|  |  |  |
| --- | --- | --- |
| Министерство науки и высшего образования  Российской Федерации | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования | | |
| «Новосибирский государственный технический университет» | | |
|  | | |
| Кафедра теоретической и прикладной информатики | | |
|  | | |
| Лабораторная работа № 3 | | |
| по дисциплине «Основы программирования» | | |
|  | | |
| **Массивы** | | |
|  | | |
|  | Факультет: | ФПМИ |
| Группа: | ПМИ-12 |
|  |  |
| Студент: | Мироненко Алиса |
|  |
|  |
| Преподаватель: | Филиппова Елена Владимировна |
|  |  |
|  | | |
| Новосибирск | | |
| 2021 | | |

**1. Условие задачи**

В данной действительной квадратной матрице порядка n найти наибольший по модулю элемент. Получить квадратную матрицу порядка n-1 путем выбрасывания из исходной какой-нибудь строки или столбца, на пересечении которых расположен элемент с найденным значением.

**2. Анализ задачи**

**Дано:** Значение n. Некоторая действительная матрица

**Результат:** Программа выведет максимальный эл-т массива, номер строки и столбца этого эл-т, а затем матрицы, построенную на основе исходной, путем выбрасывания строки и столбца согласно условию задачи (в основном варианте). Подразумеваются и частные случаи, которые рассмотрены в ходе решения.

**Решение:**   
Проанализировав условие задачи, я решила прописать два варианта получения массива, состоящего из действительных чисел: с помощью ввода с клавиатуры (вложенные циклы for) и с автоматическим заполнением случайными числами (подключение библиотек **<cstdlib>** и **<ctime>,** использование функции **srand(time(NULL)).**

Однако перед этим пользователю предлагается ввести с клавиатуры размерность массива n\*n. Далее идет проверка, если размерность больше константного значения массива (10), то программа остановится на этом моменте и выведет на экран сообщение *«ERROOR: Превышен лимит заполнения матрицы».*

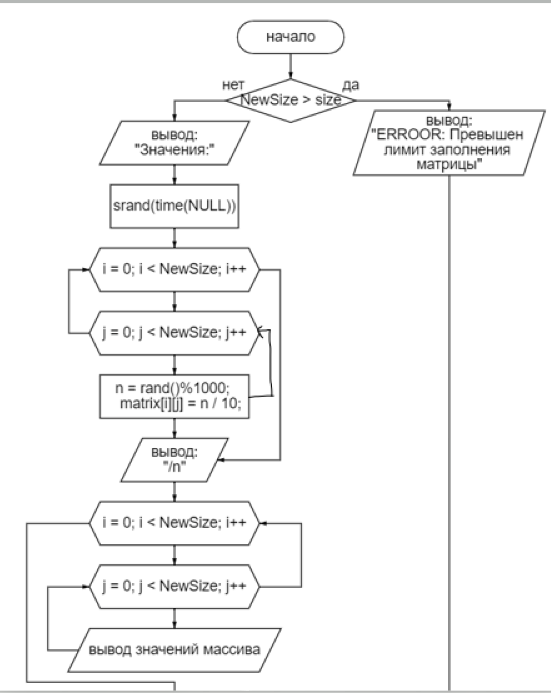
После чего были написаны вложенные циклы **for** для вывода, чтобы наглядно видеть полученный массив.

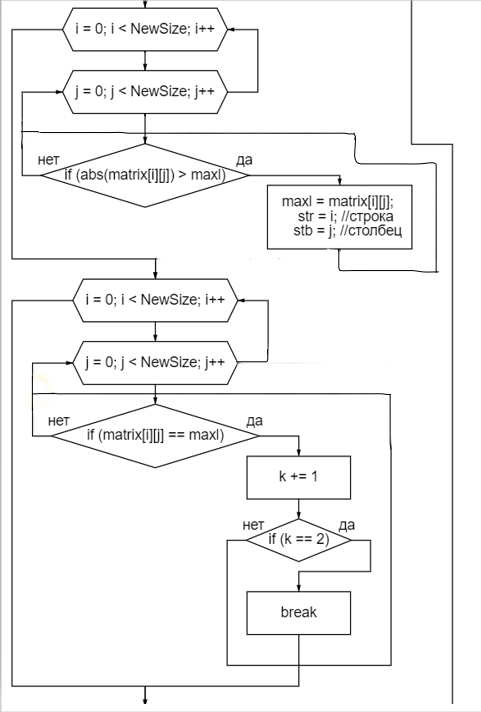
Далее были написаны вложенные циклы **for** для нахождения максимального эл-та в массиве (также его строки и столбца) и вложенные циклы **for** для проверки на наличие нескольких максимальных эл-тов в массиве. Если их несколько, то программа завершится на этом моменте и выведет на экран сообщение *"ERROOR: Невозможно выполнить программу, из-за того, что содержится более одного маскимального эл-та".*

