

Задача D Шары

Имя входного файла: balls.in
Имя выходного файла: balls.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В корзине лежит N шаров, покрашенных не более чем в M цветов. Цвета пронумерованы целыми числами от 1 до M , i -й шар окрашен в цвет C_i . Произведем следующие действия K раз:

1. вынем шар;
2. запишем номер его цвета;
3. отложим шар в сторону.

Вы должны посчитать, сколько различных последовательностей чисел может получиться в результате.

Формат входных данных

Первая строка входного файла содержит три целых числа: N , M и K ($1 \leq N \leq 200$, $1 \leq M \leq N$, $1 \leq K \leq N$). На следующей строке расположено N целых чисел C_1, \dots, C_N ($1 \leq C_i \leq M$). Все числа в строках разделены пробелами.

Формат выходных данных

Выходной файл должен содержать одно целое число без ведущих нулей – ответ к задаче.

Пример

balls.in	balls.out
5 2 3	7
1 2 1 2 2	

Пояснение к примеру

Могут получиться следующие последовательности:

(1, 1, 2), (1, 2, 1), (2, 1, 1), (1, 2, 2), (2, 1, 2), (2, 2, 1), (2, 2, 2).

Задача Е Прямые

Имя входного файла: lines.in
Имя выходного файла: lines.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дано N различных точек на плоскости: $A_1(x_1, y_1), A_2(x_2, y_2), \dots, A_N(x_N, y_N)$. Для каждой пары точек A_i и A_j ($i < j$) посчитаем количество точек A_k таких, что $k \neq i, k \neq j$ и A_k лежит на прямой, проходящей через точки A_i и A_j и сложим все эти количества. Вам нужно посчитать, что получится в результате.

Формат входных данных

Первая строка входного файла содержит одно целое число N ($1 \leq N \leq 1000$). На следующих N строках расположено по два целых числа – на i -й строке координаты точки $A_i(x_i, y_i)$. Координаты точек – целые числа в пределах от -10^6 до 10^6 . Все числа в строках разделены пробелами.

Формат выходных данных

Выходной файл должен содержать одно целое число – ответ к задаче.

Пример

lines.in	lines.out
4 0 0 1 0 0 1 1 1	0
5 0 0 2 0 0 2 2 2 1 1	6

Пояснение ко второму примеру

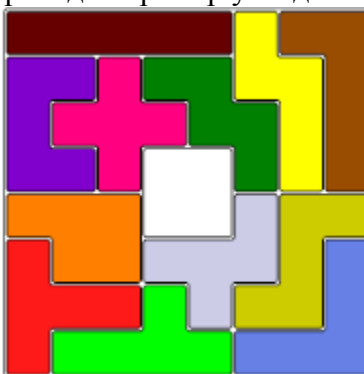
На прямых, проходящих через пары точек $(0, 0) - (2, 0), (0, 0) - (0, 2), (2, 0) - (2, 2), (0, 2) - (2, 2)$ не лежит ни одной точки, кроме этих точек.

На прямых, проходящих через пары точек $(0, 0) - (2, 2), (0, 0) - (1, 1), (2, 0) - (0, 2), (2, 0) - (1, 1), (0, 2) - (1, 1), (2, 2) - (1, 1)$ лежит по одной точке.

Задача F Пентомино

Имя входного файла: pento.in
Имя выходного файла: pento.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Вам дан набор пентомино – разноцветных фигур, состоящих из единичных квадратов одного цвета, (реберно)связных по стороне. Необходимо найти число способов укладки набора пентамино на прямоугольную доску так, чтобы все клетки доски были покрыты, никакие два пентомино не перекрывались и клетки пентомино не выходили за пределы доски. Два способа укладки считаются разными, если какая либо клетка доски покрыта разными пентомино из набора. Пентомино можно только вращать на 0, 90, 180, 270 градусов, но нельзя переворачивать, инвертировать и т.д. На рисунке приведен пример укладки 13 пентомино на доске 8x8.



Формат входных данных

В первой строке входного файла содержатся размеры доски – целые числа – количество строк n и количество столбцов m ($1 \leq n, m \leq 10$). Во второй строке находится единственное целое число – количество пентомино в наборе ($1 \leq k \leq 24$). Далее следуют k блоков, каждый из которых содержит описание пентомино. В первой строке блока находятся два целых числа – количество строк h и количество столбцов w в описании пентамино. Каждая из следующих h строк содержит ровно w символов – либо '.', либо латинская заглавная буква – цвет пентомино. В описании одного пентомино не может быть двух разных букв. Каждая буква участвует в описании не более 1 пентомино.

Формат выходных данных

Если число способов укладки пентамино превосходит 10000, выведите строку 'TOO MANY'. В противном случае выведите это число.

Пример

<i>pento.in</i>	<i>pento.out</i>	<i>Комментарий</i>	
2 2 2	4	AB	BB
1 1 A		BA	BB
2 2 BB B.		BB	BA

