

Задача А. 17 стульев

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
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 решился 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

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

стандартный ввод	стандартный вывод
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. 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 1×1 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 2×2 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

<code>nice2.in</code>	<code>nice2.out</code>
2 2	14
3 3	322

Задача D. $A + B = C$

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 готовить 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 искомых 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 Е. Леденящая игра

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

стандартный ввод	стандартный вывод
1 3 aaa aaaa a	0
1 2 aaa aaaa	0
3 3 aba ab c	1 2

Задача F. 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 G. Трипростые числа

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

Будем называть натуральное число трипростым, если в нем любые подряд идущие 3 цифры образуют трехзначное простое число. Требуется по данному N найти количество N -значных трипростых чисел.

Формат входных данных

На вход подаётся одно натуральное число N : ($3 \leq N \leq 10^4$).

Формат выходных данных

Ответ должен содержать количество N -значных трипростых чисел, которое следует вывести по модулю $10^9 + 9$.

Примеры

стандартный ввод	стандартный вывод
3	143
4	204
4793	851557205

Задача Н. Разносчик пиццы

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

Вы подрабатываете разноской пиццы. У вас есть рюкзак размера S и огромный заказ на n пицц, i -я из которых имеет размер a_i . Разумеется, доставить пиццу требуется как можно скорее. К сожалению, у вас нет ни машины, ни друзей, которые могли бы помочь, так что единственный способ перевозки — распределить все пиццы в стопки размера не более S каждая и доставлять стопки по очереди. Вам надо распределить все пиццы из заказа в минимально возможное количество стопок.

Формат входных данных

Входной файл состоит из t тестов ($1 \leq t \leq 10$). Первая строка файла содержит число t , далее следуют описания тестов. Каждый тест описывается двумя строчками: на первой располагаются целые числа n ($1 \leq n \leq 20$) и S ($1 \leq S \leq 10^9$), на второй располагаются целые числа $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq S$).

Формат выходных данных

Для каждого теста выведите на отдельной строке минимальное число стопок m , а на следующих m строчках — описание стопок. i -я из последующих строк должна содержать количество пицц в i -й стопке k_i и список из k_i номеров пицц. Каждая пицца должна встречаться ровно в одной стопке. Если есть несколько оптимальных решений, выведите любое из них.

Пример

стандартный ввод	стандартный вывод
3	1
1 10	1 1
10	2
2 10	1 1
10 10	1 2
4 10	3
5 7 5 7	1 2
	2 1 3
	1 4

Задача I. Борской мой

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2.5 секунд
Ограничение по памяти:	256 мегабайт

Рик не сильно любит проводить время со своим внуком Морти вне приключений, однако иногда выделяет немного своего времени для игры в морской бой (разумеется, с настоящими космическими кораблями, а не на бумажке).

Поле для игры в Морской бой состоит из клеток и имеет ширину w и высоту h . Корабли могут состоять из одной, двух или трех подряд идущих клеток по вертикали или горизонтали. Всего на поле должно стоять s_1 одноклеточных кораблей, s_2 двухклеточных кораблей и s_3 трехклеточных. Корабли не должны касаться сторонами или пересекаться, однако они **могут** касаться углами.

Морти подозревает, что его гениальный дед хочет побыстрее выиграть, поэтому расставляет корабли по ходу игры. Он уже сделал несколько ходов, и на поле отмечены клетки, по которым он стрелял и результаты попаданий по ним: либо в этой клетке точно стоит корабль, либо корабля точно нет. Помогите ему определить, сколько возможных вариантов расстановки кораблей осталось у Рика. Поскольку это число может быть очень большим, найдите его по модулю $10^9 + 7$.

Формат входных данных

Первая строка содержит два целых числа w и h — ширину и высоту игрового поля, соответственно ($w \leq 100$; $h \leq 8$).

Следующие h строк содержат описания строк поля. Символ '.' обозначает клетку, по которой Морти не стрелял, 'o' обозначает клетку без корабля (промах), а 'x' — клетку с попаданием по кораблю.

Последняя строка содержит числа s_1 , s_2 и s_3 — количество кораблей каждого размера, которые надо разместить на поле ($s_1 \leq 5$; $s_2 \leq 4$; $s_3 \leq 3$).

Формат выходных данных

Выведите одно число — количество различных расстановок кораблей Рика, удовлетворяющих имеющимся данным, взятое по модулю $10^9 + 7$.

Две расстановки считаются различными, если существует клетка, занятая кораблем в одной расстановке, и свободная в другой.

Примеры

стандартный ввод	стандартный вывод
4 2 .ox. x.o. 2 1 0	2
3 3 .oo x.. .xx 0 2 0	1

Задача J. Защитный узор

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	512 мегабайт

Беверли прочитала в старой книге, которую нашла в библиотеке, что некоторые узоры могут отпугивать злые силы. Теперь она хочет нарисовать специальный узор на своей входной двери, чтобы временно отпугнуть Пеннивайза.

Входная дверь Беверли представляет собой клетчатый прямоугольник размера $n \times m$. Каждая клетка прямоугольника покрашена в белый или черный цвет. Беверли считает, что узор на двери будет отпугивать Пеннивайза, если:

- На двери будет хотя бы одна черная клетка
- Если соединить ребрами соседние по стороне черные клетки, в этом графе:
 - Будет одна компонента связности
 - Не будет существовать простого цикла

Беверли может перекрасить некоторые клетки на двери, при перекрашивании цвет клетки изменяется с белого на черный, и наоборот. При этом, она хочет закончить как можно быстрее, а поэтому хочет минимизировать количество перекрашиваний. Помогите ей найти любой узор, удовлетворяющий требуемым ограничениям, и требующий минимального возможного количества перекрашиваний клеток. Конечно же, Беверли будет перекрашивать каждую клетку не более одного раза.

Формат входных данных

В первой строке даны два целых числа n и m — высота и ширина двери ($1 \leq n \leq 100, 1 \leq m \leq 10$). В следующих n строках дано по m символов «.» и «#» — описание исходного узора на двери. Символ «.» соответствует белому цвету, а «#» — черному.

Формат выходных данных

Выведите любой узор, удовлетворяющий требуемым ограничениям, и требующий минимального количества перекрашиваний клеток.

Примеры

стандартный ввод	стандартный вывод
3 3 ### #.# ###	### #.# ##.
4 3 ##. .## ### ##.	##. .## ##. #..
2 3	... #..

Замечание

- В первом тесте Беверли потребуется перекрасить минимум одну клетку.
Во втором тесте Беверли потребуется перекрасить минимум две клетки.
В третьем тесте Беверли потребуется перекрасить минимум одну клетку.