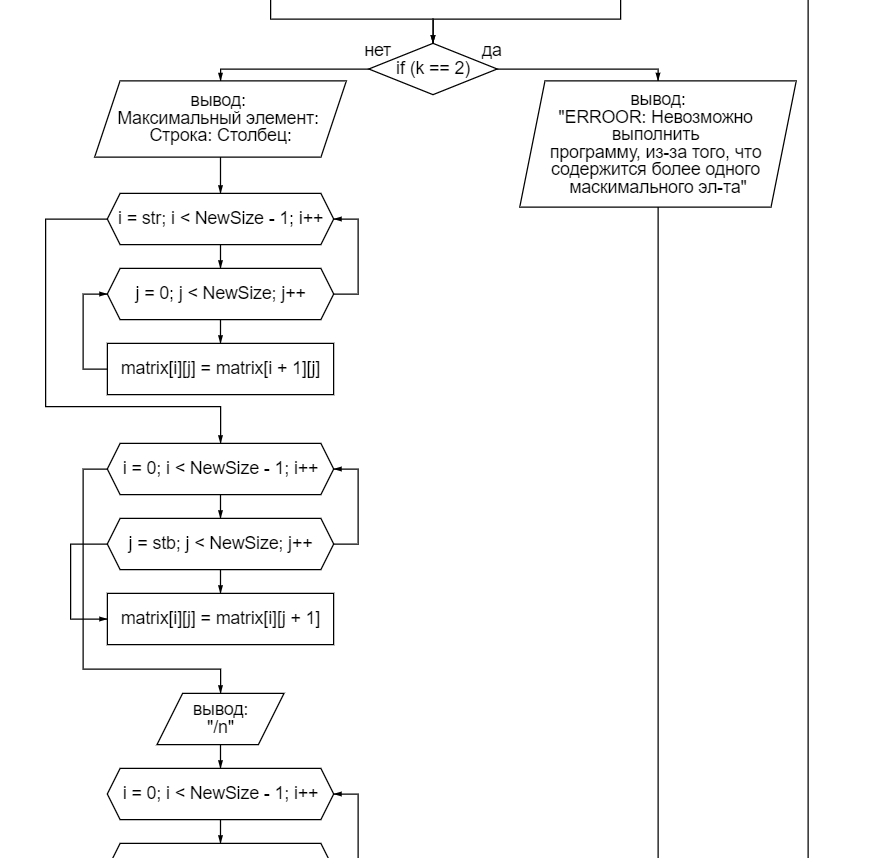
Далее программа выводит значения максимального эл-та в массиве, номер его строки и столбца.  
Поскольку удаление строк и столбцов в Си требует знания динамического программирования, было предпринято драконовское решение сдвигать их на 1 и формируя на основе данного принципа новую матрицу порядка n-1, где в выводе во вложенных циклах **for I** и **j** изменяются до n-1.

