

## Задача А. 17 стульев

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

64 мегабайта

Остап Бендер снова пытается получить причитающиеся драгоценности, но на этот раз они были заперты в шкатулке, для открытия которой необходимо иметь  $N$  ключей. По закономерной случайности каждый из ключей был спрятан в одном из  $N$  стульев, распроданных на недавнем аукционе. После аукциона эти стулья были развезены в  $N$  городов.

И вот теперь Остап решил на новую безумную затею: заехать в каждый из городов и, провернув в каждом из них аферу, выкрасть необходимые ключи. Чтобы избежать конфликтов с недоброжелателями, Остап не хочет больше одного раза появляться в каком-либо городе. Также у Остапа есть список цен за проезд между каждой парой городов. Изначально Остап находится в городе под номером 1 и после посещения всех городов может незаметно скрыться из этой страны.

Помогите Остапу найти порядок посещения городов, при котором ему потребуется потратить как можно меньше средств на странствия, и тогда, возможно, он поделится с Вами добытыми бриллиантами.

### Формат входных данных

Первая строка содержит единственное число  $N$  — количество городов ( $1 \leq N \leq 17$ ).

Следующие  $N$  строк содержат по  $N$  целых неотрицательных чисел.  $j$ -тое число в  $i$ -й строке означает стоимость проезда из города  $i$  в город  $j$  ( $0 \leq a_{ij} \leq 100$ ). Если  $a_{ij} > 0$ , то проезд стоит  $a_{ij}$  рублей, иначе — это означает, что из города  $i$  в  $j$  невозможно проехать напрямую.

### Формат выходных данных

В первой строке выведите минимальную сумму денег, необходимую для посещения всех городов Остапом. В следующей строке выведите  $N$  чисел — порядок посещения городов, при котором эта сумма достигается. Если затею Остапа невозможно вывести, то в единственной строке выходного файла выведите число -1.

### Примеры

<code>trader.in</code>	<code>trader.out</code>
3 0 3 2 3 0 6 2 6 0	8 1 3 2
5 0 6 4 0 0 6 0 7 0 7 4 7 0 0 0 0 0 0 0 2 0 7 0 2 0	20 1 3 2 5 4

## Задача В. Деловые встречи

Имя входного файла: `meetings.in`  
Имя выходного файла: `meetings.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

256 мегабайт

Алексей — успешный предприниматель, и в течение одного дня у него бывает много встреч с разными деловыми партнёрами. К сожалению, встречи бывают разные и не все приносят ему радость, после других же настроение улучшается. Также, на многие встречи не стоит приходить в слишком плохом или хорошем настроении — результат таких встреч может быть не таким, какой хочется Алексею.

К счастью, недавно Алексей научился оценивать своё настроение с помощью целых чисел. После этого для каждой встречи он оценил, при каком максимальном и минимальном настроении стоит на неё приходить, а также как изменится его настроение после этой встречи. Теперь он хочет распланировать порядок встреч так, чтобы в течение дня совершить максимальное число встреч.

Ваша задача — написать программу, которая по информации о всех встречах и настроении Алексея в начале дня находит порядок проведения встреч такой, что их количество при этом максимально.

### Формат входных данных

Первая строка входного файла содержит два целых числа  $n$  и  $k$  ( $1 \leq n \leq 20$ ,  $-100 \leq k \leq 100$ ) — количество встреч и настроение Алексея в начале дня.

Следующие  $n$  строк содержат по три целых числа  $a_i$ ,  $b_i$  и  $c_i$  ( $-100 \leq a_i, b_i, c_i \leq 100$ ) — минимальное и максимальное настроение, при котором встреча возможна, и изменение настроения по окончании встречи, соответственно.

### Формат выходных данных

В первой строке выходного файла выведите число  $m$  — максимально возможно число встреч. В следующей строке выведите  $m$  целых чисел — номера встреч в порядке их проведения. Встречи пронумерованы в порядке описания во входном файле.

