



МИНИСТЕРСТВО НАУКИ
И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«НОВОСИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»



**НГТУ
НЭТИ** | **Факультет прикладной
математики и информатики**

Кафедра прикладной математики
Практическое задание № 1
по дисциплине «Дискретные математические модели»

РЕШЕНИЕ СЛАУ В ЦЕЛЫХ ЧИСЛАХ

Студент ДЫЧКО АРСЕНИЙ
Группа ПММ-41

Преподаватель РОЯК МИХАИЛ ЭММАНУИЛОВИЧ

Новосибирск, 2025

1. Задание

Написать консольную программу, которая считывает систему уравнений, решает ее и выдает результат. Ввод и вывод -- стандартные потоки ввода-вывода. Формат входного файла: в первой строке 2 целых числа N M через пробел, $N < 15$, $M < 15$, N -- число уравнений, M -- число неизвестных. Далее N строк по $M+1$ целых чисел, свободный член -- последний (уравнения вида $ax + by + c = 0$).

Формат выходного файла: в первой строке 1 целое число K -- число свободных переменных. Далее M строк по $K+1$ чисел, свободный член последний.

Если решений нет, то в единственной строке текст NO SOLUTIONS.

2. Теоретическая часть

Рассмотрим нахождение целочисленных решений произвольной системы линейных диофантовых уравнений

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = c_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = c_2, \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = c_m, \end{cases} \quad (18)$$

где a_{ij} и c_i -- целые числа.

Для решения системы (18) вначале определим расширенную матрицу системы

$$B = \left(\begin{array}{c|c} A & -c \\ \hline I & 0 \end{array} \right), \quad (19)$$

где A -- матрица коэффициентов a_{ij} а системы (18) размером $m \times n$, I -- единичная матрица размером $n \times n$, а c -- вектор из коэффициентов c_i системы (18), размерности m .

С первыми n столбцами матрицы B можно производить следующие действия:

- переставлять их;
- вычитать из одного столбца другой, умноженный на целое число.

Также можно вычитать один из первых n столбцов из последнего и переставлять какие-либо из первых m строк матрицы B .

С помощью этих действий преобразуем матрицу B таким образом, чтобы верхние m строк имели трапецевидный вид

$$B = \left(\begin{array}{cccccccc} \tilde{a}_{11} & 0 & \dots & 0 & 0 & \dots & 0 & 0 \\ \tilde{a}_{21} & \tilde{a}_{22} & \ddots & & \vdots & & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \vdots & & \vdots & \vdots \\ \tilde{a}_{k1} & \tilde{a}_{k2} & \dots & \tilde{a}_{kk} & 0 & \dots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots & \ddots & \vdots & \vdots \\ \tilde{a}_{m1} & \tilde{a}_{m2} & \dots & \tilde{a}_{mk} & 0 & \dots & 0 & 0 \\ b_{11} & b_{12} & \dots & b_{1k} & b_{1,k+1} & \dots & b_{1n} & f_1 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nk} & b_{n,k+1} & \dots & b_{nn} & f_n \end{array} \right). \quad (20)$$

Если на каком-либо шаге невозможно сделать элемент последнего столбца нулевым, то система (18) не имеет решений в целых числах.

В случае, если матрицу удалось преобразовать к виду (20), общее решение системы (18) можно записать в виде

$$x_i = f_i + t_1 b_{i,k+1} + \dots + t_{n-k} b_{i,n} \quad (21)$$

3. Тестирование

СЛАУ	Входной файл	Выходной файл
$x = 1$	1 1 1 1	0 1
$3x = 5$	1 1 3 5	NO SOLUTIONS
$2x + 2y = 5$	1 2 2 2 5	NO SOLUTIONS
$x + y = 1$	1 2 1 1 1	1 -1 1 1 0
$2x + 3y = 5$	1 2 2 3 5	1 3 -5 -2 5
$4x + 8y = 16$	1 2 4 8 16	1 -2 4 1 0
$\begin{cases} x = 1 \\ y = 2 \\ z = 3 \end{cases}$	3 3 1 0 0 1 0 1 0 2 0 0 1 3	0 1 2 3
$\begin{cases} x + z = 1 \\ y = 2 \\ 2x + 2z = 2 \end{cases}$	3 3 1 0 1 1 0 1 0 2 2 0 2 2	1 -1 1 0 2 1 0
$\begin{cases} x + z = 1 \\ y - z = 2 \end{cases}$	2 3 1 0 1 1 0 1 -1 2	1 -1 1 1 2 1 0
$\begin{cases} x + z = 1 \\ y = 2 \\ x + z = 3 \end{cases}$	3 3 1 0 1 1 0 1 0 2 1 0 1 3	NO SOLUTIONS
$\begin{cases} x + z = 1 \\ y = 2 \end{cases}$	2 3 1 0 1 1 0 1 0 2	1 -1 1 0 2 1 0
$\begin{cases} x + y + z = 1 \\ 2x + 2y + 2z = 4 \end{cases}$	2 3 1 1 1 1 2 2 2 4	NO SOLUTIONS