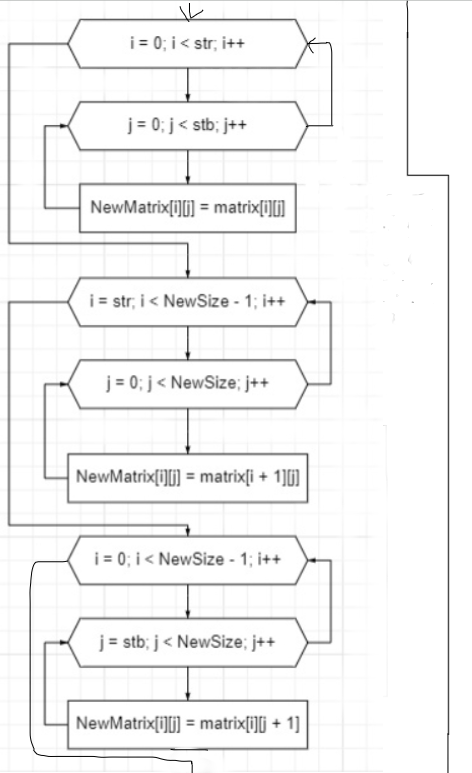
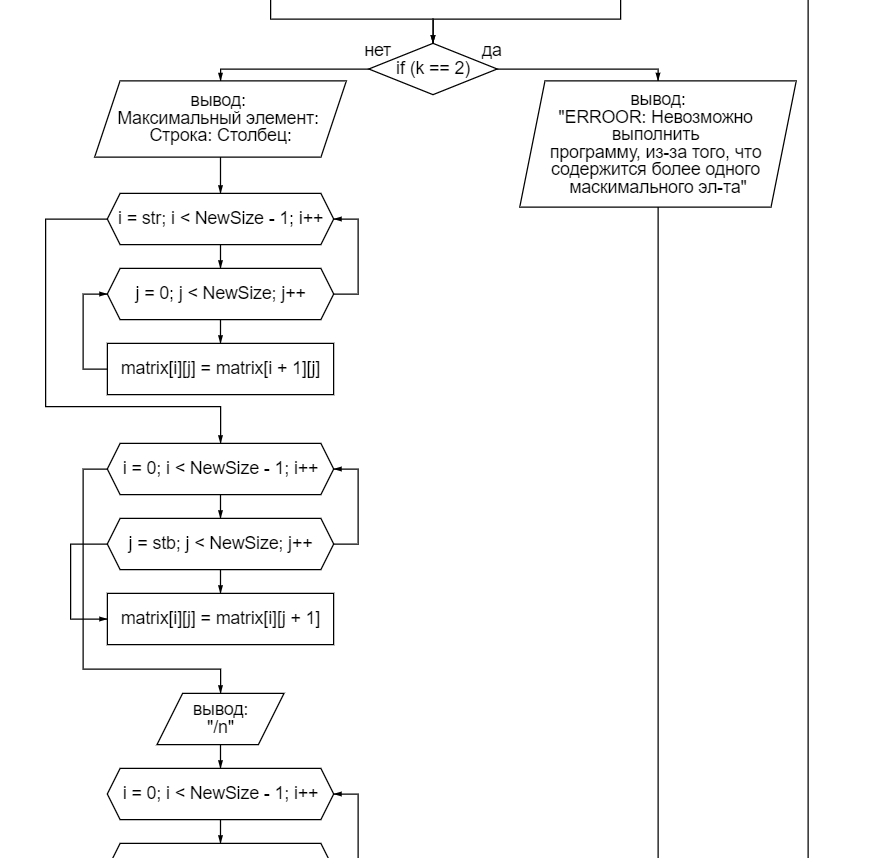
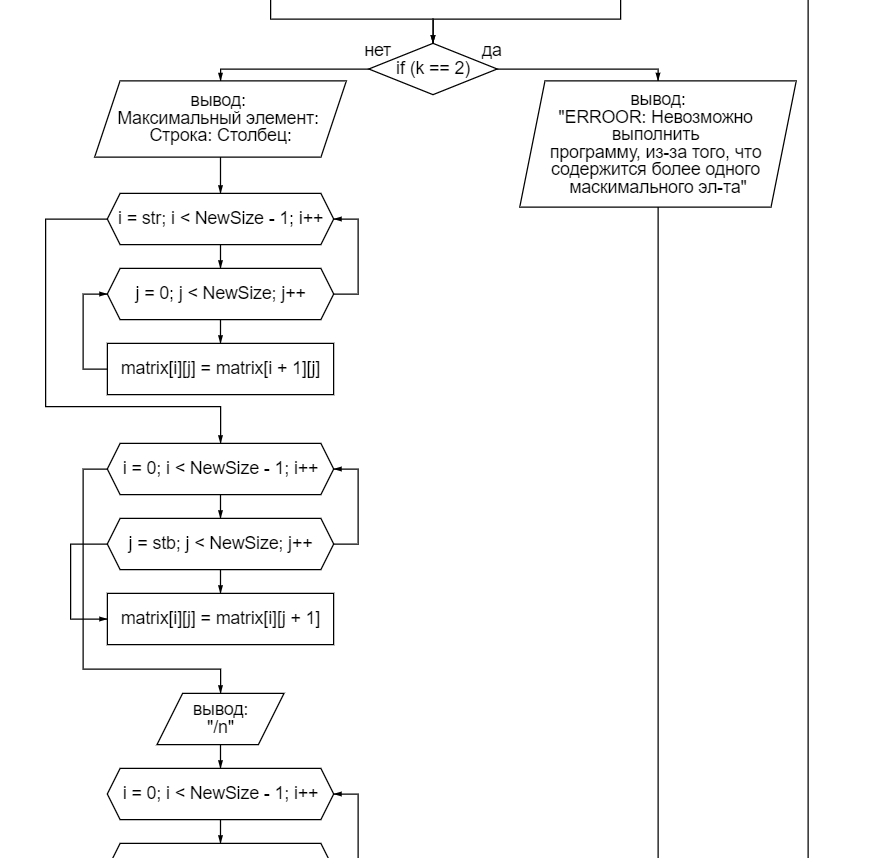
Далее производится вывод полученной матрицы порядка n-1.

