ГУАП КАФЕДРА № 43

ОТЧЕТ

ЗАЩИЩЕН С ОЦЕНКОЙ ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ассистент |  |  |  | М. А. Мурашова |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| **Лабораторная работа 6. Обработка числовых матриц** |
| по курсу: ОСНОВЫ ПРОГРАММИРОВАНИЯ |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |
| --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4134К |  | Иванов И.В. |
|  |  | подпись, дата | инициалы, фамилия |

Санкт-Петербург 2022

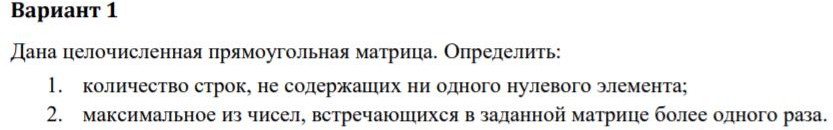
# Цель работы:

Целью работы является изучение структуры данных двумерный массив.

# Задание на лабораторную работу:

Задания на лабораторную работу приводятся в каждом варианте. При написании программ можно использовать как динамические, так и нединамические массивы. Размерность последних задаѐтся именованной константой.

# Ход выполнения:



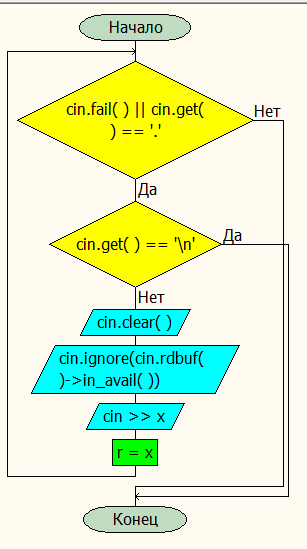
## Использованные функции:

**Имя:** checksize

**Назначение:** проверка на точку и на символы

**Побочные эффекты:** Отсутствует

## Блок-схема:



|  |  |
| --- | --- |
| **Входные данные** | **Выходные данные** |
| **влыфвы** | **Нее, так дело не пойдет** |
| **4** | **4** |

**Псевдокод:**

Если введенное значение содержит символы,

То просим ввести число повторно,

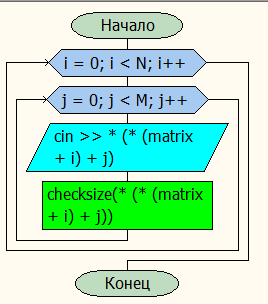
Иначе число корректно и с ним работает программа

**Имя:** enter

**Назначение:** ввод элементов

**Побочные эффекты:** отсутствуют

## Блок-схема:



**Псевдокод:**

Пока i<N

Пока j<M

Вводим матрицу

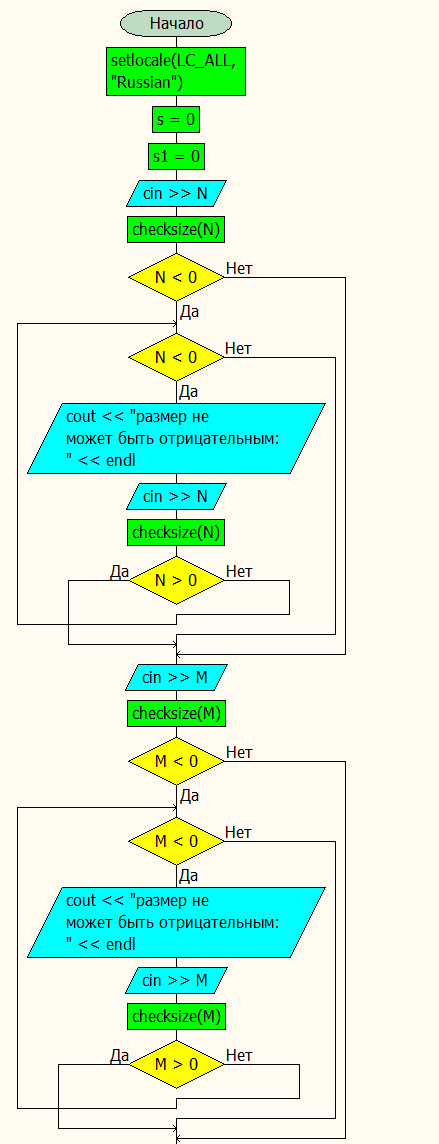
|  |  |
| --- | --- |
| **Входные данные** | **Выходные данные** |
| **1 2 3 5 8**  **6 7 7 7 7**  **8 9 0 0 0** | **1 2 3 5 8**  **6 7 7 7 7**  **8 9 0 0 0** |
| **2 5 5 3 0**  **1 1 1 4 0**  **6 4 5 8 0** | **2 5 5 3 0**  **1 1 1 4 0**  **6 4 5 8 0** |

