#include<iostream>

#include <stdio.h>

#include <math.h>

#include <stdlib.h> // Описания функций malloc

// Прототип функции приведения матрицы к ступенчатому виду. Функция возвращает ранг матрицы

int gauss(

int m, // Число строк матрицы

int n, // Число столбцов матрицы

double\* a, // Адрес массива элементов матрицы

double eps // Точность вычислений

);

int main() {

setlocale(LC\_ALL, "rus");

int m, n, i, j, rang;

double\* a;

double eps, det;

printf("Введите размеры матрицы m, n: ");

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

// выделение памяти под элементы матрицы

a = (double\*)malloc(m \* n \* sizeof(double));

printf("Введите элементы матрицы:\n");

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

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

// Вводим элемент с индексами i, j

scanf\_s("%lf", &(a[i \* n + j]));

}

}

printf("Введите точность вычислений eps: ");

scanf\_s("%lf", &eps);

// Вызываем метод Гаусса

rang = gauss(m, n, a, eps);

// Ступенчатый вид матрицы

printf("Ступенчатый вид матрицы:\n");

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

// Печатаем i-ю строку матрицы

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

printf("%10.3lf ", a[i \* n + j]);

//Формат %10.3lf означает 10 позиций на печать числа, 3 знака после точки

}

printf("\n"); // Перевести строку

}

printf("Ранг матрицы = %d\n", rang);

return 0;

}

/\* Приведение матрицы к ступенчатому виду методом Гаусса с выбором максимального элемента в столбце.

Функция возвращает ранг матрицы\*/

int gauss(

int m, // Число строк матрицы

int n, // Число столбцов матрицы

double\* a, // Адрес массива элементов матрицы

double eps // Точность вычислений

) {

int i, j, k, l;

double r;

i = 0; j = 0;

while (i < m && j < n) {

/\* минор матрицы в столбцах 0..j-1 уже приведен к ступенчатому виду, и строк с индексом i-1 содержит ненулевой эл-т

в столбце с номером, меньшим чем j, Ищем максимальный элемент в j-м столбце, начиная с i-й \*/

r = 0.0;

for (k = i; k < m; ++k) {

if (abs(a[k \* n + j]) > r) {

l = k; // Запомним номер строки

r = abs(a[k \* n + j]); // и макс. эл-т

}

}

if (r <= eps) {

/\* Все элементы j-го столбца по абсолютной

величине не превосходят eps.

Обнулим столбец, начиная с i-й строки\*/

for (k = i; k < m; ++k) {

a[k \* n + j] = 0.0;

}

++j; // Увеличим индекс столбца

continue; // Переходим к следующей итерации

}

if (l != i) {

// Меняем местами i-ю и l-ю строки

for (k = j; k < n; ++k) {

r = a[i \* n + k];

a[i \* n + k] = a[l \* n + k];

a[l \* n + k] = (-r); // Меняем знак строки

}

}

/\*abs(a[i\*n + k]) > eps. Обнуляем j-й столбец, начиная со строки i+1, применяя элем. преобразования второго рода\*/

for (k = i + 1; k < m; ++k) {

r = (-a[k \* n + j] / a[i \* n + j]);

// К k-й строке прибавляем i-ю, умноженную на r

a[k \* n + j] = 0.0;

for (l = j + 1; l < n; ++l) {

a[k \* n + l] += r \* a[i \* n + l];

}

}

++i; ++j; // Переходим к следующему минору

}

return i; // Возвращаем число ненулевых строк

}