

## Задача А. Непутевый граф

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Вам дан неориентированный граф из  $n$  вершин и  $m$  ребер. От вас требуется посчитать, сколько в нем есть путей реберной длины ровно  $k$ , по модулю  $10^9 + 7$ .

### Формат входных данных

В первой строке подаются три числа  $n, m, k$  ( $1 \leq n, m \leq 100, 1 \leq k \leq 10^4$ ) — число вершин в графе, число ребер в графе, ограничение на длину путей. В следующих  $m$  строках вводятся пары чисел  $a, b$  ( $1 \leq a, b \leq n$ ) — вершины, соединенные очередным ребром. Разрешаются кратные ребра и петли.

### Формат выходных данных

Выведите одно число — количество путей длины  $k$ . Не забудьте про взятие по модулю.

### Пример

стандартный ввод	стандартный вывод
3 3 3 1 2 2 3 1 3	24

## Задача В. Линейные уравнения

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Система линейных уравнений, как всем известно, есть множество уравнений

$$\begin{aligned} a_{11}x_1 + \dots + a_{1n}x_n &= b_1 \\ &\dots \\ a_{n1}x_1 + \dots + a_{nn}x_n &= b_n \end{aligned}$$

Ваша задача — решить её.

### Формат входных данных

В первой строке входного файла записано целое число  $n$  ( $1 \leq n \leq 20$ ). В следующих  $n$  строках записано по  $n + 1$  целых чисел:  $a_{i1}, \dots, a_{in}, b_i$ . Все эти числа не превышают 100 по абсолютному значению.

### Формат выходных данных

Первая строка выходного файла должна содержать одно из следующих сообщений:

- `impossible` — решений нет
- `infinity` — бесконечно много решений
- `single` — единственное решение. В этом случае вторая строка должна содержать  $n$  чисел  $x_1, \dots, x_n$ , разделенных пробелами. Решение должно быть выведено с точностью не менее трех знаков после десятичной точки.

### Примеры

стандартный ввод	стандартный вывод
2 1 1 1 2 2 2	infinity
2 1 2 0 1 2 1	impossible
2 1 2 1 2 1 0	single -0.333333333 0.666666667

## Задача С. Полные квадраты

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Множество целых положительных чисел будем называть *полноквадратным*, если произведение его элементов является полным квадратом (равно 1, 4, 9, 16, 25, 36, ...).

Задано множество  $A$ . Определите, сколько непустых подмножеств  $B$  множества  $A$  являются полноквадратными.

### Формат входных данных

В первой строке записано целое число  $N$  — количество элементов множества ( $1 \leq N \leq 100$ ). Во второй строке записаны  $N$  попарно различных чисел  $a_i$  — элементы множества ( $1 \leq a_i \leq 10^9$ ).

### Формат выходных данных

Выведите одно число — количество полноквадратных подмножеств по модулю 1 000 000 007.

### Пример

стандартный ввод	стандартный вывод
4 49 20 500 7	3

## Задача D. Двоичный Гаусс

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Дан набор из  $n$  битовых векторов размера  $n$  и еще один вектор. Получить этот вектор как хог исходных.

### Формат входных данных

В первой строке записано число  $n$  ( $1 \leq n \leq 300$ ). В следующих  $n$  строках записаны исходные вектора, в последней строке записан вектор, который нужно получить.

### Формат выходных данных

Если решений нет, выведите **No solution**. Если решений несколько, выведите **Multiple solutions**. Если решение единственное, выведите номера векторов (вектора нумеруются с 0), которые нужно сложить, чтобы получить данный вектор. Номера выведите в порядке возрастания.

### Примеры

стандартный ввод	стандартный вывод
3 100 111 101 010	1 2
3 100 111 011 010	No solution
3 111 010 101 000	Multiple solutions

## Задача E. Число возможных векторов

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Дан набор из  $m$  битовых векторов размера  $n$  ( $1 \leq n, m \leq 50$ ). Сколько различных векторов можно получить как хог этих векторов.

### Формат входных данных

В первой строке записаны числа  $n$  и  $m$ . В следующих  $m$  строках записаны вектора.

### Формат выходных данных

Выведите число векторов, которые можно получить

### Примеры

стандартный ввод	стандартный вывод
2 2 10 11	4
3 4 100 111 011 010	8
3 4 111 111 111 000	2

## Задача F. Совершенство

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Павел Андреевич очень любит все совершенное. Его новая любовь - совершенные паросочетания! Не огорчайте Павла Андреевича, проверьте, есть ли в данном графе совершенное паросочетание.

