

Задача А. Сумма цифр в строке

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

Построим бесконечную последовательность строк S_0, S_1, S_2, \dots из двух символов — цифр ‘1’ и ‘2’ — следующим образом:

- $S_0 = "1"$;
- S_{i+1} получается из S_i одновременной заменой всех цифр ‘1’ на строки “11212”, а всех цифр ‘2’ — на “1121212”.

Так, $S_1 = "11212"$, $S_2 = \underbrace{11212}_1 \underbrace{11212}_1 \underbrace{1121212}_2 \underbrace{11212}_1 \underbrace{1121212}_2, \dots$

Заметим, что каждая строка содержит в качестве префиксов все предыдущие.

Определим S_∞ как бесконечную последовательность цифр, содержащую в качестве префиксов все строки S_i . Определим S_l^r как подстроку S_∞ , состоящую из всех её элементов с номерами от l до r , включительно. Например, $S_1^6 = "112121"$, $S_{17}^{17} = "2"$.

По данным числам l и r найдите сумму цифр в строке S_l^r .

Формат входного файла

Первая строка входного файла содержит целое число t ($1 \leq t \leq 50\,000$) — количество запросов. В следующих t строках записаны запросы; i -я из них содержит два целых числа l_i и r_i , разделённых одним пробелом ($1 \leq l_i \leq r_i \leq 10^9$).

Формат выходного файла

Выведите в выходной файл t чисел, по одному числу на строке — ответы на запросы в том порядке, в котором они заданы во входном файле.

Пример

digitsum.in	digitsum.out
2	8
1 6	2
17 17	

Задача В. Разрез пополам

Имя входного файла: `half.in`
Имя выходного файла: `half.out`
Ограничение по времени: 6 секунд
Ограничение по памяти: 64 мегабайта

Дан неориентированный граф из n вершин, где n чётно. Необходимо разбить вершины графа на два равных по размеру множества так, чтобы количество рёбер с концами в разных множествах было минимально.

Формат входного файла

Первая строка содержит два целых числа n и m — количество вершин и рёбер графа ($2 \leq n \leq 30$, n чётно).

Следующие m строк содержат описания рёбер — каждое ребро задано номерами своих концов. Вершины нумеруются с единицы. Гарантируется, что в графе нет кратных рёбер и петель.

Формат выходного файла

Выведите в первую строку выходного файла через пробел номера $\frac{n}{2}$ вершин, которые входят в одно из множеств оптимального разбиения вместе с первой вершиной. Вершины должны быть перечислены в порядке возрастания номеров. Если оптимальных разбиений несколько, то разрешается вывести любое.

Пример

half.in	half.out
6 8 1 2 6 1 2 3 5 2 2 6 4 3 4 5 6 5	1 2 6

Задача С. Следующая строка

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

Назовём строку из нулей и единиц *простой*, если она лексикографически меньше любого своего собственного суффикса. Например, строка «00101» простая, а «00000» — нет (любой её собственный суффикс меньше всей строки).

Необходимо по простой строке найти следующую в лексикографическом порядке простую строку такой же длины.

Формат входного файла

Входной файл содержит простую строку длины n ($2 \leq n \leq 10\,000$).

Формат выходного файла

В выходном файле должна находиться следующая в лексикографическом порядке простая строка длины n . Гарантируется, что она существует.

Пример

<code>next.in</code>	<code>next.out</code>
00111	01011