

## Задача А. Невозрастающая подпоследовательность

Имя входного файла: `subseq.in`  
Имя выходного файла: `subseq.out`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Вам требуется написать программу, которая по заданной последовательности находит максимальную невозрастающую её подпоследовательность (т.е. такую последовательность чисел  $a_{i_1}, a_{i_2}, \dots, a_{i_k}$  ( $i_1 < i_2 < \dots < i_k$ ), что  $a_{i_1} \geq a_{i_2} \geq \dots \geq a_{i_k}$  и не существует последовательности с теми же свойствами длиной  $k + 1$ ).

### Формат входных данных

В первой строке задано число  $n$  — количество элементов последовательности ( $1 \leq n \leq 239\,017$ ). В последующих строках идут сами числа последовательности  $a_i$ , отделенные друг от друга произвольным количеством пробелов и переводов строки (все числа не превосходят по модулю  $2^{31} - 2$ ).

### Формат выходных данных

Вам необходимо выдать в первой строке выходного файла число  $k$  — длину максимальной невозрастающей подпоследовательности. Во второй строке должны быть выведены все номера элементов исходной последовательности  $i_j$ , образующих искомую подпоследовательность. Номера выводятся в порядке возрастания. Если оптимальных решений несколько, разрешается выводить любое.

### Примеры

<code>subseq.in</code>	<code>subseq.out</code>
5	3
5 8 10 4 1	1 4 5

## Задача В. Собеседование в «Отбой»

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

Вы проходите собеседование в легендарную команду «Отбой». Вы успешно отвечали на все вопросы, и теперь от главного приза — членства в команде «Отбой», дающего доступ к безлимитным запасам кефирчика, вас отделяет последняя задача — посчитать количество чисел на отрезке с  $l$  по  $r$ , сумма цифр которых кратна числу  $k$ . Более формально — вам нужно посчитать количество чисел  $i$  ( $l \leq i \leq r$ ), у которых сумма цифр кратна  $k$ .

### Формат входных данных

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

### Формат выходных данных

В единственную строку выходного файла выведите ответ на задачу.

### Примеры

<code>coolnumbers.in</code>	<code>coolnumbers.out</code>
1 10 1	10
1 10 2	4

## Задача С. Футболки

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

На зарядку сегодня утром пришло  $N$  ЛКШат, они построились в ряд. Разумеется, ребята ходят в разноцветных футболках. Никита Павлович, наш многоуважаемый физрук, заметил, что можно попросить некоторых ребят присесть, и тогда для ребят, которые останутся стоять, будет выполнено следующее: последовательность цветов их футболок при перечислении слева направо будет такой же как и последовательность при перечислении справа налево, то есть будет палиндромом.

Например, если на зарядку пришли Ксюша в зеленой футболке, Слава в желтой, Артём в красной и Вася в зеленой, то можно попросить присесть Славу, тогда последовательность цветов будет «зеленый, красный, зеленый» как слева направо, так и справа налево. Аналогично можно попросить присесть Артёма (последовательность будет «зеленый, желтый, зеленый»), Славу и Артёма одновременно или любой набор из троих ребят. Таким образом, всего есть 7 способов добиться того, чтобы последовательность цветов была палиндромом.

Помогите Никите Павловичу найти количество способов попросить некоторых ЛКШат присесть, чтобы последовательность цветов футболок оставшихся стоять была палиндромом. Поскольку это число может быть очень большим, выведите его по модулю  $10^9$ .

### Формат входных данных

Первая строка входного файла содержит число  $N$  — количество ЛКШат, пришедших на зарядку ( $1 \leq N \leq 2000$ ). Вторая строка содержит  $N$  целых чисел, каждое из которых задает цвет футболки ЛКШонка и изменяется в пределах от 1 до  $10^9$ . Разные цвета задаются разными числами, а одинаковые — одинаковыми.

### Формат выходных данных

Выведите в выходной файл одно число — искомое количество способов по модулю  $10^9$ .

### Примеры

tshirts.in	tshirts.out
4 1 2 3 1	7

## Задача D. Шоколадка

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

Команда «Отбой» участвует в очередном марафоне по «Угадай мелодию. Rock version». Чтобы было чем подкрепиться во время игры, команда взяла с собой большую прямоугольную плитку шоколада размерами  $w \times h$ . У команды есть список из  $n$  пар чисел — размеры шоколадок, которые команда считает счастливыми. Прежде чем приступить к поеданию шоколадки, участники команды решили поделить имеющуюся плитку на счастливые шоколадки. Для этого они действуют следующим образом: сначала плитка шоколада ломается на 2 части по линии, строго параллельной одной из своих сторон, после чего каждую из полученных частей они могут продолжить ломать аналогичным образом.

Вам поручили определить, какое максимальное количество счастливых шоколадок команда сможет получить, действуя по данной схеме. Шоколадки, полученные поворотом счастливых, счастливыми не являются.

