

КАФЕДРА №

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

должность, уч. степень, звание

подпись, дата

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

Обработка числовых матриц

по курсу: ОСНОВЫ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

подпись, дата

инициалы, фамилия

Санкт-Петербург 2022

1.Цель работы: Целью работы является изучение структуры данных двумерный массив

2.Задачи работы: Задания на лабораторную работу приводятся в каждом варианте. При написании программ можно использовать как динамические, так и нединамические массивы. Размерность последних задаётся именованной константой. Если использовались операторы динамического выделения памяти, то следует вставить дополнительный код, обнаруживающий утечки памяти. Все утечки памяти должны быть устранены. Путём перестановки элементов квадратной вещественной матрицы добиться того, чтобы её максимальный элемент находился в левом верхнем углу, следующий по величине – в позиции (2,2), следующий по величине – в позиции (3,3) и т.д., заполнить, таким образом, всю главную диагональ. -Найти номер первой их строк, не содержащих ни одного положительного элемента.

3.Описание функций:

main-основная функция

1)Имя: main

Назначение: ввод размеров матрицы, её наполнение числами, выполнение заданий из варианта.

Входные данные: нет

Выходные данные: нет

Побочный эффект: отсутствует.

Тестовые данные:

Ввод: size=3

-2	-8	-1
-2	0	4
4	-1	3

Вывод:

4	-8	-1
-2	4	-2
0	-1	3

В первой строке содержатся все отрицательные элементы

Прототип: int main()

Псевдокод:

ввод количества строк и столбцов в матрице

создание нового массива

ввод значений массива

вывод массива на экран

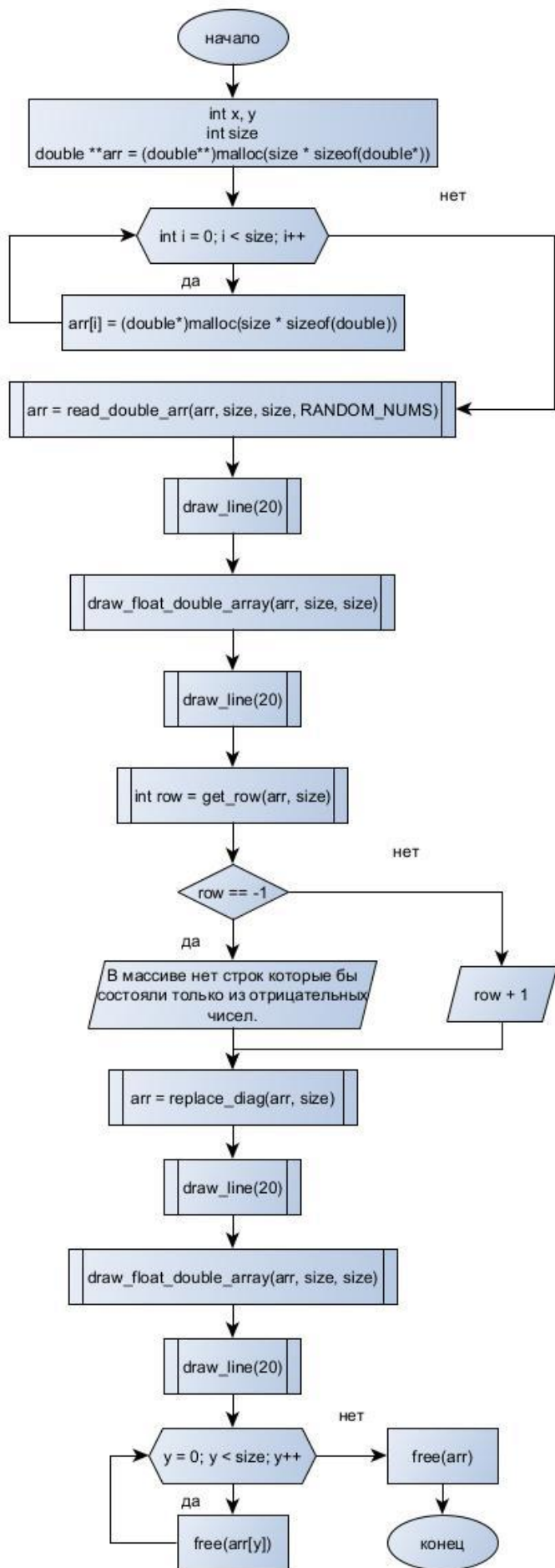
находим значение строки состоящей из отрицательных элементов

