

Задача А. 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 $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
1 100	54

Задача В. 17 стульев

Имя входного файла: `trader.in`
Имя выходного файла: `trader.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 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 -той 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 С. Деловые встречи

Имя входного файла: `meetings.in`
Имя выходного файла: `meetings.out`
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

<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 2 3

Задача D. Раскраска графа

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

Дан граф из n вершин, раскрасьте его в минимально возможное число цветов так, чтобы никакие две вершины, соединенные ребром, не были одного цвета.

Формат входных данных

В первой строке содержится число t — количество тестовых примеров ($1 \leq t \leq 5$).

Далее содержится t тестовых случаев, заданных в следующем формате:

В первой строке записаны числа n и m — количество вершин и ребер соответственно ($1 \leq n \leq 17$, $0 \leq m \leq \frac{n \cdot (n-1)}{2}$).

Затем идет m строк, в которых содержится по два числа v_i u_i , что означает, что вершины v_i и u_i соединены ребром ($1 \leq v_i, u_i \leq n, v_i \neq u_i$).

Гарантируется, что все ребра в каждом тестовом случае различны.

Формат выходных данных

Для каждого тестового случая в первой строке выведите минимальное число цветов k .

Во второй строке выведите n чисел a_i — цвета вершин ($1 \leq a_i \leq k$).

Примеры

стандартный ввод	стандартный вывод
3	3
3 3	3 2 1
1 2	2
2 3	1 2 2 1 1
3 1	3
5 3	1 3 1 1 2 1
2 1	
3 1	
4 2	
6 7	
1 2	
1 5	
2 5	
2 3	
2 4	
5 6	
5 4	

Задача Е. Выбери поменьше

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

Задано число k . Рассматривается множество всех натуральных чисел, имеющих сумму попарных произведений цифр, равную k (такие $m = \overline{a_l \dots a_1 a_0}$, что $\sum_{0 \leq i < j \leq l} a_i a_j = k$). Нужно найти минимальное число в этом множестве.

Формат входных данных

Во входном файле содержится одно целое число k ($1 \leq k \leq 5000$) — требуемая сумма попарных произведений цифр.

Формат выходных данных

В выходной файл должно быть выведено минимально возможное число (без лидирующих нулей и пробелов), имеющее сумму попарных произведений цифр, равную k . Если такого числа не существует, выведите -1 .

Примеры

<code>minimal.in</code>	<code>minimal.out</code>
3	13
58	2236

Задача F. Сеть

Имя входного файла: `network.in`
Имя выходного файла: `network.out`
Ограничение по времени: 1.3 секунды
Ограничение по памяти: 256 мегабайт

В компьютерной сети вашей фирмы n компьютеров. В последнее время свитч, к которому они подключены, сильно барахлит, и потому не любые два компьютера могут связаться друг с другом. Кроме того, если компьютер a обменивается информацией с компьютером b , то никакие другие компьютеры не могут в это время обмениваться информацией ни с a , ни с b . Вам необходимо вычислить максимальное количество компьютеров, которые могут одновременно участвовать в процессе обмена информацией.

Формат входных данных

В первой строке файла задано число n ($1 \leq n \leq 18$). Далее идут n строк по n символов, причём j -й символ i -й строки равен 'Y', если i -й и j -й компьютеры могут обмениваться информацией, иначе он равен 'N'. Верно, что i -й символ i -й строки всегда равен 'N' и, кроме того, матрица символов симметрична.

Формат выходных данных

Выведите максимальное количество компьютеров, которые могут одновременно участвовать в процессе обмена информацией.

Пример

<code>network.in</code>	<code>network.out</code>
5 NYYYY YNNNN YNNNY YNNNY YNYYN	4

Задача G. Сокращение многоугольника

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

Пусть задан некоторый выпуклый n -угольник. Сокращенным многоугольником назовем невырожденный выпуклый многоугольник, множество вершин которого является подмножеством множества вершин исходного многоугольника. Напомним, что многоугольник называется невырожденным, если его площадь больше нуля.

Ваша задача — найти сокращенный многоугольник наибольшей площади среди тех сокращенных многоугольников, периметр которых не превышает P .

Формат входных данных

Первая строка входного файла содержит количество вершин многоугольника n ($3 \leq n \leq 20$) и целое число P ($0 \leq P \leq 10^4$).

Каждая из последующих n строк содержит по два целых числа — координаты соответствующей вершины многоугольника. Вершины заданы в порядке обхода по часовой стрелке, их координаты не превосходят по модулю 100.

Никакие три последовательные вершины многоугольника не лежат на одной прямой.

Формат выходных данных

В выходной файл выведите наибольшую площадь искомого многоугольника. При проверке ответ будет считаться правильным, если он отличается от правильного не более, чем на 10^{-4} .

Если ни одного удовлетворяющего ограничениям сокращенного многоугольника не существует, выведите в выходной файл число 0.

Примеры

<code>reduce.in</code>	<code>reduce.out</code>
7 15 0 0 0 10 5 10 7 9 10 6 5 -2 3 -2	2.5
7 30 0 0 0 10 5 10 7 9 10 6 5 -2 3 -2	43.0

Задача Н. Ролевая игра

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

Когда у Жени бывает свободное время, он любит тратить его на свою любимую ролевую игру. Конечно же, у него есть мечта — достичь 255 уровня, то есть набрать необходимое количество очков опыта. К сожалению, у него не так много свободного времени, потому он хочет использовать его максимально эффективно и просит Вас помочь ему в этом.

Набирать очки опыта в игре можно посредством выполнения некоторых заданий. Для каждого задания известно время его выполнения, сколько очков оно приносит и список заданий, которые необходимо выполнить до того, как это задание станет доступно.

Помогите Жене выяснить, какое максимальное количество опыта, он сможет набрать за свободное время.

Формат входных данных

Первая строка входного файла содержит два целых числа n и m ($1 \leq n \leq 16, 1 \leq m \leq 1000$) — количество заданий и количество свободного времени у Жени в минутах. В следующих n строках следуют описания заданий по одному в каждой строке в следующем формате: сначала два целых числа t_i, p_i и k_i ($1 \leq t_i, p_i \leq 1000, 0 \leq k_i < n$) — время в минутах, которое занимает выполнение i -го задания, сколько опыта получит персонаж Жени при его выполнении, и количество заданий, которые необходимо выполнить до i -го, соответственно. Далее следуют k_i различных чисел от 1 до n — номера заданий, которые необходимо выполнить до i -го.

Формат выходных данных

В первой строке выведите единственное целое число — максимальное количество опыта, которое Женья может набрать за свободное время. В следующей строке выведите номера заданий, которые необходимо выполнить для получения такого количества опыта, в том порядке, в котором их необходимо выполнять.

Примеры

<code>rpg.in</code>	<code>rpg.out</code>
2 10 5 3 1 2 5 2 0	5 2 1
3 9 4 1 0 5 2 1 1 6 3 1 1	3 1 2