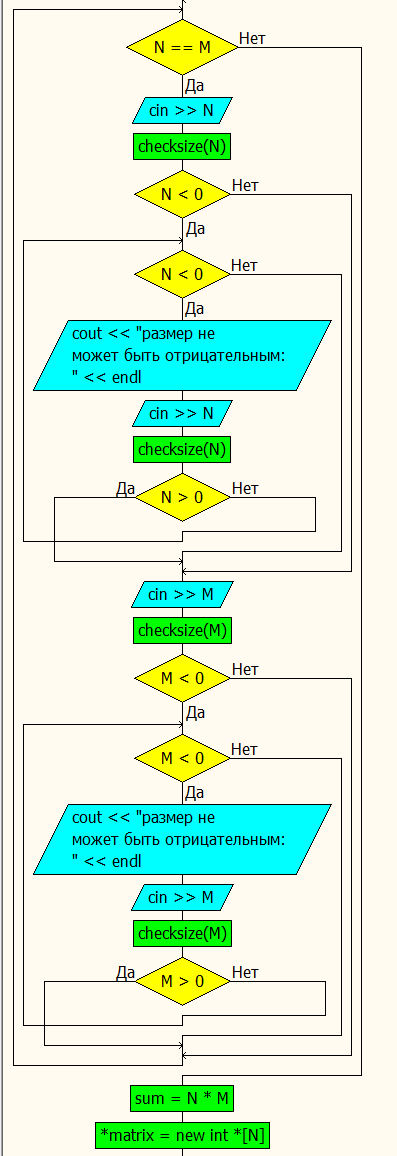
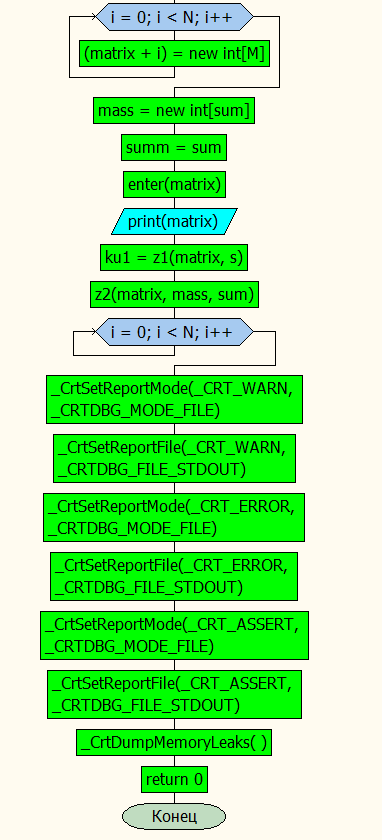
**Имя:** main

## Назначение:

**Побочные эффекты:** нет

## Блок-схема:





|  |  |
| --- | --- |
| **Входные данные** | **Выходные данные** |
|  |  |
|  |  |

**Псевдокод:**

Проверка на отрицательность матрицы и проверка, что матрица прямоугольная

Ввод размера матрицы

Вывод всех функций программы

Удаление матрицы

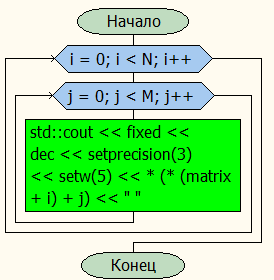
Проверка на утечку памяти

**Имя:** print

**Назначение:** вывод элементов

**Побочные эффекты:** отсутствуют

## Блок-схема:



**Псевдокод:**

Вывод элементов матрицы

**Имя:** z1

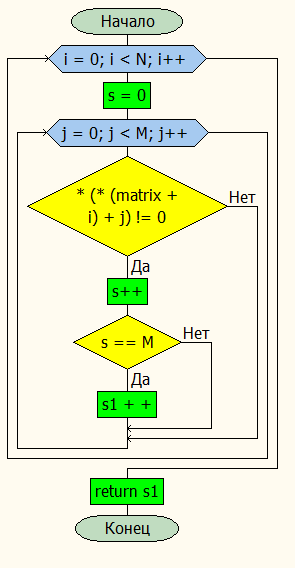
## Назначение: количество строк, не содержащих ни одного нулевого элемента

## Входные данные: \*k

**Выходные данные: s1**

**Побочные эффекты:** отсутствуют

## Блок-схема:



|  |  |
| --- | --- |
| **Входные данные** | **Выходные данные** |
| **4 3 0 1**  **5 6 8 9**  **2 0 7 1** | **2** |
| **4 9 2 1**  **5 6 0 9**  **1 0 7 3** | **1** |

