

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 №6

По курсу «Алгоритмизация и программирование»

ФИО студента	Номер группы	Дата
Андреев Евгений Игоревич	БПМ-191	

Москва – 2019 г.

ЗАДАНИЕ (вариант №1)

Дана целочисленная матрица размера $m \times n$, где $2 \leq m, n \leq 10$.

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.

Номер варианта	Задание
1	Найти наименьшее из значений элементов первого из столбцов, которые обладают наибольшей суммой модулей элементов.

РЕШЕНИЕ

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

int scanf_m( ) // функция для ввода количества строк
{
    int m; // введение локальной целочисленной переменной
    do // цикл для ввода значения
    {
        printf("Введите количество строк(от 2 до 10): "); // вывод указания
        if(scanf("%d", &m) == 0 || getchar( ) != '\n') // считывание количества строк и проверка на ввод
            "левых значений"
        {
            printf("Нужно ввести целое число от 2 до 10.\nПовторите попытку.");
            // вывод указаний
            exit(0); // завершение программы
        }
    }
    while(m > 10 || m < 2); // проверка ввода
    return m; // возвращение количества строк
}

int scanf_n( ) // функция для ввода количества столбцов
{
    int n; // введение локальной целочисленной переменной
    do // цикл для ввода значения
    {
        printf("Введите количество столбцов(от 2 до 10): "); // вывод указания
        if(scanf("%d", &n) == 0 || getchar( ) != '\n') // считывание количества столбцов и проверка на ввод
            "левых значений"
        {
            printf("Нужно ввести целое число от 2 до 10.\nПовторите попытку.");
            // вывод указаний
            exit(0); // завершение программы
        }
    }
    while(n > 10 || n < 2); // проверка ввода
    printf("Ввод массива: \n"); // вывод оповещения о вводе массива
    return n; // возвращение количества столбцов
}
```

```

int** arrayInput(int size_m,int size_n)           // функция для объявления массива и считывания его
элементов
{
    int** array;                                // объявление массива
    array = (int**)malloc(size_m * sizeof(int*)); // выделение памяти размером size_m * sizeof(int*) байт
                                                    // для хранения строк
    if (NULL == array)                          // цикл для проверки, выделила ли ОС память на массив
    {
        printf("ОС не выделила память. Exit...\n"); // вывод ошибки
        return(0);                                // завершение программы
    }
    int i,j;                                    // введение целочисленных локальных переменных
    for (i = 0; i < size_m; i++)                // цикл по строкам
    {
        array[i] = (int*)malloc(size_n * sizeof(int)); // выделение памяти размером size_n * sizeof(int*) байт
                                                    // для хранения столбцов
        if (NULL == array[i])                  // цикл для проверки, выделила ли ОС память на массив
        {
            printf("ОС не выделила память. Exit...\n"); // вывод ошибки
            return(0);                                // завершение программы
        }
        for (j = 0; j < size_n; j++)            // цикл по столбцам
        {
            printf("array[%d][%d] = ", i + 1, j + 1); // вывод индексов элемента в массиве
            if(scanf("%d", &array[i][j]) == 0 || getchar() != '\n')
                                                    // ввод элементов массива и проверка на ввод "левых значений"
            {
                printf("Элементы массива по условию целочисленные.\nПовторите попытку.");
                exit(0); // вывод указаний
                // завершение программы
            }
        }
    }
    printf("Вывод массива: \n");                // вывод предупреждения о выводе массива
    return array;                               // возвращение массива
}

void arrayOutput(int** a, int size_m, int size_n) // функция для вывода элементов массива
{
    int i,j;                                    // введение целочисленных локальных переменных
    for (i = 0; i < size_m; i++)                // цикл для вывода строк
    {
        for (j = 0; j < size_n; j++)            // цикл для вывода столбцов
        {
            printf("a[%d][%d] = %d, ", i + 1, j + 1, a[i][j]); // вывод элемента массива
        }
        printf("\n");                          // перевод строки
    }
    printf("\n");
}

int max(int** a, int size_m, int size_n)         // функция для нахождения индекса столбца
                                                    // с наибольшей суммой модулей элементов
{
    int i, j, jmax = 0;                        // введение целочисленных локальных переменных
    double sc[size_n], dmax = a[0][0];         // введение вещественных локальных переменных
    for(i = 0; i < size_n; i++) sc[i] = 0;     // цикл для заполнения массива нулями
    for(i = 0; i < size_m; i++)                // цикл для строк
    {
        for(j = 0; j < size_n; j++)            // цикл для нахождения суммы модулей элементов столбцов
        {
            sc[j] += fabs(a[i][j]);            // суммирование модулей элементов в столбцах
        }
    }
}

```