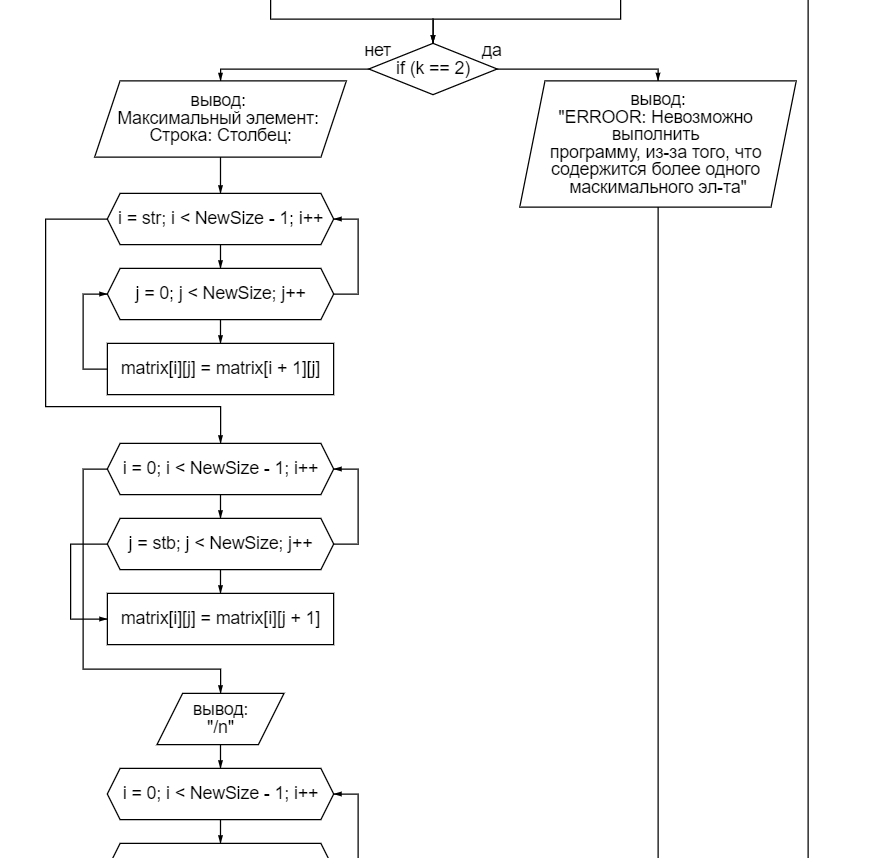
**3. Алгоритм решения задачи**

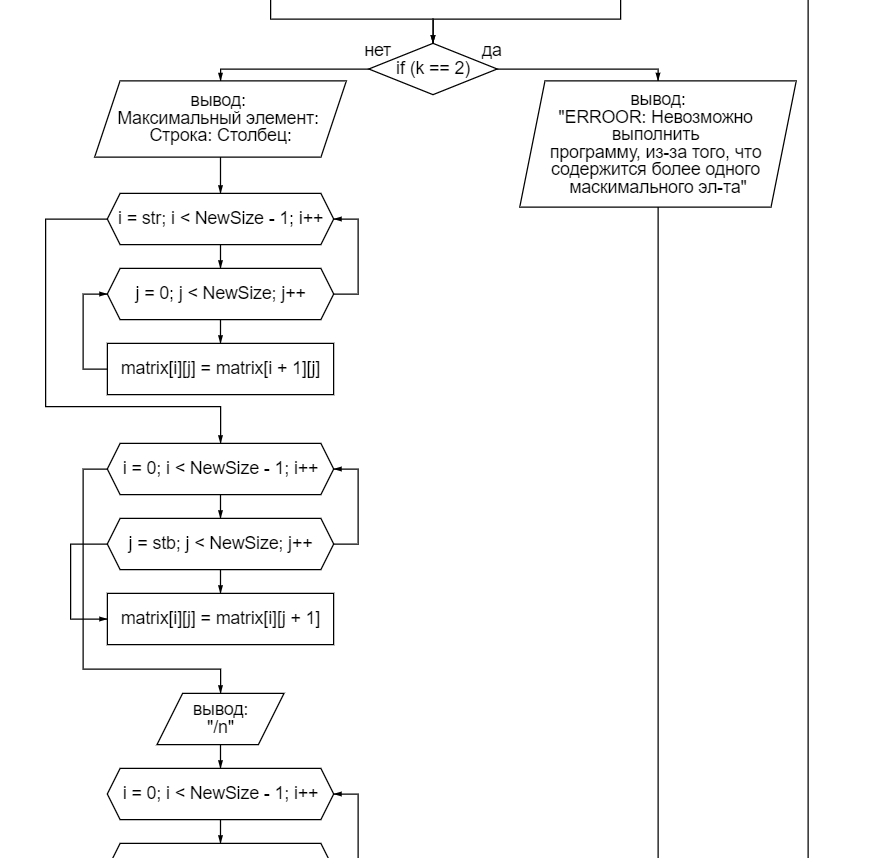
****

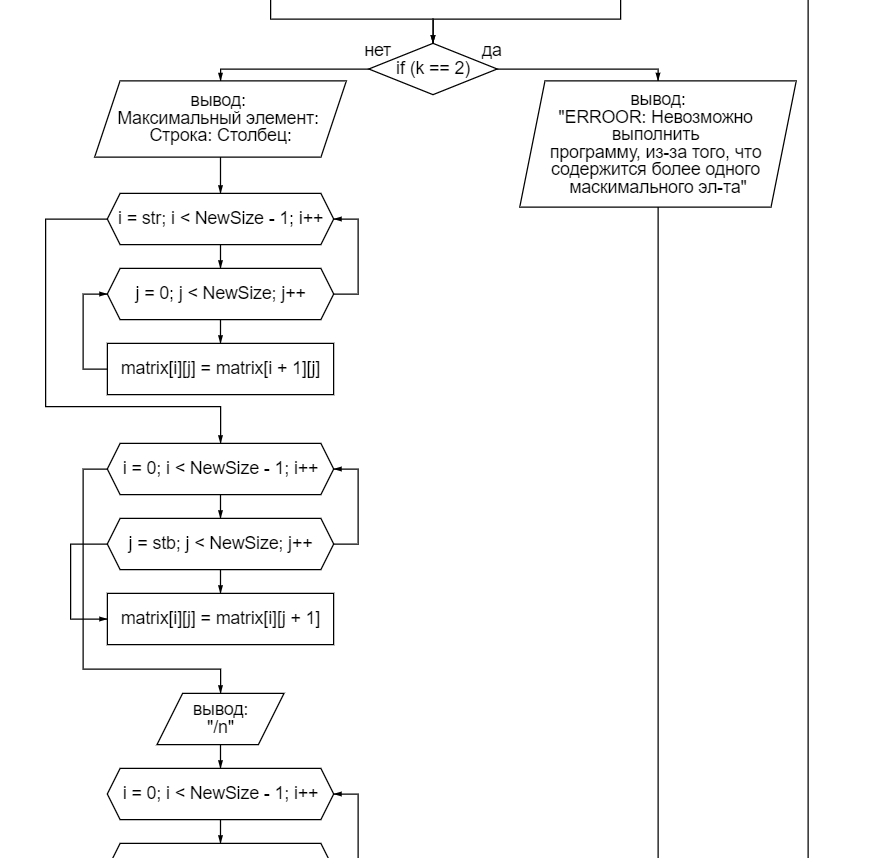
****

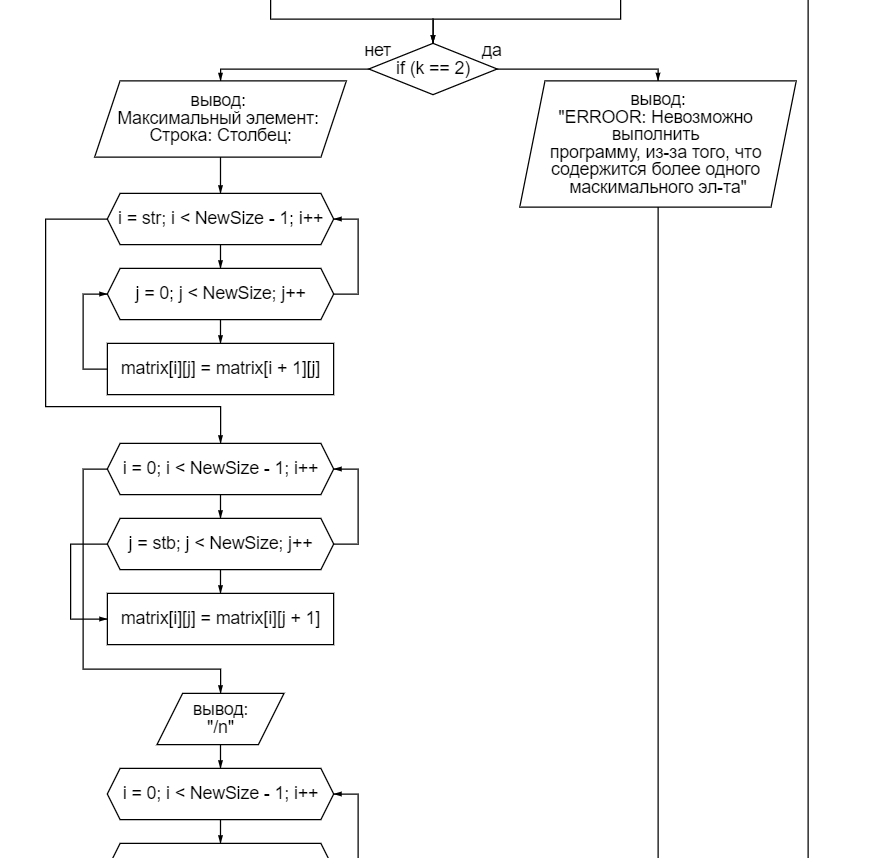
****

****

****

****

****

****

****

**4. Тело программы.**

#include <stdio.h>

#include <locale.h>

#include <math.h>

#include <stdlib.h>

#include <cstdlib>

#include <ctime>

int main()

{

setlocale(LC\_CTYPE, "Russian");

float n;

int k = 0; //счетчик макс эл-тов

float maxl = 0; //максимальный эл-т массива

int str = -1;//строка макс эл-та

int stb = -1;//столбец макс эл-та

const int size = 10;//макс размерность массива

int NewSize;//введенная размерность

float matrix[size][size];//массив исходный

float NewMatrix[size][size];//массив новый

printf("Введите размерность матрицы\n");

scanf\_s("%d", &NewSize);

if (NewSize > size) {

printf("ERROOR: Превышен лимит заполнения матрицы\n");

return 1;

}

printf("Значения:\n");

//for (int i = 0; i < NewSize; i++) { //заполняем массив вручную

//for (int j = 0; j < NewSize; j++) {

//scanf\_s("%f", &matrix[i][j]);

//}

//}

srand(time(NULL)); //заполнение массива случайными числами

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

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

n = rand() % 1000;

matrix[i][j] = n / 10;

}

}

printf("\n");

for (int i = 0; i < NewSize; i++) { //вывод значений

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

printf("%.1f\t", matrix[i][j]);

