

### Задача А. Тайны команды «Отбой»

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

Вы нашли секретный форум лидеров команды «Отбой». Конечно же, чтобы войти на форум, надо правильно ответить на ключевой вопрос. Ответ на секретный вопрос —  $k$ -ое по счёту красивое число на отрезке от  $a$  до  $b$  (нумерация с единицы). Напомним, что число  $x$  называется *красивым*, если сумма его цифр делится на  $n$ . Торопитесь, у вас есть всего две секунды на то, чтобы ответить на секретный вопрос, иначе вас рассекретят и выгонят из команды.

#### Формат входных данных

В единственной строке входного файла заданы четыре числа  $l, r, n, k$  ( $1 \leq l \leq r \leq 10^{18}, 1 \leq n \leq 1000, 1 \leq k \leq 10^{18}$ ).

#### Формат выходных данных

В выходной файл выведите  $k$ -ое красивое число на отрезке с  $l$  до  $r$ . Если такого числа не существует, то выведите -1.

#### Примеры

<code>kth-coolnumber.in</code>	<code>kth-coolnumber.out</code>
1 10 1 1	1
1 10 1 11	-1

### Задача В. Предыдущая правильная скобочная последовательность

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

По правильной скобочной последовательности вывести последовательность, предшествующую ей в лексикографическом порядке.

#### Формат входных данных

На вход подается одна строка, состоящая из  $2N$  ( $N \leq 10000$ ) символов - исходная правильная скобочная последовательность.

#### Формат выходных данных

Вывести в единственной строке искомую последовательность. Если исходная последовательность - первая в лексикографическом порядке, то следует вывести последнюю в лексикографическом порядке правильную скобочную последовательность такой же длины.

#### Примеры

<code>prev.in</code>	<code>prev.out</code>
<code>()()</code>	<code>(())</code>
<code>()(())</code>	<code>((()))</code>

### Задача С. Разбиения на слагаемые

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

Перечислите все разбиения целого положительного числа  $N$  на целые положительные слагаемые. Разбиения должны обладать следующими свойствами:

- Слагаемые в разбиениях идут в невозрастающем порядке.
- Разбиения перечисляются в лексикографическом порядке.

#### Формат входных данных

Во входном файле находится единственное число  $N$  ( $1 \leq N \leq 40$ ).

#### Формат выходных данных

В выходной файл выведите искомые разбиения по одному на строку.

#### Примеры

<code>partition.in</code>	<code>partition.out</code>
4	1 1 1 1 2 1 1 2 2 3 1 4

### Задача D. Номер по перестановке букв

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

*Перестановкой букв строки S* называются все различные строки, получающиеся из неё перестановкой букв.

Естественно, что все перестановки букв можно упорядочить лексикографически как строки. Например, для строки `abba` список упорядоченных перестановок букв будет выглядеть так: `aabb, abab, abba, baab, baba, bbaa`.

В этой задаче требуется найти лексикографический номер строки во множестве всех перестановок её букв (нумерация ведётся с нуля).

**Формат входных данных**

В первой строке входного файла дана строка  $S$ , состоящая не более чем из 20 маленьких букв латинского алфавита.

**Формат выходных данных**

В выходной файл выведите номер строки во множестве перестановок её букв.

**Примеры**

<code>stringr.in</code>	<code>stringr.out</code>
abab	1
baba	4