### Формат входных данных

В первой строке даны два числа  $N$  и  $M$  — количество вершин и количество ребер в  $G$ .  $N \leq 100$ . Следующие  $M$  строк содержат числа  $a_i$  и  $b_i$  — ребра графа. Гарантируется отсутствие петель и кратных ребер

### Формат выходных данных

Выведите слово «YES» или слово «NO» — ответ на поставленную задачу.

### Примеры

стандартный ввод	стандартный вывод
6 7 1 2 2 3 1 3 5 6 6 4 4 5 6 2	YES
3 3 1 2 2 3 1 3	NO

## Задача G. Игра со ксором

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Вам даны два человека, пронумерованные от 0 до 1, и переменная  $x$  изначально равная 0. Два человека играют в игру на протяжении  $N$  раундом. Во время раунда  $i$  происходит следующее:

- Игрок с номером  $S_i$  меняет  $x$  на  $x \oplus A_i$  либо пропускает ход.

Игрок 0 хочет добиться того, чтобы  $x = 0$  к концу игры, а игрок 1 хочет добиться того, чтобы  $x \neq 0$  к концу игры.

Определите, какой из игроков победит, при оптимальной игре обоих. Найдите ответ для  $T$  тестовых случаев.

### Формат входных данных

Первая строка входных данных содержит целое число  $T$  ( $1 \leq T \leq 100$ ). В первой строке каждого тестового случая содержится целое число  $N$  ( $1 \leq N \leq 200$ ). Во второй строке каждого тестового случая содержится последовательность  $(A_1, A_2, \dots, A_N)$  ( $1 \leq A_i \leq 10^{18}$ ). В третьей строке каждого тестового случая содержится строка  $S$ , состоящая из 0 и 1.

### Формат выходных данных

Для каждого тестового случая выведите 0, если  $x$  станет равным 0 к концу игры, выведите 1 в ином случае.

### Пример

стандартный ввод	стандартный вывод
3	1
2	0
1 2	0
10	
2	
1 1	
10	
6	
2 3 4 5 6 7	
111000	

## Задача Н. Число остовных деревьев

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Дан связный неориентированный граф. Найдите число его остовных деревьев по модулю  $10^9 + 7$ .

### Формат входных данных

Первая строка входного файла содержит натуральные числа  $n$  и  $m$  — количество вершин и ребер графа ( $1 \leq n \leq 100$ ,  $1 \leq m \leq 1000$ ). Далее следуют  $m$  строк, задающих ребра. Граф не содержит кратных ребер и петель.

### Формат выходных данных

Выведите число остовных деревьев по модулю  $10^9 + 7$ .

### Примеры

стандартный ввод	стандартный вывод
4 4 2 1 4 2 2 3 4 3	3
5 4 4 1 2 5 2 1 2 3	1

## Задача I. Электричество в каждый дом!

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

Известный чешский математик Отакар Борувка крайне увлечен проектированием электросети Моравии. В Моравии  $N$  городов, и различные строительные фирмы уже предложили Отакару  $M$  проектов построения ЛЭП между какими-то двумя городами. Известно, что в погоне за индивидуальностью и неповторимым строительным почерком, каждая фирма предлагает все свои проекты по одинаковой стоимости, отличной от стоимостей проектов других фирм. Также известно, что каждая фирма предлагает не более *трех* проектов. Можете считать, что фирмы достаточно сообразительны, чтобы не предлагать проектов соединения какого-то города с самим собой, но вполне может возникнуть ситуация, что одна или несколько фирм предлагают больше одного проекта соединения одной и той же пары городов.

Борувка собрал всех своих друзей и поручил им задачу спроектировать электросеть минимальной стоимости. Как вы, наверное, уже догадались, электрическая сеть является остовным деревом, а Борувку интересуют только сети, стоимость постройки которых минимальна.

Борувка всегда был уверен, что минимальное остовное дерево у графа одно, и представьте себе его удивление, когда каждый из друзей принес ему свой проект, утверждая что его-то дерево и есть минимальное. Подозревая неладное, он думает, что причиной разных ответов стали ребра одинакового веса. Помогите ему — посчитайте количество возможных электросетей минимальной стоимости, состоящих из ЛЭП, предложенных Борувке.

### Формат входных данных

В первой строке входного файла заданы два целых положительных числа  $N$  и  $M$ , не превосходящие 100 000.