Если ответов с максимальным числом встреч несколько, выведите любой.

### Примеры

<code>meetings.in</code>	<code>meetings.out</code>
3 0 1 3 3 0 1 2 1 3 1	3 2 3 1
3 1 -10 -5 3 -5 5 -2 -3 2 1	2 3 2

## Задача С. Последовательности из 0 и 1 без трех единиц подряд

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

По данному натуральному  $n$  определите количество последовательностей длины  $n$  из «0» и «1», не содержащих трех единиц подряд.

### Формат входных данных

В первой и единственной строке задано число  $n$ .  $1 \leq n \leq 10^9$ .

### Формат выходных данных

Выведите ответ по модулю  $10^9 + 7$ .

### Пример

стандартный ввод	стандартный вывод
3	7

## Задача D. Симпатичные узоры 2

Имя входного файла: nice2.in  
Имя выходного файла: nice2.out  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

Компания BrokenTiles планирует заняться выкладыванием во дворах у состоятельных клиентов узор из черных и белых плиток, каждая из которых имеет размер  $1 \times 1$  метр. Известно, что дворы всех состоятельных людей имеют наиболее модную на сегодня форму прямоугольника  $n \times m$  метров.

Однако при составлении финансового плана у директора этой организации появилось целых две серьезных проблемы: во первых, каждый новый клиент очевидно захочет, чтобы узор, выложенный у него во дворе, отличался от узоров всех остальных клиентов этой фирмы, а во вторых, этот узор должен быть симпатичным.

Как показало исследование, узор является симпатичным, если в нем нигде не встречается квадрата  $2 \times 2$  метра, полностью покрытого плитками одного цвета.

Для составления финансового плана директору необходимо узнать, сколько клиентов он сможет обслужить, прежде чем симпатичные узоры данного размера закончатся. Помогите ему!

### Формат входных данных

На первой строке входного файла находятся два натуральных числа  $n$  и  $m$ .  $1 \leq n \cdot m \leq 300$ .

### Формат выходных данных

Выведите в выходной файл единственное число — количество различных симпатичных узоров, которые можно выложить во дворе размера  $n \times m$  по модулю  $2^{30} + 1$ . Узоры, получающиеся друг из друга сдвигом, поворотом или отражением считаются различными.

### Примеры

nice2.in	nice2.out
2 2	14
3 3	322

## Задача E. $A + B = C$

Имя входного файла: `aplusb.in`  
Имя выходного файла: `aplusb.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Часто для пробного тура на различных олимпиадах по информатике предлагается задача « $A + B$ », в которой по заданным целым числам  $A$  и  $B$  требуется найти их сумму.

При проведении городской олимпиады по информатике председатель жюри решил сам подготовить тесты для такой задачи. Для этого он использовал свою оригинальную методику, которая заключалась в следующем: сначала готовятся предполагаемые правильные ответы, а затем подбираются входные данные, соответствующие этим ответам.

Пусть председатель жюри выбрал число  $C$ , запись которого состоит из  $n$  десятичных цифр и не начинается с нуля. Теперь он хочет подобрать такие целые положительные числа  $A$  и  $B$ , чтобы их сумма была равна  $C$ , и запись каждого из них также состояла из  $n$  десятичных цифр и не начиналась с нуля. В дополнение к этому председатель жюри старается подобрать такие числа  $A$  и  $B$ , чтобы каждое из них было красивым. Красивым в его понимании является число, запись которого не содержит двух одинаковых подряд идущих цифр. Например, число 1272 считается красивым, а число 1227 — нет.

Требуется написать программу, которая для заданного натурального числа  $C$  вычисляет количество пар красивых положительных чисел  $A$  и  $B$ , сумма которых равна  $C$ . Поскольку количество пар красивых чисел может быть большим, необходимо вывести остаток от деления этого количества на число  $10^9 + 7$ .