**Псевдокод:**

Пока и меньше н

Пока дж меньше м

Если элемент матрицы не равен 0

То с+1

Иначе заново заходим в цикл фор

Если с равно м

То с1+1

**Имя:** z2

## Назначение: максимальное из чисел, встречающихся в заданной матрице более одного раза

## Входные данные: \*k Выходные данные:

**Побочные эффекты:** отсутствуют

## Блок-схема:

|  |  |
| --- | --- |
| **Входные данные** | **Выходные данные** |
| **2 3 4 5 5 5** | **5** |
| **3 4 5 6 6 6** | **6** |

**Псевдокод:**

Превращение матрицы в один массив

Попарная сортировка по убыванию

Если элемент равен следующим

Макс=элемент

ЫЫ

# Листинг всей программы:

}#define \_CRTDBG\_MAP\_ALLOC #include <stdlib.h>

#include <crtdbg.h> #ifdef \_DEBUG #ifndef DBG\_NEW

#define DBG\_NEW new ( \_NORMAL\_BLOCK , FILE , LINE )

#define newDBG\_NEW #endif

#endif #include<cmath> #include <iostream> #include <iomanip> using namespace std; int N = 0;

int M = 1;

void checksize(int& r) {

while (cin.fail() || cin.get() == '.')

{

if (cin.get() == '\n') {

break;

}

double x; cin.clear();

cin.ignore(cin.rdbuf()->in\_avail()); std::cout << " нее, так дело не пойдет : "; cin >> x;

r = x;

}

}

int z1(int \*\*matrix,int s1) {

int s;

for (int i = 0; i < N; i++) { s = 0;

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

if (\*(\*(matrix + i) + j) != 0) {

s++;

if (s == M) {

s1++;

}

}

}

}

return s1;

}

void z2(int\*\*matrix, int \*mass,int& sum) { int s;

int ku,ku1,ku2; int max;

max = 0;

ku1 = 0;

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

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

mass[ku1++] = \*(\*(matrix + i) + j);

}

}

for (int i = 1; i < sum; ++i)

{

for (int r = 0; r < sum - i; r++)

{

if (mass[r] < mass[r + 1])

{

int temp = mass[r]; mass[r] = mass[r + 1]; mass[r + 1] = temp;

}

}

}

for (int i = 0; i < sum; i++)

std::cout << mass[i] << " "; for (int i = 0; i < sum; i++) {

if (mass[i] == mass[i + 1]) {

max = mass[i]; break;

}

else {

}

}

mass[i] = mass[i + 1];

std::cout << endl;

std::cout << "максимальное из чисел, встречающихся в заданной матрице более одного раза " << max;

}

void enter(int \*\*matrix) {

std::cout << "Введите значения элементов " << endl; for (int i = 0; i < N; i++) {

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

cin >> \*(\*(matrix + i) + j); checksize(\*(\*(matrix + i) + j));

}

}

}

void print(int \*\*matrix) {

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

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

std::cout << fixed << dec << setprecision(3) << setw(5) << \*(\*(matrix + i) + j) << " ";

}

std::cout << endl;

}

}

int main() {

setlocale(LC\_ALL, "Russian"); int s, s1, sum,f;

s = 0;

s1 = 0;

std::cout << "Введите размер" << endl; cin >> N; checksize(N);

if (N < 0) {

while (N < 0) {

cout << "размер не может быть отрицательным: " << endl; cin >> N;

checksize(N); if (N > 0) {

break;

}

else {

}

}

}

continue;

cin >> M; checksize(M); if (M < 0) {

while (M < 0) {

cout << "размер не может быть отрицательным: " << endl; cin >> M;

checksize(M); if (M > 0) {

break;

}

else {

continue;

}

}

}

while (N == M) {

std::cout << "Матрица должна быть прямоугольной" << endl; cin >> N; checksize(N);

if (N < 0) {

while (N < 0) {

cout << "размер не может быть отрицательным: " << endl; cin >> N;

checksize(N); if (N > 0) {

break;

}

else {

}

}

}

continue;

cin >> M; checksize(M); if (M < 0) {

while (M < 0) {

cout << "размер не может быть отрицательным: " << endl; cin >> M;

checksize(M); if (M > 0) {

break;

}

else {

}

}

}

continue;

}

sum = N \* M;

int\*\* matrix = new int\* [N]; for (int i = 0; i < N; i++)

