

Задача А. Счастливая перестановка

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

Петя любит счастливые числа. Всем известно, что счастливыми являются положительные целые числа, в десятичной записи которых содержатся только счастливые цифры 4 и 7. Например, числа 47, 744, 4 являются счастливыми, а 5, 17, 467 — не являются.

Однажды во сне Петя увидел лексикографически k -ую перестановку целых чисел от 1 до n . Определите, сколько счастливых чисел стоит в этой перестановке на позициях, номера которых также являются счастливыми числами.

Формат входных данных

В первой строке задано два целых числа n и k ($1 \leq n, k \leq 10^9$) — количество элементов в перестановке и номер перестановки в лексикографическом порядке.

Формат выходных данных

Если не существует k -ой перестановки чисел от 1 до n , выведите одно число «-1» (без кавычек). Иначе выведите ответ на задачу: количество таких i , что одновременно i , и a_i являются счастливыми числами.

Примеры

стандартный ввод	стандартный вывод
7 4	1
4 7	1

Задача В. Декартовы деревья

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

Недавно на лекции в университете Вася узнал, что такое декартово дерево. Декартово дерево — это двоичное дерево, в каждом узле которого хранятся два значения: ключ и приоритет. Оно является деревом поиска по множеству ключей и кучей на максимум по приоритетам, то есть

1. ключ любой вершины в левом поддереве вершины v меньше, чем ключ вершины v ;
2. ключ любой вершины в правом поддереве вершины v больше, чем ключ вершины v ;
3. приоритеты сыновей вершины v не больше, чем приоритет самой вершины v .

На контрольной работе Васе досталась следующая задача: дано n пар чисел вида (ключ, значение), i -я из которых выглядит как (i, y_i) , и нужно узнать, сколько существует способов построить декартово дерево, используя как ключ вершины i число i , а как приоритет — y_i . Поскольку это число может оказаться достаточно большим, требуется найти остаток от его деления на число $10^9 + 7$.

Формат входных данных

Первая строка содержит одно натуральное число t — число тестовых примеров во входных данных. Далее следуют описания тестов. Описание каждого теста состоит из двух строк. Первая строка содержит одно целое число n ($1 \leq n \leq 2 \cdot 10^5$) — количество вершин в дереве. Вторая строка содержит n целых чисел y_i ($1 \leq y_i \leq 10^9$) — приоритет i -й вершины дерева. Сумма n по всем тестам не превосходит $2 \cdot 10^5$.

Формат выходных данных

Для каждого теста в отдельной строке выведите одно целое число — количество различных декартовых деревьев, которые можно построить на данном наборе приоритетов, по модулю $10^9 + 7$.

Пример

стандартный ввод	стандартный вывод
2	1
4	10
2 4 1 3	
6	
7 3 3 1 1 3	

Задача С.

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

Определим по индукции множество \mathcal{R} *правильных скобочных последовательностей*:

- $\varepsilon \in \mathcal{R}$ (пустая строка)
- $A \in \mathcal{R} \Rightarrow (A) \in \mathcal{R}$
- $A \in \mathcal{R}, B \in \mathcal{R} \Rightarrow AB \in \mathcal{R}$

Пусть теперь \mathcal{R}_n — это множество правильных скобочных последовательностей из $2n$ символов — n открывающих и n закрывающих скобок.

Упорядочим элементы множества \mathcal{R}_n лексикографически с порядком символов $'(' < ')'$.

По данным числам n и p найдите p -ый в этом порядке элемент множества \mathcal{R}_n .

Формат входных данных

В первой строке входного файла заданы через пробел два целых числа n и p ($0 \leq n \leq 20$, $0 \leq p \leq 2 \cdot 10^9$). Скобочные последовательности нумеруются с нуля.

Формат выходных данных

В первой строке выходного файла выведите $2n$ символов без пробелов — p -ю правильную скобочную последовательность длины $2n$.

Если для данного n не существует p -я правильная скобочная последовательность, выведите в первой строке "N/A".

Задача D. Номер правильной последовательности

Имя входного файла:	<code>brackets2num2.in</code>
Имя выходного файла:	<code>brackets2num2.out</code>
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Во входном файле задана правильная скобочная последовательность с двумя типами скобок. Выведите в выходной ее номер в лексикографическом порядке среди всех правильных скобочных последовательностей с таким же количеством открывающихся скобок, '(', ')', '[', ']'. Последовательности занумерованы, начиная с 0. Количество открывающихся скобок в последовательности — от 1 до 20.

Формат входных данных

Во входном файле записана строка из круглых и квадратных скобок. Длина строки не превосходит 20 символов.

Формат выходных данных

Выведите одно число — ответ на задачу.

Задача Е. Новогодняя гирлянда

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

256 мегабайт

Дети в детском саду как-то раз решили повесить к Новому году гирлянду. Но это оказалось для них очень трудной задачей. На помощь пришёл Дед Мороз, который теперь каждый Новый год приносит с собой гирлянду и помогает её повесить.

Гирлянда представляет собой ломаную в плоскости, состоящую из n звеньев. Гирлянда начинается в точке $(0, 0)$, возле электророзетки и должна заканчиваться в точке $(n, 0)$. Число n называется длиной гирлянды. Каждое звено может располагаться либо горизонтально, либо под углом 45° к оси OX . Длина горизонтальной проекции любого звена равна 1. При этом не должно быть вершины ломаной с отрицательной координатой y , а также двух последовательных вершин с нулевой координатой y . Поднимающимся (опускающимся) назовём звено ломаной, у которого координата y правого конца больше (соответственно, меньше) координаты y левого конца. Звено, у которого координаты y концов совпадают, назовём горизонтальным. Обозначим поднимающееся звено буквой u , опускающееся — буквой d , а горизонтальное — буквой h . Тогда гирлянда кодируется строкой из n символов. У Деда Мороза есть волшебная книга, в которой перечислены все гирлянды длины n в виде строк. Хотя книга и волшебная, строки в ней располагаются в обычном лексикографическом порядке, по возрастанию. Дед Мороз отметил на полях книги галочкой гирлянду, которую повесил в прошлый раз. В этот Новый год он хочет повесить следующую в книге гирлянду. Найдите эту гирлянду без использования волшебной книги.

Формат входных данных

В первой строке вводится целое число n ($2 \leq n \leq 100\,000$). Во второй — строчка из n букв (все буквы: u , d , либо h) — прошлогодняя гирлянда.

Формат выходных данных

Выведите в виде строки гирлянду, которую Дед Мороз Павлович должен прихватить с собой в этот Новый год, либо `No solution`, если такой гирлянды не существует.

Пример

<code>garland.in</code>	<code>garland.out</code>
6 uhduhd	uhhdud

Задача F. Числа Каталана

Имя входного файла: `catalan.in`
Имя выходного файла: `catalan.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Посчитайте количество правильных скобочных последовательностей длины $2n$ (n открывающихся и n закрывающихся), составленных только из круглых скобок, по модулю $10^9 + 7$.

Формат входных данных

В единственной строке дано целое число n ($1 \leq n \leq 10^6$).

Формат выходных данных

Выведите остаток от деления количества искомых правильных скобочных последовательностей по модулю $10^9 + 7$.

Пример

<code>catalan.in</code>	<code>catalan.out</code>
3	5

Замечание

TL для Python: 3 секунды