перестановка элементов массива в главную диагональ по убыванию

вывод нового массива

удаление массива

Блок-схема:



2)Имя: `read_size_arr()`

Назначение: ввод размера матрицы с проверкой

Входные данные: нет

Выходные данные: `size`

Побочный эффект: отсутствует.

Прототип: `int read_size_arr()`

Псевдокод:

Ввод числа

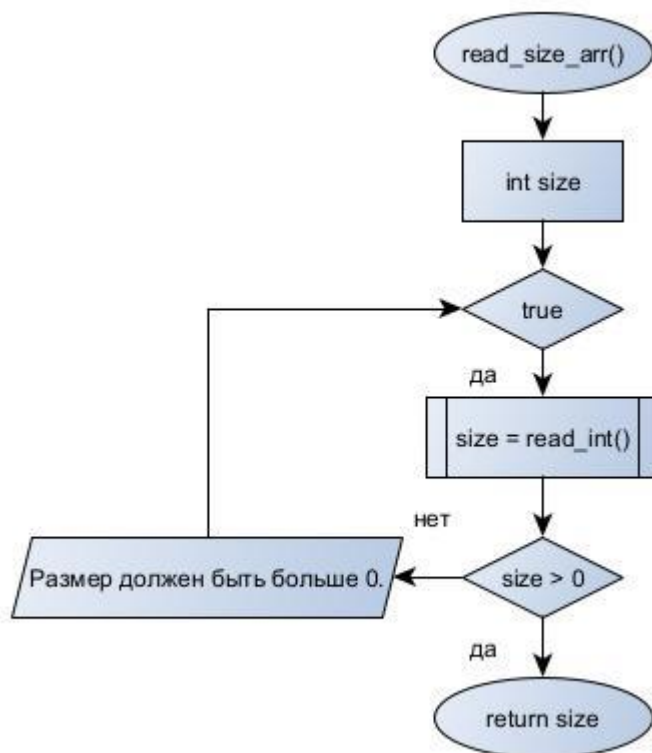
Если число не является таковым, то запросить ввод ещё раз

Если число больше нуля, то хорошо, если нет, то повторить ввод

Возврат в `main` корректного числа

Тестовые данные: (вход: комментарий для пользователя, выход: `size`)

Блок-схема:



3)Имя: `draw_line()`

Назначение: вывод линии на экран

Входные данные: нет

Выходные данные: нет

Побочный эффект: отсутствует.

Прототип: `void draw_line()`

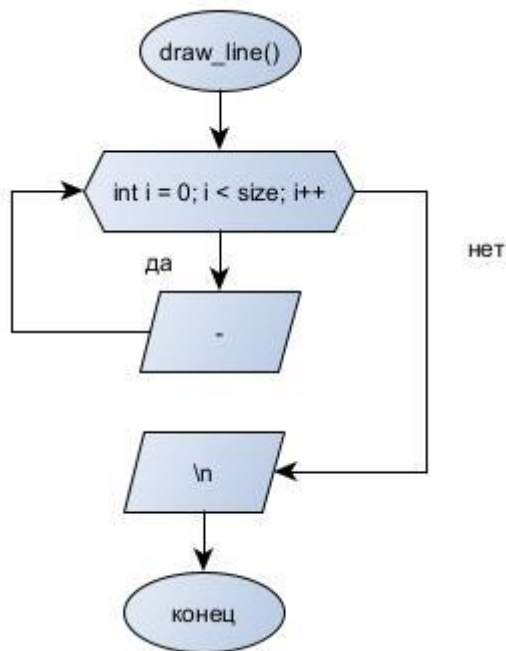
Псевдокод:

Пока `i` будет меньше размера, то

Вывод линии

Тестовые данные: (вход: количество линий, выход: вывод на экран линии)

Блок-схема:



4)Имя: `read_int()`

Назначение: проверка на корректность введенного числа

Входные данные: нет

Выходные данные: x

Побочный эффект: отсутствует.