\*(matrix + i) = new int[M]; int\* mass = new int[sum]; int& summ = sum;

int ku1; enter(matrix); print(matrix);

ku1 = z1(matrix, s);

std::cout << "количество строк, не содержащих ни одного нулевого элемента " << ku1 << endl; z2(matrix, mass, sum);

for (int i = 0; i < N; i++)

delete[]matrix[i]; delete[]matrix; delete[]mass;

\_CrtSetReportMode(\_CRT\_WARN, \_CRTDBG\_MODE\_FILE);

\_CrtSetReportFile(\_CRT\_WARN, \_CRTDBG\_FILE\_STDOUT);

\_CrtSetReportMode(\_CRT\_ERROR, \_CRTDBG\_MODE\_FILE);

\_CrtSetReportFile(\_CRT\_ERROR, \_CRTDBG\_FILE\_STDOUT);

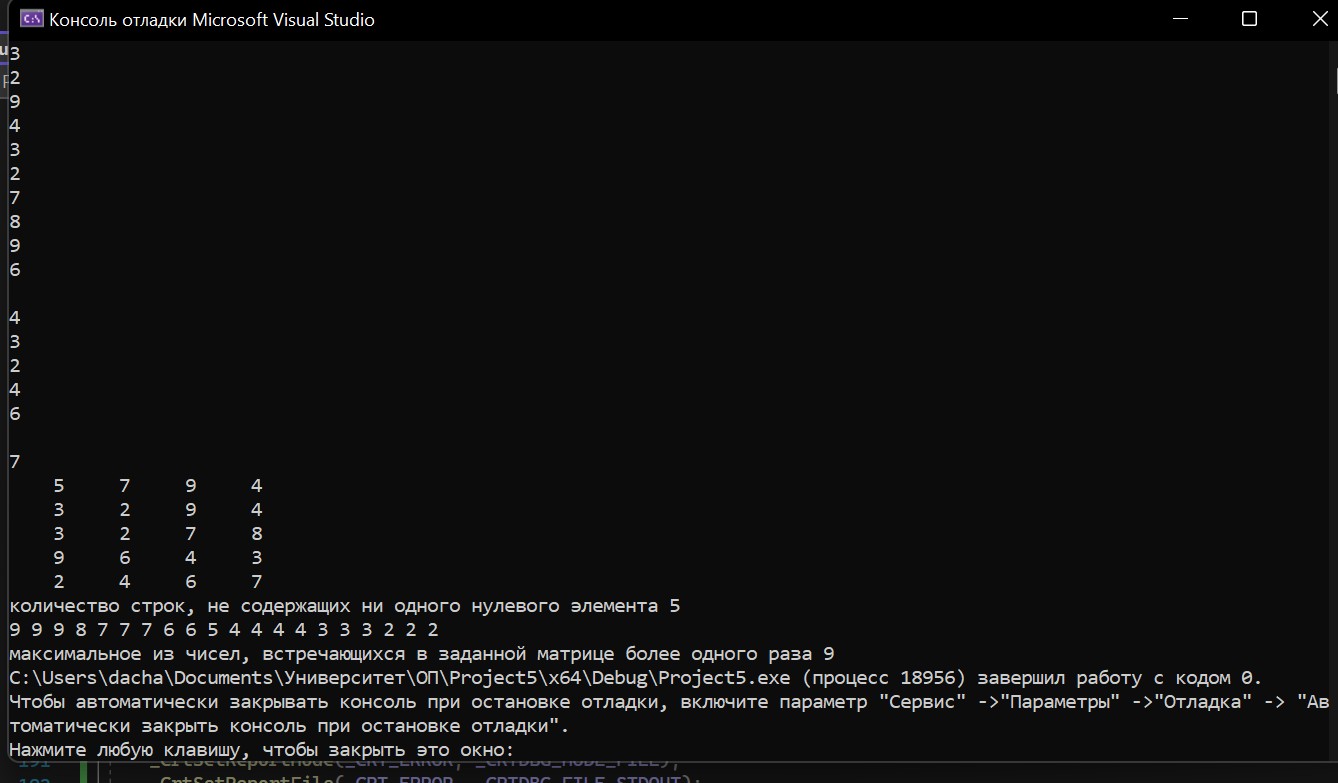
\_CrtSetReportMode(\_CRT\_ASSERT, \_CRTDBG\_MODE\_FILE);

\_CrtSetReportFile(\_CRT\_ASSERT, \_CRTDBG\_FILE\_STDOUT);

\_CrtDumpMemoryLeaks(); return 0;

}

# Результат работы программы:



**Вывод:**

Я изучил структуру данных двумерного массива.

# Достоинства программы:

Скорость работы.

Проверка значений.

Наличие пользовательского интерфейса.

# Недостатки программы:

Недостатков нет.