### Формат входных данных

Входной файл содержит одно целое положительное число  $C$ . Число  $C$  не начинается с нуля. Количество цифр в записи числа не превышает 100 000.

### Формат выходных данных

Выходной файл должен содержать одно целое число — остаток от деления количества искомым пар красивых чисел  $A$  и  $B$  на число  $10^9 + 7$ .

### Примеры

<code>aplusb.in</code>	<code>aplusb.out</code>
9	8
56	29
123	0

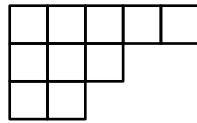
## Задача F. Увидеть Юнга и умереть

Имя входного файла:	young.in
Имя выходного файла:	young.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

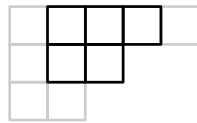
Диаграммы Юнга используются для того, чтобы изобразить разбиение числа на слагаемые. Разбиение числа  $n$  на слагаемые представляет собой сумму вида  $n = m_1 + m_2 + \dots + m_k$ , где  $m_1 \geq m_2 \geq \dots \geq m_k$ .

Диаграмма состоит из  $n$  квадратиков, организованных в виде  $k$  рядов, где  $k$  количество слагаемых в разбиении. Ряд, соответствующий числу  $m_i$ , содержит  $m_i$  квадратиков. Все ряды выровнены по левому краю и упорядочены от более длинного к более короткому.

Например, диаграмма Юнга, приведенная на рисунке, соответствует разбиению  $10 = 5 + 3 + 2$ .



Иногда можно вписать одну диаграмму Юнга в другую. Диаграмму  $X$  можно вписать в диаграмму  $Y$ , если можно удалить некоторые квадратики из диаграммы  $Y$  так, чтобы получилась диаграмма  $X$ . Отметим, что разрешается только удалять некоторые квадратики, вращать или отражать диаграмму не разрешается. Например, диаграмма для разбиения  $5 = 3 + 2$  может быть вписана в диаграмму для разбиения  $10 = 5 + 3 + 2$ , как показано на рисунке.



С другой стороны, диаграмму для разбиения  $8 = 4 + 4$  нельзя вписать в диаграмму для разбиения  $10 = 5 + 3 + 2$ .

Для заданного  $n$  найдите такое разбиение  $n$  на слагаемые, что в соответствующую ему диаграмму Юнга можно вписать максимальное количество различных диаграмм.

Например, в диаграмму для разбиения  $10 = 5 + 3 + 2$  можно вписать 36 различных диаграмм. Однако это не максимальное значение. В диаграмму для разбиения  $10 = 4 + 2 + 2 + 1 + 1$  можно вписать 41 диаграмму Юнга.

### Формат входных данных

Входной файл содержит целое число  $n$  ( $1 \leq n \leq 100$ ).

### Формат выходных данных

На первой строке выходного файла выведите максимальное число диаграмм Юнга, которые можно вписать в некоторую диаграмму, соответствующую разбиению на слагаемые числа  $n$ .

На второй строке выведите одно или более целых чисел — количество квадратиков в каждом из рядов оптимальной диаграммы.

### Пример

young.in	young.out
10	41 4 3 2 1

## Задача G. Леденящая игра

Имя входного файла: `game.in`  
Имя выходного файла: `game.out`  
Ограничение по времени: 3 секунды  
Ограничение по памяти: 256 мегабайт

256 мегабайт

Чтобы попасть в команду к Шкиперу пингвин должен пройти ряд испытаний: полоса препятствий от Шкипера, спарринг с Рико, расшифровка кода от Прапора и задача от Ковальски.

