-9.exe (процесс 11620) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

**Рис.20 Сообщение об отсутствии положительных элементов**

### 3. Заключение

Во время прохождения учебной практики я научился основам работы с GIT, понял его основную задачу и смог закрепить знания и навыки в написании программ на языке C/C++.