### Формат входных данных

В первой строке входного файла заданы три целых числа  $w, h, n$  — размеры плитки шоколада и количество вариантов размера счастливых шоколадок соответственно ( $1 \leq w, h \leq 300, 1 \leq n \leq w \times h$ ). В следующих  $n$  строках заданы пары целых чисел  $w_i, h_i$  — размеры счастливых шоколадок ( $1 \leq w_i \leq w, 1 \leq h_i \leq h$ ).

### Формат выходных данных

В единственную строку выходного файла выведите максимальное количество счастливых шоколадок, на которые можно разрезать данную плитку.

### Примеры

<code>chocolate.in</code>	<code>chocolate.out</code>
21 11 4 10 4 6 2 7 5 15 10	15
9 12 5 1 12 2 6 3 4 4 3 6 2	9

## Задача Е. Выбери поменьше

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

Задано число  $k$ . Рассматривается множество всех натуральных чисел, имеющих сумму попарных произведений цифр, равную  $k$  (такие  $m = \overline{a_l \dots a_1 a_0}$ , что  $\sum_{0 \leq i < j \leq l} a_i a_j = k$ ). Нужно найти минимальное число в этом множестве.

### Формат входных данных

Во входном файле содержится одно целое число  $k$  ( $1 \leq k \leq 5000$ ) — требуемая сумма попарных произведений цифр.

### Формат выходных данных

В выходной файл должно быть выведено минимально возможное число (без лидирующих нулей и пробелов), имеющее сумму попарных произведений цифр, равную  $k$ . Если такого числа не существует, выведите  $-1$ .

### Примеры

<code>minimal.in</code>	<code>minimal.out</code>
3	13
58	2236

## Задача F. Психотренинг

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

На очередном психологическом тренинге  $n$  участников сборов играют в занимательную игру. Участники игры рассаживаются по кругу и получают номера от 1 до  $n$  против часовой стрелки. После этого главный психолог отсчитывает против часовой стрелки  $k$ -го участника игры, начиная с первого. Этот участник выходит из круга и может идти на ужин. А остальные продолжают участие в тренинге. Главный психолог отсчитывает еще  $k$  участников, начиная со следующего после выбывшего. Участник, который оказался  $k$ -ым, тоже покидает тренинг, и так далее.

Участники сборов решили сесть в круг таким образом, чтобы один вредный тип пошел ужинать последним. Для этого они хотят установить, какой номер он должен для этого получить. Помогите им.

### Формат входных данных

Входной файл содержит два целых числа:  $n$  и  $k$  ( $1 \leq n \leq 10^{18}$ ,  $1 \leq k \leq 1000$ ).

### Формат выходных данных

Выведите в выходной файл одно число — номер участника, который пойдет на ужин последним.

### Примеры

<code>psyche.in</code>	<code>psyche.out</code>
5 3	4

## Задача G. Восстановление HTML-файла

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

Петя недавно скачал поврежденный HTML-файл. Он выглядит как обычный HTML-файл, но в нем есть некоторые несоответствия тэгов. Ваша задача — удалив минимальное количество открывающихся и закрывающихся тэгов, сделать так, чтобы структура тэгов стала правильной.

Более формально, HTML-файл состоит из произвольных символов с ASCII кодами из диапазона от 32 до 126, а также Linux-style переводов строки (символов с кодом 10). Тэги открываются следующим образом: `<Имя тэга Параметры>` и закрываются следующим образом: `</Имя тэга>`. Имя тэга — строка, состоящая из больших и маленьких латинских букв, которые считаются различными. Имя тэга отделяется от Параметров как минимум одним пробелом (но не переводом строки). Параметры могут содержать произвольные допустимые ASCII символы кроме `<`, `>` и переводов строк. Также допускаются открывающиеся тэги без параметров и записываются в следующей форме: `<Имя тэга>`.

HTML-файл считается правильным, если каждому открывающемуся тэгу можно привести в соответствие следующий далее в файле закрывающийся тэг таким образом, чтобы часть файла между этими тэгами также представляла собой правильный HTML-файл, и аналогично можно привести в соответствие каждому закрывающемуся тэгу ровно один открывающийся тэг, идущий ранее в файле. HTML-файл, не содержащий тэгов, также является правильным.

Открывающийся тэг считается соответствующим закрывающемуся, если у них одинаковое имя тэга.

### Формат входных данных

Входной файл представляется из себя поврежденный HTML-файл, который требуется исправить. Его длина не превосходит 10 000 байт. Количество открывающихся и закрывающихся тэгов не превышает 500. Символы `<` и `>` не встречаются нигде, кроме открывающихся и закрывающихся тэгов.

### Формат выходных данных

Выведите единственное целое число — минимальное количество открывающихся и закрывающихся тэгов, которые требуется удалить, чтобы файл оказался правильным HTML-файлом.

### Примеры

html.in	html.out
<pre>&lt;a href=http://kruzhok.newmail.ru&gt; &lt;b someone has corrupted this file&gt; It was a good file before... &lt;/a&gt; &lt;/b&gt;</pre>	2
<pre>&lt;a&gt;&lt;b&gt;That's good&lt;/b&gt;&lt;/a&gt;</pre>	0