

Задача А. Восстановление

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

Дарья обнаружила ошибку в своей программе, которая удаляет все символы из строки кроме "(" и ")". Оказывается, некоторые символы заменяются на что-то нечитаемое.

Теперь её заинтересовал вопрос, сколько различных правильных скобочных последовательностей длины $2n$ могут являться результатом исправленного алгоритма, то есть не будут противоречить данным, которые он так и потерял.

Формат входного файла

Единственная строка входного файла содержит строку из круглых скобок и знаков вопроса, где вопросами обозначены утраченные символы. Длина строки не превосходит 10000.

Формат выходного файла

Выведите одно число — количество различных скобочных последовательностей, удовлетворяющих шаблону Дарьи, по модулю $10^9 + 7$.

Пример

<code>recover.in</code>	<code>recover.out</code>
<code>(??())?</code>	2

Примечания

Вместо `ruthon3` при сдаче этой задачи используйте `руру3`.

Задача В. Неглубокие последовательности

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

Полина написала программу, удаляющую из строки все символы кроме "(" и ")". Теперь её заинтересовал вопрос, сколько различных правильных скобочных последовательностей длины $2n$ она может получить.

Известно, что Полина по политическим убеждениям запускает свою программу только на корректных математических выражениях, максимальная вложенность скобок в которых составляет в точности k .

Формат входного файла

Единственная строка входного файла содержит два числа n ($1 \leq n \leq 50$) и k ($1 \leq k \leq n$).

Формат выходного файла

Выведите одно число — искомое количество последовательностей по модулю $10^9 + 7$.

Примеры

	<code>deer.in</code>	<code>deer.out</code>
3	1	1
3	2	3
3	3	1

Задача С. Построение

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

Помогите Генриетте написать программу, генерирующую k -ю в лексикографическом порядке правильную скобочную последовательность, состоящую из $2n$ скобок.

Формат входного файла

В единственной строке через пробел записаны целые числа n и k , при этом $1 \leq n \leq 18$.

Формат выходного файла

Выведите искомую правильную скобочную последовательность. Гарантируется, что она существует.

Пример

	<code>build.in</code>	<code>build.out</code>
3	4	<code>()(())</code>

Задача D. Новогодняя гирлянда

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

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 входного файла

В первой строке вводится целое число n ($2 \leq n \leq 100000$). Во второй — строчка из n букв (все буквы: u , d , либо h) — прошлогодняя гирлянда.

Формат выходного файла

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

garland.in	garland.out
6 uhduhd	uhhdud

Задача Е. Удаление скобок — 2

Имя входного файла: `erase2.in`
Имя выходного файла: `erase2.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 64 мегабайта

Дана строка, составленная из круглых, квадратных и фигурных скобок. Определите, какое наименьшее количество символов необходимо удалить из этой строки, чтобы оставшиеся символы образовывали правильную скобочную последовательность.

Формат входного файла

Во входном файле записана строка из круглых, квадратных и фигурных скобок. Длина строки не превосходит 100 символов.

Формат выходного файла

Выведите строку максимальной длины, являющуюся правильной скобочной последовательностью, которую можно получить из исходной строки удалением некоторых символов. Если возможных ответов несколько, выведите любой из них.

Примеры

erase2.in	erase2.out
([]	[]
{ ([(] { }) }	([] { }
] { [

Задача F. Тесты к задаче “Проверка ПСП”

Имя входного файла: `--`
Имя выходного файла: `--`
Ограничение по времени: `--`
Ограничение по памяти: `--`

Разработайте систему тестов для задачи “Проверка ПСП” предыдущего дня.

Вам нужно сделать несколько тестов (от 1 до 30), удовлетворяющих условию задачи. К каждому тесту необходимо также указать правильный ответ. Тесты необходимо сохранить в файлах `001.dat`, `002.dat`, `003.dat` и т.д., правильные ответы на них — в файлах `001.ans`, `002.ans`, `003.ans` соответственно.

На проверку вам необходимо сдать архив в формате `zip` или `tar.gz`. Внутри этого архива должен быть каталог с именем `tests`. Внутри этого каталога должно быть не более 30 тестов и ответов на них. Все тесты должны строго соответствовать формату входных и выходных данных, описанных в условии задачи.

Сданный файл получает ОК, если он имеет правильный формат и структуру, содержит от 1 до 30 корректных тестов и правильные ответы на них (правильные решения выдают такие же ответы), а все неправильные решения не проходят хотя бы один тест из числа предложенных вами.

Для создания `zip`-архивов рекомендуется использовать архиваторы `7-zip` или `WinRAR`. Использовать встроенные средства Windows для создания `zip`-архивов нельзя, так как они создают некорректные архивы.