

**Задача А. Гладиолус**

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

Числовая последовательность задана рекуррентной формулой:  $a_{i+1} = (ka_i + b) \bmod m$ .  
 Найдите ее наибольшую возрастающую подпоследовательность.

**Формат входного файла**

Программа получает на вход пять целых чисел: длину последовательности  $n$  ( $1 \leq n \leq 10^5$ ), начальный элемент последовательности  $a_1$ , параметры  $k$ ,  $b$ ,  $m$  для вычисления последующих членов последовательности ( $1 \leq m \leq 10^4$ ,  $0 \leq k < m$ ,  $0 \leq b < m$ ,  $0 \leq a_1 < m$ ).

**Формат выходного файла**

Требуется вывести наибольшую возрастающую подпоследовательность данной последовательности, разделяя числа пробелами. Если таких последовательностей несколько, необходимо вывести одну (любую) из них.

**Примеры**

gladiolus.in	gladiolus.out
5 41 2 1 100	41 67 71

**Note**

В данном примере последовательность состоит из 5 элементов:  $a_1 = 41$ ,  $a_{i+1} = (2a_i + 1) \bmod 100$ , то есть последовательность имеет вид 41, 83, 67, 35, 71.

**Задача В. Дубы**

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

На аллее перед зданием Министерства Обороны в ряд высажены  $n$  дубов. В связи с грядущим приездом главнокомандующего, было принято решение срубить несколько деревьев для придания аллее более милитаристического вида.

Внутренние распоряжки министерства позволяют срубить дуб только в двух случаях:

- если и ближайший дуб слева, и ближайший дуб справа строго ниже, чем данный дуб;
- если и ближайший дуб слева, и ближайший дуб справа строго выше, чем данный дуб.

В частности, согласно этому правилу, нельзя срубить крайний левый и крайний правый дубы.

Министр хочет выработать такой план вырубki, чтобы в итоге осталось несколько дубов, высоты которых образуют неубывающую последовательность, то есть чтобы каждый дуб был не ниже, чем все дубы, стоящие слева от него. При этом, как человек любящий флору, министр хочет, чтобы было срублено минимальное возможное количество деревьев.

Помогите сотрудникам министерства составить оптимальный план вырубki аллеи или выяснить, что срубить дубы соответствующим образом невозможно.

**Формат входного файла**

Первая строка входного файла содержит целое число  $n$  — количество дубов, растущих на аллее ( $2 \leq n \leq 200$ ). Вторая строка содержит  $n$  чисел — высоты дубов, приведенные слева направо. Высоты дубов — положительные целые числа, не превышающие 1 000.

**Формат выходного файла**

Если оставить последовательность дубов с неубывающими высотами невозможно, выходной файл должен содержать только одно число  $-1$ .

В случае, если искомый план существует, в первую строку выходного файла выведите целое число  $m$  — минимальное количество дубов, которые необходимо срубить. В следующие  $m$  строк выведите оптимальный план вырубki деревьев — номера дубов в том порядке, в котором их следует срубить, по одному номеру на строке.

Дубы нумеруются слева направо натуральными числами от 1 до  $n$ .

Если планов с наименьшим числом срубаемых дубов несколько, выведите любой из них.

**Примеры**

oaks.in	oaks.out
5	2
3 2 4 8 5	2
	4

**Задача С. Сумма кубов**

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

Напишите программу, которая находит минимальное натуральное число с заданной суммой кубов цифр.

**Формат входного файла**

Первая строка входного файла содержит требуемую сумму кубов цифр  $n$  ( $1 \leq n \leq 8\,000$ ).

**Формат выходного файла**

Выходной файл должен содержать одно число — ответ на задачу.

**Примеры**

sumcubes.in	sumcubes.out
1	1
9	12

## Задача D. Числа

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

Дана последовательность чисел  $a_1, a_2, \dots, a_n$ . За одну операцию разрешается удалить любое (кроме крайних) число, заплатив за это штраф, равный произведению этого числа на сумму соседних. Требуется удалить все числа, кроме крайних, с минимальным суммарным штрафом.

Пример начальной последовательности:

`1 50 51 50 1`

удаляем четвертое число, штраф  $50 \cdot (1 + 51) = 2600$ , получаем

`1 50 51 1`

удаляем третье число, штраф  $51 \cdot (50 + 1) = 2601$ , получаем

`1 50 1`

удаляем второе число, штраф  $50 \cdot (1 + 1) = 100$ .

Итого, штраф 5301.

### Формат входного файла

В первой строке входного файла расположено одно число  $n$  ( $1 \leq n \leq 100$ ) — количество чисел в последовательности.

Во второй строке находятся  $n$  целых чисел  $a_1, a_2, \dots, a_n$ ; никакое из чисел по модулю не превосходит 100.

### Формат выходного файла

Выведите в выходной файл одно число — минимальный суммарный штраф.

### Примеры

<code>numbers.in</code>	<code>numbers.out</code>
5 1 50 51 50 1	5301

## Задача E. 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 входного файла

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 выходного файла

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

### Примеры

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

## Задача F. Скобочная последовательность из двух типов скобок по номеру

Имя входного файла: `parenth2.in`  
 Имя выходного файла: `parenth2.out`  
 Ограничение по времени: 2 секунды  
 Ограничение по памяти: 64 мегабайта

*Правильной скобочной последовательностью из двух типов скобок* из  $2n$  скобок называется такая последовательность круглых и квадратных скобок, которая может встречаться в некотором арифметическом выражении. Например, `() [] ()` и `(( )) ()` являются пра-

вильными скобочными последовательностями из двух типов скобок, а  $((()))$  и  $([])$  [ — нет.

Все правильные скобочные последовательности из двух типов скобок можно упорядочить в лексикографическом порядке, считая, что порядок скобок соответствует их кодам символов:  $(' < ') < '[' < ']$ . Скажем, при  $n = 2$  список упорядоченных правильных скобочных последовательностей из двух типов скобок будет выглядеть так:  $(())$ ,  $(())$ ,  $()[]$ ,  $([])$ ,  $[(())]$ ,  $[][]$ ,  $[][]()$ ,  $[][][]$ .

В этой задаче требуется найти правильную скобочную последовательность из двух типов скобок по лексикографическому номеру (нумерация ведется с нуля).

### Формат входного файла

Два числа  $n$  и  $x$  ( $1 \leq n \leq 20$ ),  $x$  задает номер существующей правильной скобочной последовательности из двух типов скобок.

### Формат выходного файла

Выведите строку из  $2n$  круглых и квадратных скобок, задающих требуемую правильную скобочную последовательность из двух типов скобок.

### Примеры

	parenth2.in	parenth2.out
2 1		$(())$
2 4		$[(())]$