Следующие  $M$  строк содержат по 3 целых числа каждая:  $1 \leq a_i, b_i \leq N$  и  $1 \leq c_i \leq 10^9$  — города, соединенные соответствующей ЛЭП, и ее стоимость. Гарантируется, что для любого числа  $c_i$  найдется не более трех ЛЭП, имеющих такую стоимость.

### Формат выходных данных

Выведите одно число — ответ на задачу. Так как это число может оказаться довольно большим, выведите остаток от деления на  $10^9 + 7$ .

## Примеры

countspans.in	countspans.out
2 2 1 2 1 2 1 1	2
3 5 1 2 2 2 3 3 3 1 3 3 1 1 2 3 1	1
5 10 2 5 8 4 5 8 2 4 3 4 2 3 4 2 3 5 4 6 5 1 6 1 3 5 3 1 5 4 5 7	6

## Задача J. Максимизировать сумму XOR

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	3 секунды
Ограничение по памяти:	512 мегабайт

Будем обозначать как  $\oplus$  операцию *побитового «исключающего или»* для целых чисел. В языках программирования C++ и Java она обозначается символом «^», в паскале и Python — ключевым словом «xor». Например,  $9 \oplus 3 = 1001_2 \oplus 11_2 = 1010_2 = 10$ .

Даны два массива  $A$  и  $B$  длины  $n$ . Обозначим как  $X(A)$  для массива  $A$  результат вычисления побитового «исключающего или» от всех элементов массива:  $X(A) = A_1 \oplus A_2 \oplus \dots \oplus A_n$ . Аналогично, введем обозначение  $X(B) = B_1 \oplus B_2 \oplus \dots \oplus B_n$ .

Для каждого  $i$  от 1 до  $n$  разрешается поменять местами элементы  $A_i$  и  $B_i$ . Необходимо определить, какие из этих обменов надо сделать, чтобы максимизировать сумму  $X(A) + X(B)$ .

### Формат входных данных

В первой строке входных данных находится число  $n$  — количество элементов ( $1 \leq n \leq 10^5$ ). В следующей строке находится  $n$  элементов массива  $A$  ( $0 \leq A_i \leq 10^{18}$ ). В следующей строке в таком же формате дан массив  $B$ .

### Формат выходных данных

В первой строке выведите максимальную возможную сумму и число  $k$  — количество необходимых обменов. В следующей строке выведите  $k$  различных чисел от 1 до  $n$  — индексы элементов, которые надо поменять.

### Пример

стандартный ввод	стандартный вывод
2	6 1
1 1	1
2 2	

### Замечание

В примере после обмена массивы равны  $A = [2, 1]$  и  $B = [1, 2]$ , соответственно.  
 $X(A) = 2 \oplus 1 = 10_2 \oplus 1_2 = 11_2 = 3$ ,  $X(B) = 3$ ,  $X(A) + X(B) = 6$ .

## Задача К. Дерево + дерево

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 1.5 секунд  
Ограничение по памяти: 256 мегабайт

Дан граф  $G$  из  $N$  вершин, пронумерованных от 1 до  $N$ . Изначально в графе 0 ребер. Вам даны две последовательности  $(u_1, u_2, \dots, u_M)$  и  $(v_1, v_2, \dots, v_M)$  длины  $M$ .

Вы выполняете следующую операцию  $N - 1$  раз:

- Выберите случайное  $i$  ( $1 \leq i \leq M$ ). Добавьте в  $G$  ребро, соединяющее вершины  $u_i$  и  $v_i$ .

Обратите внимание на то, что данная операция добавит ребро между вершинами, даже если оно уже присутствует в графе. Иными словами, в графе  $G$  могут быть кратные ребра.

Для каждого  $K = 1, 2, \dots, N - 1$  найдите вероятность того, что  $G$  будет деревом после выполнения  $K$  операций по модулю 998244353.

### Формат входных данных

В первой строке содержатся два целых числа  $N, M$ . ( $2 \leq N \leq 14, N - 1 \leq M \leq 500$ ) В следующих  $M$  строках содержатся пары целых чисел ( $1 \leq u_i, v_i \leq N$ ). Гарантируется, что  $u_i \neq v_i$ .

### Формат выходных данных

Выведите  $N - 1$  строку. В  $i$ -е строке должна содержаться вероятность того, что  $G$  является лесом после  $i$  операций по модулю 998244353.

### Примеры

стандартный ввод	стандартный вывод
3 2 1 2 2 3	1 499122177
4 5 1 2 1 2 1 4 2 3 2 4	1 758665709 918384805