Задача A. Восстановление

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

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

Теперь его заинтересовал вопрос, сколько различных правильных скобочных последовательностей длины $2n$ могут являться результатом исправленного алгоритма, то есть не будут противоречить данным, которые он таки не 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 1000.

Формат выходного файла

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

Пример

recover.in	recover.out
(??()?	2

Задача B. Неглубокие последовательности

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

Вася написал программу, удаляющую из строки все символы кроме "(" и ")". Теперь его заинтересовал вопрос, сколько различных правильных скобочных последовательностей длины $2n$ он может получить.

Известно, что Вася 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 ($1 \leqslant n \leqslant 50$) и k ($1 \leqslant k \leqslant n$).

Формат выходного файла

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

Примеры

deep.in	deep.out
3 1	1
3 2	3
3 3	1

Задача C. Построение

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

Помогите Васе 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 $1 \leqslant n \leqslant 18$.

Формат выходного файла

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

Пример

build.in	build.out
3 4	()(())

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 Петрович, 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 назывется 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 у правого конца больше (соответственно, меньше) координаты у левого конца. Звено, у которого координаты у 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 Петровича 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 Петрович 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 ($2 \leqslant n \leqslant 100000$). 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 Петрович 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 E. Марсианские тарелки

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

В марсианском ресторане могут готовить n блюд, а воскресный обед состоит из l блюд. Некоторые из них могут появиться на воскресном обеде более одного раза, некоторые могут вообще не появиться. Во время обеда тарелки складываются одна

на другую. Официант ровно $2l$ раз заходит в зал и либо приносит очередное блюдо (после того, как съедено блюдо с верхней тарелки), либо забирает пустую верхнюю тарелку. Тем не менее для некоторых упорядоченных пар блюд существует марсианский обычай: первые из блюд не могут появиться в зале пока на столе стоят тарелки от вторых. Такие пары называются несовместимыми.

Назовем расписанием для официанта порядок, в котором он приносит и уносит тарелки. Таким образом, мы имеем $2l = t$ событий в расписании. Вы должны вычислить количество различных расписаний для воскресного обеда из l блюд по модулю p.

Формат входного файла

Первая строка входного файла содержит p ($2 \leqslant p \leqslant 10^4$), t ($1 \leqslant t \leqslant 200$) — количество событий в расписании, n ($1 \leqslant n \leqslant 10$) — количество блюд в ресторане и m ($1 \leqslant m \leqslant 100$) — количество несовместимых пар. Каждая из следующих m строк содержит упорядоченную пару чисел i и j, которые означают, что тарелка j не может быть принесена пока на столе стоит тарелка с номером i. Данное во входном файле число t четно.

Формат выходного файла

Выведите одно число — количество различных расписаний по модулю p.

Примеры

martian.in	martian.out
10000 4 2 1 1 2	7
9999 6 10 2 2 3 6 7	4866

Примечание к первому примеру. Если обозначить принос тарелки с блюдом 1 как "+1", а унос тарелки с блюдом 1, как "−1" и т.д., то возможны такие расписания:

$+1\ +1\ -1\ -1$
$+1\ -1\ +1\ -1$
$+2\ +2\ -2\ -2$
$+2\ -2\ +2\ -2$
$+1\ -1\ +2\ -2$
$+2\ -2\ +1\ -1$
$+2\ +1\ -1\ -2$