Прототип: `int read_int()`

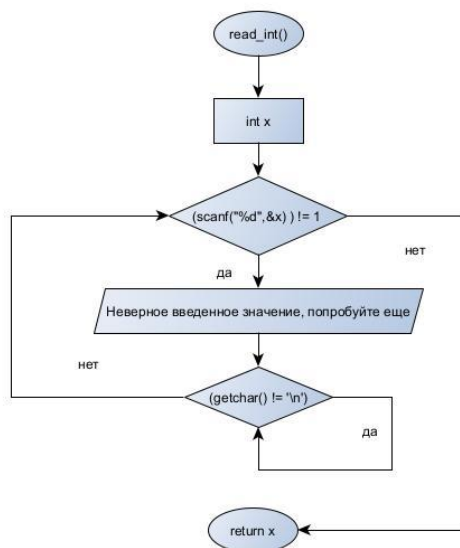
Псевдокод:

Ввод символа

Если символ =перенос строки, то прекратить ввод

Тестовые данные: (вход: нет, выход: x)

Блок-схема:



5)Имя: `read_double()`

Назначение: проверка на корректность введенного числа

Входные данные: нет

Выходные данные: x

Побочный эффект: отсутствует.

Прототип: `double read_double()`

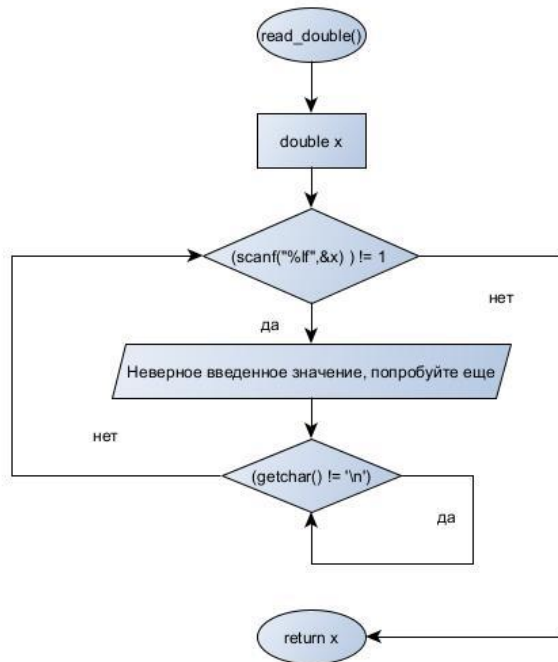
Псевдокод:

Ввод символа

Если символ =перенос строки, то прекратить ввод

Тестовые данные: (вход: нет, выход: x)

Блок-схема:



6)Имя: `replace_diag()`

Назначение: функция для перестановки элементов в массиве

Входные данные: `**arr,size`

Выходные данные: `**arr`

Побочный эффект: отсутствует.