Вы, пингвин-новобранец, успешно дошли до последнего испытания. Ковальски предлагает вам сыграть в следующую игру. Вам дается  $m$  наборов разноцветных льдинок, каждая одного из  $n$  цветов. Различные цвета обозначаются различными прописными буквами латинского алфавита. Вы можете взять какое-то подмножество этих наборов при условии, что льдинка каждого цвета будет встречаться не более одного раза в этом подмножестве. Пусть вы выбрали  $k$  наборов с индексами  $i_1, i_2, \dots, i_k$ , тогда ваш выигрыш составляет  $\sum_{j=1}^k l_{i_j} - k$  баллов, где  $l_{i_j}$  — количество льдинок в наборе  $i_j$ .

Ковальски требует найти подмножество с максимальным количеством баллов.

От вас требуется найти любое подмножество, подходящее под условия Ковальски.

### Формат входных данных

В первой строке входного файла находится число  $n$  ( $1 \leq n \leq 17$ ) — количество различных цветов. Вторая строка входного файла содержит число  $m$  ( $1 \leq m \leq 200000$ ) — количество различных наборов льдинок. В следующих  $m$  строках перечислены сами наборы. Набор с номером  $i$  задаётся строкой из первых  $n$  строчных латинских букв. Длина каждой строки не больше 17 символов.

### Формат выходных данных

В первой строке выходного файла выведите  $k$  — количество наборов в ответе. Во второй строке выходного файла выведите  $k$  чисел — индексы наборов, входящих в ответ, в произвольном порядке.

### Примеры

<code>game.in</code>	<code>game.out</code>
1 3 aaa aaaa a	0
1 2 aaa aaaa	0
3 3 aba ab c	1 2

## Задача Н. Интересные числа

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

Софья считает число интересным, если его цифры идут в неубывающем порядке. Например, числа 123, 1111 или 888999 – интересные.

Софья заинтересовалась, сколько существует интересных положительных чисел, лежащих в диапазоне от  $L$  до  $R$  включительно. Это число может оказаться довольно большим для больших  $L$  и  $R$ , поэтому Софья хочет найти остаток от деления этого числа на  $10^9 + 7$ .

Требуется написать программу, которая по заданным  $L$  и  $R$  определяет количество интересных чисел, лежащих в диапазоне от  $L$  до  $R$  включительно, и выводит остаток от деления этого числа на  $10^9 + 7$ .

### Формат входных данных

Входной файл содержит две строки. Первая строка содержит число  $L$ , вторая строка содержит число  $R$  ( $1 \leq L \leq R \leq 10^{100}$ ).

### Формат выходных данных

Выходной файл должен одно целое число — остаток от деления количества интересных чисел, лежащих в диапазоне от  $L$  до  $R$  включительно, на  $10^9 + 7$ .

### Пример

стандартный ввод	стандартный вывод
1 100	54



## Задача I. Bricks. Кирпичная стена

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 512 мегабайт

64 мегабайта

Мы хотим построить стену высоты  $N$ , длины  $M$  и толщины 1. У нас есть бесконечное количество кирпичей  $1 \times 1 \times 2$  и  $1 \times 1 \times 3$ . Кирпичи можно класть только горизонтально, т.е. длинная сторона должна идти вдоль длины стены.

Рассмотрим общие вертикальные стороны кирпичей (т.е. находящиеся внутри стены). Для прочности стены должно выполняться следующее: ни для каких двух соседних слоев кирпичей не должно существовать общей вертикальной границы, проходящей через оба слоя. Например, такая стена:

2	2
2	2

не подходит под требования, поскольку граница посередине проходит через оба слоя.

Найдите количество вариантов выстроить стену, удовлетворяющую требованиям, по модулю  $10^9 + 7$ .

### Формат входных данных

В первой строке записаны числа  $N$  и  $M$  ( $1 \leq N \leq 10^{18}$ ,  $1 \leq M \leq 16$ ).

### Формат выходных данных

Выведите ответ на задачу по модулю  $10^9 + 7$ .

### Примеры

стандартный ввод	стандартный вывод
1 5	2
2 4	0