4. Текст программы

```
#include <iostream>
#include <fstream>
#include <vector>
#include <cmath>

using namespace std;

class test
{
private:
    int n, m; //Число уравнений и число неизвестных
    vector<vector<int>> A;
    vector<vector<int>> A_b;
    int size; //Размерность матрицы nxm
    int k; //Число свободных переменных
public:
    void input(string name)
    {
        //ifstream input(name);

        cin >> n >> m;

        size = n + m;

        A.resize(size);
        A_b.resize(size);
        for (int i = 0; i < size; i++)
        {
            A[i].resize(m + 1);
            A_b[i].resize(m + 1);
        }

        for (int i = 0; i < n; i++) {
            for (int j = 0; j < m + 1; j++) {
                cin >> A[i][j];
                A_b[i][j] = A[i][j];
            }
            A[i][m] *= -1;
        }
        //input.close();

        for (int i = n, j = 0; i < size; i++, j++)
            A[i][j] = 1;
    }

    bool SLAE()
    {
        //Приводим матрицу к нижнему треугольному виду
        for (int i = 0; i < n; i++) {
            bool row_done = false;

            while (!row_done) {
                //Ищем минимальный элемент в строке
                bool finded = false;
                int min_value = 0;
                for (int j = i; j < m; j++) {
                    if (!finded) {
                        if (A[i][j] != 0) {
                            min_value = j;
                            finded = true;
                        }
                    }
                }
                if (!finded)
                    return false;
            }
        }
        return true;
    }
};
```

```

        if (abs(A[i][j]) < abs(A[i][min_value]) && A[i][j] != 0)
            min_value = j;
    }

    if (!finded && A[i][m] == 0) {
        A.erase(A.begin() + i);
        i--;
        n--;
        size--;
        continue;
    }

    if (finded) {
        //Вычитаем столбец с минимальным значением из остальных
        for (int j = i; j < m; j++)
            if (j != min_value && A[i][j] != 0) {
                int q = A[i][j] / A[i][min_value];
                for (int k = i; k < A.size(); k++)
                    A[k][j] -= A[k][min_value] * q;
            }

        //Если диагональный элемент обнулится, то меняем его местами с ве-
        дущим
        if (A[i][i] == 0)
            for (int k = i; k < size; k++)
                swap(A[k][i], A[k][min_value]);
    }

    //Проверяем все недиагональные элементы на наличие ненулевых
    bool in_diag = false;
    for (int j = i + 1; j < m; j++) {
        if (A[i][j] != 0) {
            in_diag = true;
            break;
        }
    }

    //Если такие не найдены, проверяем правую часть на отличие от нуля и
    невозможность обратить её в ноль путём вычета столбца
    if (!in_diag) {
        if (A[i][i] != 0) {
            if (A[i][m] % A[i][i] != 0) {
                return false;
            }
        }
        else {
            int q = A[i][m] / A[i][i];
            for (int k = i; k < A.size(); k++)
                A[k][m] -= A[k][i] * q;
        }
    }
    else
        return false;

    if (A[i][i] == 1) {
        int q = A[i][m];
        for (int k = i; k < A.size(); k++)
            A[k][m] -= A[k][i] * q;
    }

    //Если правая часть нулевая, заканчиваем обработку строки
    if (A[i][m] == 0)
        row_done = true;
    }
}
}

```

```

    return true;
}

bool CheckTest()
{
    k = m - n; //Число свободных переменных

    //Если число свободных переменных будет < 0, то нету решения
    if (k < 0)
        return false;

    vector<int> t_val(k);

    for (int i = 0; i < k; i++)
        t_val[i] = rand();

    vector<int> x;

    for (int j = n; j < m + n; j++)
        x.push_back(A[j][m]);

    for (int s = n, i = 0; s < size; s++, i++)
        for (int j = k - 1; j >= 0; j--)
            x[i] += A[s][m - (j + 1)] * t_val[j];

    vector<int> test(n);

    for (int i = 0; i < test.size(); i++)
        for (int j = 0; j < A_b[i].size() - 1; j++)
            test[i] += A_b[i][j] * x[j];

    // Сделаем проверку нормы разности вектора с его изначальным вектором правой
части
    int result = 0;

    for (int i = 0; i < test.size(); i++)
        result += sqrt((test[i] - A_b[i][m]) - (test[i] - A_b[i][m]));

    if (result != 0)
        return false;

    return true;
}

void output(string out, bool flag)
{
    //ofstream output(out);
    cout << endl;

    if (!flag) {
        cout << "NO SOLUTIONS";
    }
    else
    {
        cout << k << endl; // Выводим свободное число

        for (int i = n; i < size; i++) {
            for (int j = m - k; j < m + 1; j++) {
                cout << A[i][j] << " ";
            }
            cout << std::endl;
        }
    }
    //output.close();
}

```

```
};  
  
int main() {  
    test ss;  
  
    ss.input("input.txt");  
    bool flag = ss.SLAE();  
  
    if(!flag)  
        ss.output("output.txt", flag);  
  
    flag = ss.CheckTest();  
  
    ss.output("output.txt", flag);  
  
    return 0;  
}
```