

## Задача А. Восстановление

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

Вася написал программу, которая удаляет все символы из строки кроме "(" и ")", но обнаружил, что некоторые скобки при этом программа заменяет 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 — остаток от деления на  $10^9 + 7$  количества различных скобочных последовательностей, удовлетворяющих Васиному шаблону.

### Пример

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

## Задача В. 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  $n$  ( $1 \leq n \leq 50$ ) и  $k$  ( $1 \leq k \leq n$ ).

### Формат выходного файла

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

### Примеры

<code>deep.in</code>	<code>deep.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 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  $1 \leq n \leq 18$ .

## 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 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  $n$  называется длиной гирлянды. Каждое звено может располагаться либо горизонтально, либо под углом  $45^\circ$  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  $y$  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  $n$  ( $2 \leq n \leq 100000$ ). 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 Петрович 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 uhhdud	uhhdud

## Задача E. Удаление скобок — 2

Имя входного файла: `erase.in`  
Имя выходного файла: `erase.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

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