Прототип: `double **replace_diag()`

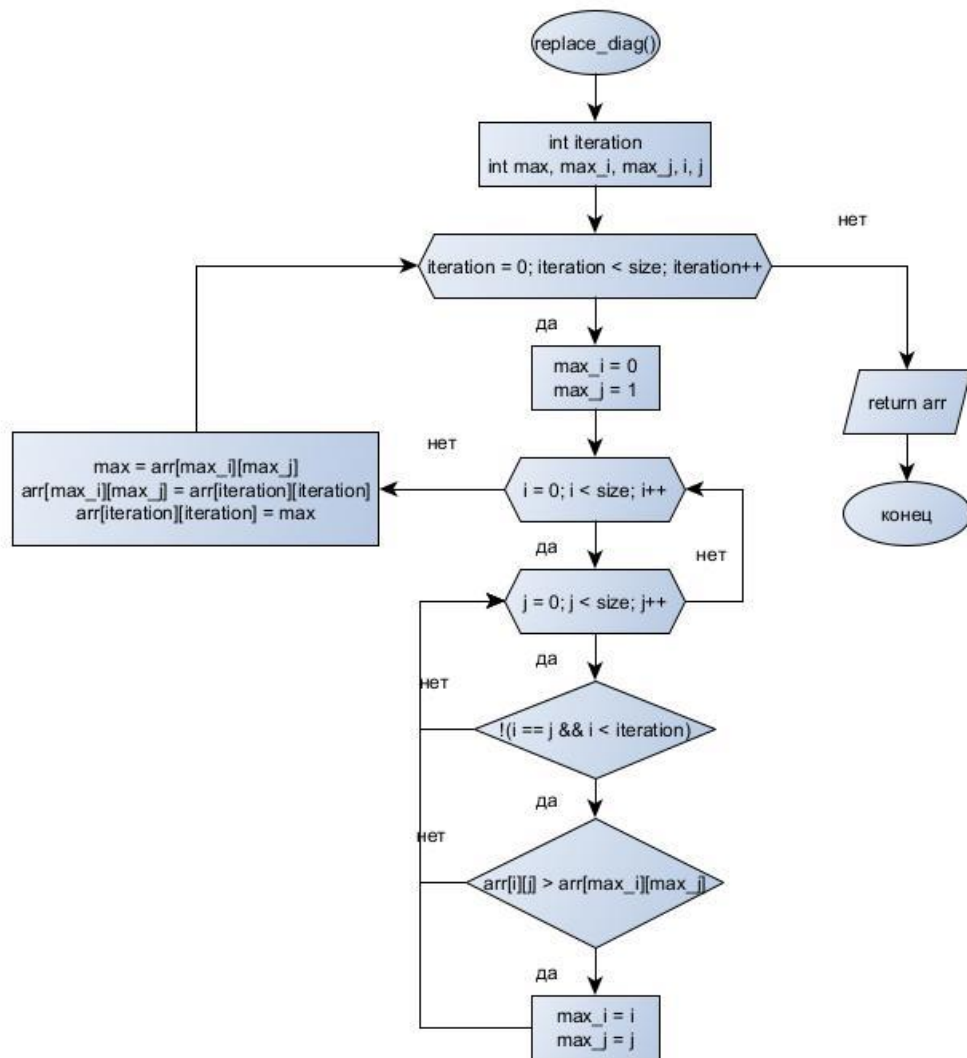
Псевдокод:

Перебор по диагонали

Перебор по массиву

Если $i \neq j$ и $I >$ количества итераций и текущий элемент больше максимального, то Максимальному элементу присваивается текущее значение

Блок-схема:



7)Имя: get_row()

Назначение: функция для нахождения номера строки состоящей из отрицательных элементов

Входные данные: **arr,size

Выходные данные: y or -1

Побочный эффект: отсутствует.

Прототип: int get_row()

Псевдокод:

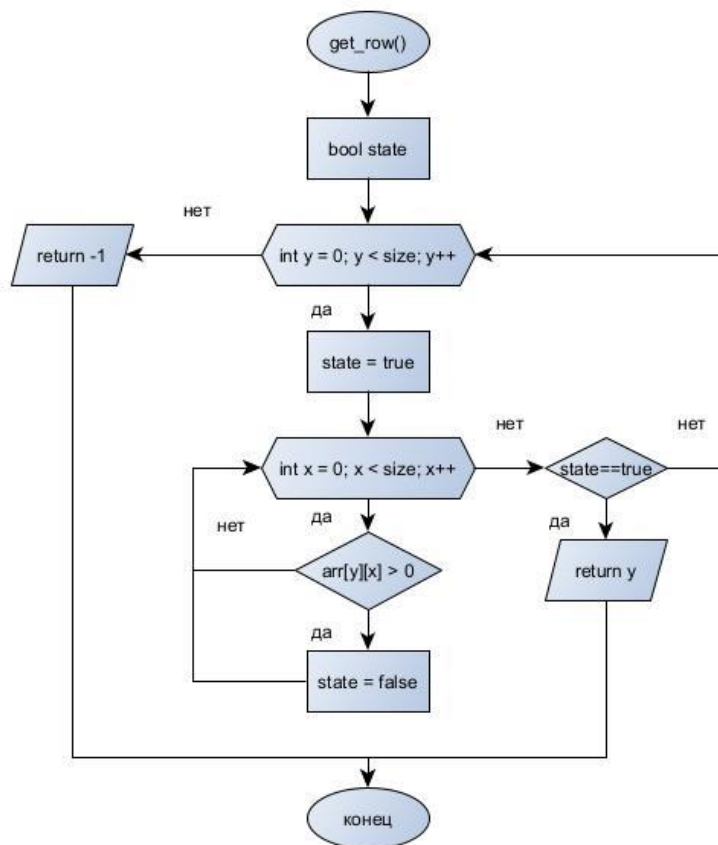
Создается флаг со значением true

Перебирается массив

Если находит в строке положительный элемент, то флаг со значением false

Если строка вся прошла проверку, то вывод ее номера и остановка функции

Блок-схема:



8)Имя: `read_double_array()`

Назначение: функция для заполнения двумерного массива

Входные данные: `size`

Выходные данные: `**arr`

Побочный эффект: отсутствует.

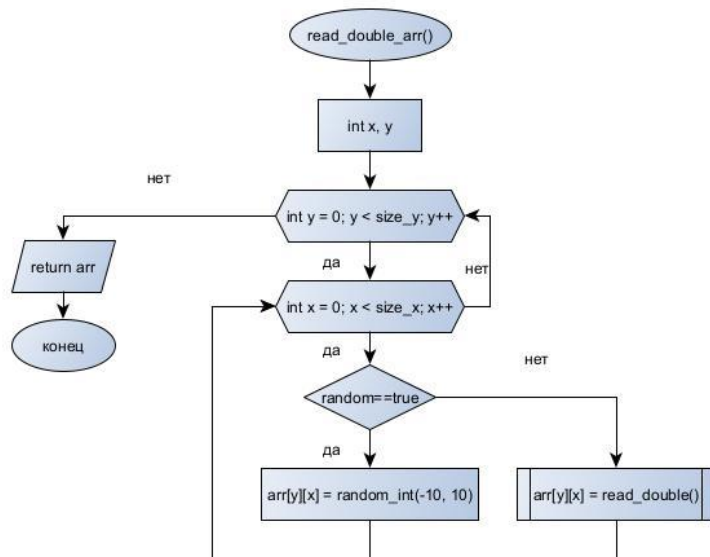
Прототип: `double **read_double_array()`

Псевдокод:

Перебор массива

Ввод значения элемента от пользователя и его проверка

Блок-схема:



9)Имя: draw_float_double_array()

Назначение: функция для вывода двумерного массива на экран

Входные данные: **arr,size

Выходные данные: нет

Побочный эффект: отсутствует.

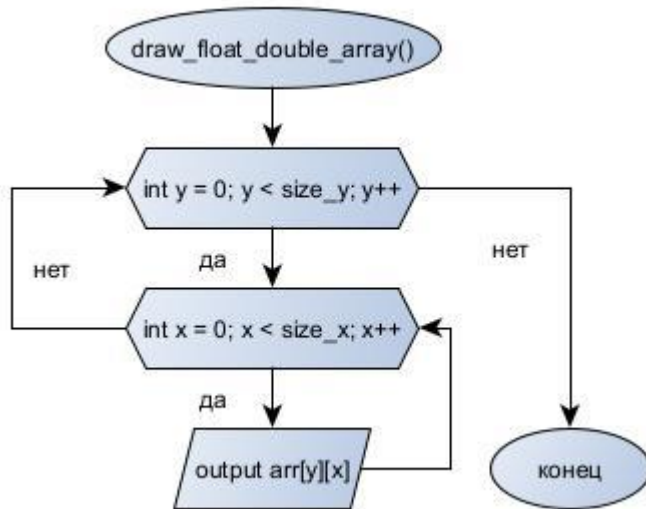
Прототип: void draw_float_double_array()