}

printf("\n");

}

for (int i = 0; i < NewSize; i++) { //сравнение по модулю

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

if (abs(matrix[i][j]) > maxl) {

maxl = abs(matrix[i][j]);

str = i; //строка

stb = j; //столбец

}

}

}

for (int i = 0; i < NewSize; i++) { //проверка на несколько максимумов в массиве

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

if (matrix[i][j] == maxl) {

k += 1;

if (k == 2) {

break;

}

}

}

}

if (k == 2) {

printf("ERROOR: Невозможно выполнить программу, из-за того, что содержится более одного маскимального эл-та");

return 1;

}

printf("Модуль максимального эл-та: %.1f\n Строка: %d\n Столбец: %d\n", maxl, str + 1, stb + 1);

for (int i = 0; i < str; i++) { //сохраняем часть где индекс строки < str и столбца < stb, формируем матрицу порядка n-1

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

NewMatrix[i][j] = matrix[i][j];

}

}

for (int i = str; i < NewSize - 1; i++) { //"удаление" строки, формируем матрицу порядка n-1

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

NewMatrix[i][j] = matrix[i + 1][j];

}

}

for (int i = 0; i < NewSize - 1; i++) { //"удаление" столбца, формируем матрицу порядка n-1

for (int j = stb; j < NewSize; j++) {

NewMatrix[i][j] = matrix[i][j + 1];

}

}

for (int i = 0; i < NewSize - 1; i++) { //вывод значений

for (int j = 0; j < NewSize - 1; j++) {

printf("%.1f\t", NewMatrix[i][j]);

}

printf("\n");

}

return 0;

}**5. Тесты.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **№** | **n** | **Матрица** | **Результат** | **Модуль макс эл-та, строка, столбец** | **Примечание** |
| **1** | 3 | 33.5 68.9 108.0  92.3 54.0 127.0  929.0 13.0 -13.1 | 68.9 108.0  54.0 127.0 | 929, 3, 1 | Матрица 3\*3, есть отрицательные эл-ты, вещественные числа |
| **2** | 3 | 3.0 2.0 4.0  5.0 8.0 9.0  9.0 1.0 7.0 | ERROOR: Невозможно выполнить программу, из-за того, что содержится более одного маскимального эл-та | (нет) | Матрица 3\*3, вещественные числа, есть два макс эл-та |
| **3** | 3 | -11.5 -7.0 -9.6  -5.0 -1.0 -3.4  -8.0 -50.0 -3.3 | -11.5 -9.6  -5.0 -3.4 | 50, 3, 2 | Матрица 3\*3, вещественные числа, все числа отрицательные |
| **4** | 5 | 34.0 45.5 96.5 25.1 85.6  0.2 10.5 89.2 93.0 16.5  22.8 76.3 59.0 9.2 19.8  67.1 59.3 94.4 68.0 70.0  86.0 0.2 47.5 5.1 86.8 | 0.2 10.5 93.0 16.5  22.8 76.3 9.2 19.8  67.1 59.3 68.0 70.0  86.0 0.2 5.1 86.8 | 96.5, 1, 3 | Матрица 5\*5, вещественные числа |
| **5** | 7 | 64.0 26.8 6.9 62.0 52.4 81.3 53.8  72.7 66.9 59.9 62.0 81.5 75.0 2.6  26.1 14.4 24.8 89.7 26.6 28.2 51.0  45.6 81.5 3.4 85.4 72.0 55.5 64.8  64.1 73.7 17.4 82.5 35.6 52.8 34.4  2.4 55.2 3.4 52.3 64.9 76.7 7.1  86.7 23.9 67.3 0.1 72.5 85.5 15.8 | 64.0 26.8 6.9 52.4 81.3 53.8  72.7 66.9 59.9 81.5 75.0 2.6  45.6 81.5 3.4 72.0 55.5 64.8  64.1 73.7 17.4 35.6 52.8 34.4  2.4 55.2 3.4 64.9 76.7 7.1  86.7 23.9 67.3 72.5 85.5 15.8 | 89.7, 3, 4 | Матрица 7\*7, вещественные числа |

**6. Результат работы.**

На всех тестах программа выдала ожидаемый результат, следовательно, задача решена верно.