**Правительство Российской Федерации**

**Федеральное государственное автономное образовательное учреждение высшего профессионального образования   
"Национальный исследовательский университет   
"Высшая школа экономики"**

Московский институт электроники и математики Национального

исследовательского университета "Высшая школа экономики"

Департамент прикладной математики

**ОТЧЕТ**

**По лабораторной работе №5**

**По курсу «Алгоритмизация и программирование»**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  | ФИО студента | | Номер группы | Дата |
| Колодин Матвей Алексеевич | БПМ213 | 19.11.21 |
|  |
|  |
|  |

**Москва – 2021 г.**

**ЗАДАНИЕ:**

**Дана целочисленная матрица размера, где.**

**Программа должна быть разбита на несколько функций и обязательно содержать:**

**1) функцию формирования исходного массива;**

**2) функцию вывода исходного массива;**

**3) одну или более функций, реализующих вычислительную часть алгоритма.**

**Все функции должны содержать список параметров, причём адрес массива должен**

**передаваться как параметр функции. Функция main должна содержать только операторы вызова функций.**

**Использовать статический массив. Дополнительных массивов не использовать!**

**(вариант №14)**

**РЕШЕНИЕ:**

#include <math.h>

#include <stdio.h>

#include <malloc.h>

#include <stdlib.h>

void inp(int a[][10], int m, int n){ // 1. Формируем массив.

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

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

scanf("%d", &a[i][j]);

}

}

}

void outp(int a[][10], int m, int n){ // 2. Выводим массив.

printf("\n");

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

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

printf("%d ", a[i][j]);

}

printf("\n");

}

}

void clc(int a[][10], int m, int n){// 3. Вычислительная часть.

int minn = pow(2,31) - 1;

for(int i = m - 1; i >= 0; --i){ // побочная диагональ для m x n из левого нижнего угла идет.

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

if(a[i][j] < 0){

if(a[i][(m-1)-i] % 2 == 0 && (m-1)-i < n){

printf("%d is even. \n", a[i][(m-1)-i]);

if(a[i][(m-1)-i] <= minn){

minn = a[i][(m-1)-i];

}

}

}

}

}

if(minn == pow(2,31) - 1){

printf("Number doesn't exist");

}

else{

printf("Result: %d", minn);

}

}

int main(){

int m, n;

printf("Input m and n:\n");

scanf("%d %d", &m, &n);

int a[m][10];

// 1. Формируем массив.

inp(a, m, n);

// 2. Выводим массив.

outp(a, m, n);

// 3. Вычислительная часть.

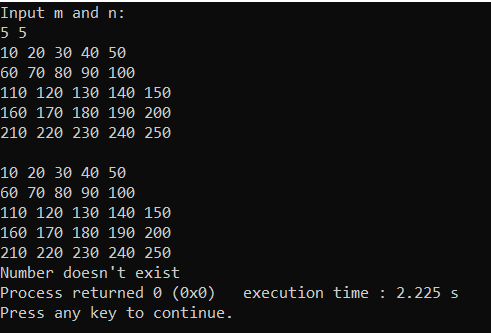
clc(a, m, n);

}

**ТЕСТЫ:**

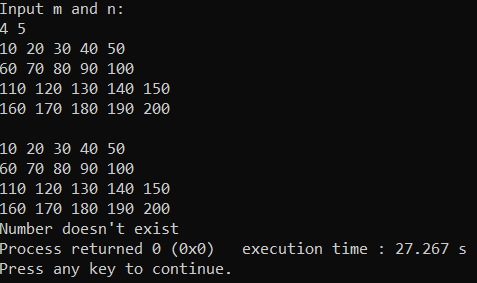
1. Первый блок тестов **(1-3)** направлен на проверку корректной работы программы в случае отсутствия искомого числа.
2. Второй блок тестов **(4-5)** направлен на проверку корректной работы программы в случае существования ответа.

**Тест № 1**

**

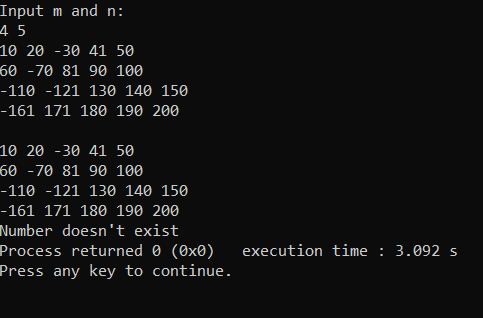
*(Квадратная матрица. Отрицательных чисел нет.)*

**Тест № 2**

**

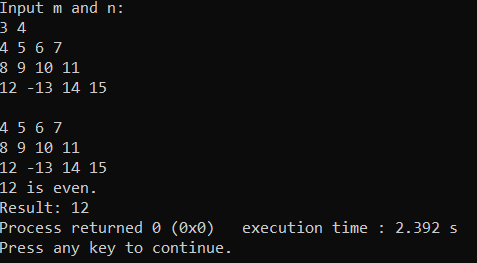
*(Проверка корректной работы программы в случае отсутствия отрицательных чисел. Прямоугольная матрица (вопрос корректного нахождения побочной диагонали).)*

**Тест № 3**



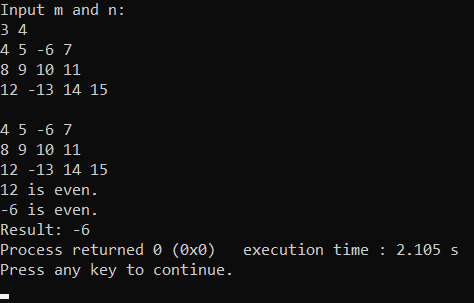
*(Проверка корректной работы программы в случае существования отрицательных чисел. Прямоугольная матрица. На побочной диагонали нет четных чисел.)*

**Тест № 4**



*(Проверка корректной работы программы в случае существования отрицательных чисел. Прямоугольная матрица. На побочной диагонали есть единственное четное число, в пересечении со строчкой, у которой есть отрицательные элементы.)*

**Тест № 5**



*(Проверка корректной работы программы в случае существования отрицательных чисел. Прямоугольная матрица. На пересечении с побочной диагональю есть несколько отрицательных чисел. На побочной диагонали отрицательное четное число.)*