Псевдокод:

Перебор массива и вывод элементов на экран

Если конец строки, то перенос

Блок-схема:



5.ЛИСТИНГ КОДА:

```
#define RANDOM_NUMS false

#include <iostream>
using namespace std;

#include "libs/lib.h"
#include <cmath>
#include <time.h>

// работа с массивами
#include "libs/array.h"
#include "clear_arr.h"

void clear_scr() {
    cout << "\e[1;1H\e[2J";
}

void draw_line(int size) {
    for (int i = 0; i < size; i++)
        cout << '-';
    cout << endl;
}

int random_int(int a, int b) {
    return rand() % b + a;
}
```

```

int get_N(int r, int n) {
    int out = r % (n + 1);
    if (out < 1)
        return out + 1;
    else
        return out;
}

double read_double(){
    double x;
    while ( (scanf("%lf",&x) ) != 1 ) {
        printf("Неверное введенное значение, попробуйте еще: ");
        while(getchar() != '\n');
    }
    return x;
}

int read_int(){
    int x;
    while ( (scanf("%d",&x) ) != 1 ) {
        printf("Неверное введенное значение, попробуйте еще: ");
        while(getchar() != '\n');
    }
    return x;
}

// функция для ввода размера матрицы (с проверкой)
int read_size_arr(const char *promt = "") {
    int size;
    while (true) {
        cout << promt;
        size = read_int();
        if (size > 0) {
            break;
        } else {
            cout << "Размер должен быть больше 0." << endl;
        }
    }
    return size;
}

int main() {
    // смена кодировки
    system("chcp 65001");

    // очистка терминала
    //clear_scr();

    // рандом
    srand(time(NULL));

    int x, y;
    int size;

    // ввод размера массива
    size = read_size_arr("Размер квадратной матрицы: ");

    // создаём новый массив
    double **arr = (double**)malloc(size * sizeof(double*));
    for(int i = 0; i < size; i++) {
        arr[i] = (double*)malloc(size * sizeof(double));
    }
}

```

```

// ввод значений массива
arr = read_double_arr(arr, size, size, RANDOM_NUMS);

// выводим массив
draw_line(20);
draw_float_double_array(arr, size, size);
draw_line(20);

// нахождение строки состоящей из отрицательных элементов
int row = get_row(arr, size);
if (row == -1)
    cout << "В массиве нет строк которые бы состояли только из отрицательных чисел." << endl;
else
    cout << "Номер первой из строк, не содержащий ни одного положительного элемента: " << (row + 1) << endl;

// переставляем элементы
arr = replace_diag(arr, size);

// выводим новый массив
draw_line(20);
draw_float_double_array(arr, size, size);
draw_line(20);

// очистка памяти
for(y = 0; y < size; y++) {
    free(arr[y]);
}
free(arr);

return 0;
}

```

6.Пример выполнения программы:

main

```

C:\Windows\system32\cmd.exe
Active code page: 65001
Размер квадратной матрицы: 3
Array[0][0] = -2
Array[0][1] = -8
Array[0][2] = -1
Array[1][0] = -2
Array[1][1] = 0
Array[1][2] = 4
Array[2][0] = 4
Array[2][1] = -1
Array[2][2] = 3
-----
-2      -8      -1
-2       0       4
4       -1       3
-----
Номер первой из строк, не содержащий ни одного положительного элемента: 1
-----
4       -8      -1
-2       4      -2
0       -1       3
-----
C:\Users\nikit\Desktop\[ ПРОЕКТЫ ]\basic_programming_GUAP\17 вариант\6>pause
Press any key to continue . . .

```

7.Анализ результатов и выводы:

В ходе этой лабораторной работы мы изучили структуры данных двумерных массивов, а также работать с ними.

Из недостатков критических не обнаружено.

Из плюсов можно отметить, что заполнение программа работает корректно.