```

    }
    for(j = 0; j < size_n; j++)           // цикл для сравнения сумм модулей элементов в столбцах
    {
        if(sc[j] > dmax)                  // сравнение сумм
        {
            dmax = sc[j];                // присваивание найденного большего значения элемента
            jmax = j;                    // сохранение его индекса в переменную
        }
    }
    return jmax;                          // возвращение индекса найденной строки
}

int min(int** a, int size_m, int size_n) // функция для определения минимального элемента в
столбце
{
    int i, j, dmin = a[0][max(a, size_m, size_n)]; // введение целочисленных локальных переменных
    for(i = 0; i < size_m; i++)                  // цикл для определения минимального элемента в столбце
    {
        if (a[i][max(a, size_m, size_n)] < dmin) // условие для нахождения минимального элемента в
столбце
        {
            dmin = a[i][max(a, size_m, size_n)]; // присваивание найденного значения минимуму
        }
    }
    printf("Наименьшее значение элемента в столбце %d = %d.", max(a, size_m, size_n) + 1, dmin);
    // вывод минимального значения
    free(a); // освобождение массива
    // функция free() возвращает память назад в кучу
    // в результате эта память может выделяться снова
    return dmin; // возвращение минимального элемента в столбце
}

int main( ) // старт программы
{
    int m = scanf_m( ), n = scanf_n( ); // ввод целочисленных переменных
    int** a; // объявление двумерного массива
    a = arrayInput(m, n); // создание массива и ввод его элементов
    arrayOutput(a, m, n); // вывод исходного массива
    min(a, m, n); // вывод минимального элемента в столбце
    return 0;
}

```

ТЕСТЫ

Тест № 1

Защита ввода:

```
Введите количество строк(от 2 до 10): 3s
Нужно ввести целое число от 2 до 10.
Повторите попытку.
```

```
Введите количество строк(от 2 до 10): 2
Введите количество столбцов: 2d
Нужно ввести целое число от 2 до 10.
Повторите попытку.
```

```
Введите количество строк(от 2 до 10): 1
Введите количество строк(от 2 до 10): 11
Введите количество строк(от 2 до 10): 2
Введите количество столбцов: 1
Введите количество столбцов: 11
Введите количество столбцов: 10
Ввод массива:
array[1][1] = 12.2a
Элементы массива по условию целочисленные.
Повторите попытку.
```

Тест № 2

Случай, когда максимальные суммы столбцов совпадают:

```
Введите количество строк(от 2 до 10): 2
Введите количество столбцов(от 2 до 10): 3
Ввод массива:
array[1][1] = 1
array[1][2] = 2
array[1][3] = 3
array[2][1] = 3
array[2][2] = 2
array[2][3] = 1
Вывод массива:
a[1][1] = 1, a[1][2] = 2, a[1][3] = 3,
a[2][1] = 3, a[2][2] = 2, a[2][3] = 1,
Наименьшее значение элемента в столбце 1 = 1.
```

Тест № 3

Случай, когда все элементы равны нулю:

```
Введите количество строк(от 2 до 10): 3
Введите количество столбцов(от 2 до 10): 3
Ввод массива:
array[1][1] = 0
array[1][2] = 0
array[1][3] = 0
array[2][1] = 0
array[2][2] = 0
array[2][3] = 0
array[3][1] = 0
array[3][2] = 0
array[3][3] = 0
Вывод массива:
a[1][1] = 0, a[1][2] = 0, a[1][3] = 0,
a[2][1] = 0, a[2][2] = 0, a[2][3] = 0,
a[3][1] = 0, a[3][2] = 0, a[3][3] = 0,
Наименьшее значение элемента в столбце 1 = 0.
```

Тест № 4

Случаи с отрицательными элементами:

```
Введите количество строк(от 2 до 10): 2
Введите количество столбцов(от 2 до 10): 3
Ввод массива:
array[1][1] = 2
array[1][2] = -3
array[1][3] = -5
array[2][1] = -4
array[2][2] = 6
array[2][3] = 7
Вывод массива:
a[1][1] = 2, a[1][2] = -3, a[1][3] = -5,
a[2][1] = -4, a[2][2] = 6, a[2][3] = 7,
Наименьшее значение элемента в столбце 3 = -5.
```

```
Введите количество строк(от 2 до 10): 2
Введите количество столбцов(от 2 до 10): 3
Ввод массива:
array[1][1] = 1
array[1][2] = 1
array[1][3] = 1
array[2][1] = -4
array[2][2] = -6
array[2][3] = -8
Вывод массива:
a[1][1] = 1, a[1][2] = 1, a[1][3] = 1,
a[2][1] = -4, a[2][2] = -6, a[2][3] = -8,
Наименьшее значение элемента в столбце 3 = -8.
```