

## Задача А. Проверка ПСП

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

Дана строка, состоящая из круглых, квадратных и фигурных скобок. Нужно проверить, является ли она правильной скобочной последовательностью.

### Формат входных данных

Во входном файле записана скобочная последовательность длиной не более 10000 символов.

### Формат выходных данных

Выведите **YES**, если скобочная последовательность является правильной, и **NO** в противном случае.

### Примеры

<code>check.in</code>	<code>check.out</code>
<code>([] ())</code>	<code>YES</code>
<code>([]]</code>	<code>NO</code>

## Задача В. Количество ПСП

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

Посчитайте количество правильных скобочных последовательностей длины  $2n$  ( $n$  открывающихся скобок и  $n$  закрывающихся), составленных из круглых и квадратных скобок так, что внутри любой пары круглых скобок нет квадратных скобок.

### Формат входных данных

В единственной строке через пробел записано целое неотрицательное число  $n$ , не превосходящее 1000.

### Формат выходных данных

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

### Примеры

	<code>quant.in</code>	<code>quant.out</code>
1	1	2
2	2	7

### Замечание

TL для Python: 2 секунды

## Задача С. Удалите скобки

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 скобок. Определите, какое наименьшее количество символов необходимо удалить из этой строки, чтобы оставшиеся символы образовывали правильную скобочную последовательность.

### Формат входных данных

Во входном файле записана строка из круглых скобок. Длина строки не превосходит 100 000 символов.

### Формат выходных данных

Выведите единственное целое число — ответ на поставленную задачу.

### Примеры

<code>erase.in</code>	<code>erase.out</code>
<code>()()</code>	2
<code>)()((</code>	5
<code>((())</code>	0

## Задача D. Культорги и скобки

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

Встретились однажды три культорга ЛКШ ...

- Первый культорг написала мелом на доске скобочную последовательность.
- Второму культоргу стало интересно, существует ли циклический сдвиг, превращающий эту последовательность в правильную.
- Третий же культорг, немного подумав, сказал, сколько таких сдвигов существует.

Вам известна скобочная последовательность, записанная первым культоргом. Найдите число, которое произнёс третий культорг.

Циклическим сдвигом строки называется перенос некоторого (возможно, нулевого) количества символов из конца строки в её начало без изменения их порядка.

### Формат входных данных

В единственной строке дана скобочная последовательность, записанная первым культоргом. Длина последовательности не равна нулю и не превышает 100000 символов.

### Формат выходных данных

Выведите количество циклических сдвигов, превращающих записанную скобочную последовательность в правильную.

### Примеры

<code>brackets.in</code>	<code>brackets.out</code>
<code>)((</code>	1
<code>)()(</code>	2
<code>()</code>	1

## Задача Е. Лексикографический порядок

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

Выведите все правильные скобочные последовательности заданной длины в лексикографическом порядке.

### Формат входных данных

Во входном файле записано натуральное число  $n$ , не превосходящее 10.

### Формат выходных данных

Выведите все правильные скобочные последовательности длины  $2n$  в лексикографическом порядке, по одной последовательности в строке.

### Примеры

lexical.in	lexical.out
3	((())) (()()) ()()() (())() (())()

## Задача F. Шифровка

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

Мюллер много раз пытался поймать Штирлица с полочным, но тот всё время выкручивался. Как-то раз Штирлиц просматривал электронную почту. В это время незаметно вошел Мюллер и увидел, как у него на экране появился бессмысленный набор символов.

“Шифровка”, — подумал Мюллер.

“UTF-8”, — подумал Штирлиц.

Известно, что Штирлиц шифрует текст простым алгоритмом: он многократно вставляет в произвольное место текста две одинаковые буквы. Вы должны восстановить исходный текст. В исходном тексте нет двух одинаковых букв подряд.

### Формат входных данных

В единственной строке записана шифровка Штирлица, состоящая из строчных латинских букв. Длина шифровки не превосходит 200000.

### Формат выходных данных

Выведите восстановленный текст.

### Примеры

<code>decode.in</code>	<code>decode.out</code>
<code>wwstdaadierfflitzzz</code>	<code>stierlitz</code>