

## Задача А. Наибольшая общая подпоследовательность

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Даны две последовательности. Найдите длину их наибольшей общей подпоследовательности (подпоследовательность — это то, что можно получить из данной последовательности вычеркиванием некоторых элементов).

### Формат входных данных

В первой строке входного файла через пробел записаны  $N$  членов первой последовательности ( $1 \leq N \leq 1000$ ) — целых чисел, не превосходящих 10 000 по модулю. Во второй строке через пробел записаны  $M$  членов второй последовательности ( $1 \leq M \leq 1000$ ) — целые числа, не превосходящие 10 000 по модулю.

### Формат выходных данных

В первую строку выходного файла требуется вывести единственное целое число: длину наибольшей общей подпоследовательности или число 0, если такой не существует. Во вторую строку выходного файла требуется вывести самую наибольшую общую подпоследовательность, через пробел (если подпоследовательностей несколько, выведите любую).

### Примеры

стандартный ввод	стандартный вывод
1 2 3	2
2 1 3 5	2 3

## Задача В. 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 С. ЗОРП 1

Имя входного файла: knapsack-1.in  
Имя выходного файла: knapsack-1.out  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Перед вами лежат  $n$  котиков. Каждый котик характеризуется своим весом  $w_i$  и своей мимимишностью  $c_i$ . Вы хотите выбрать некоторое число котиков суммарным весом не более чем  $S$  так, чтобы их суммарная мимимишность была максимально возможной.

### Формат входных данных

В первой строке содержатся два целых числа  $n$  и  $S$  — число котиков и максимальный допустимый суммарный вес ( $1 \leq n \leq 100$ ,  $1 \leq S \leq 10^4$ ). Следующие  $n$  строк содержат по два целых числа  $w_i$  и  $c_i$  — вес и мимимишность  $i$ -го котика ( $1 \leq w_i \leq 10^4$ ,  $0 \leq c_i \leq 10^7$ ).

### Формат выходных данных

В первой строке выведите суммарную мимимишность выбранных котиков. Во вторую строку выведите целое число  $k$  — количество выбранных котиков. В третьей строке выведите  $k$  чисел — номера выбранных котиков. Если оптимальных ответов несколько, то разрешается вывести любой из них.

### Примеры

knapsack-1.in	knapsack-1.out
3 10	11
1 2	2
4 3	3 1
8 9	

## Задача D. Наибольшая общая возрастающая подпоследовательность

Имя входного файла: `lcis.in`  
Имя выходного файла: `lcis.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Даны две последовательности чисел —  $a$  и  $b$ . Нужно найти наибольшую общую возрастающую подпоследовательность. Более формально: такие  $1 \leq i_1 < i_2 < \dots < i_k \leq a.n$  и  $1 \leq j_1 < j_2 < \dots < j_k \leq b.n$ , что  $\forall t : a_{i_t} = b_{j_t}, a_{i_t} < a_{i_{t+1}}$  и  $k$  максимально.

### Формат входных данных

На первой строке целые числа  $n$  и  $m$  от 1 до 3 000 — длины последовательностей. Вторая строка содержит  $n$  целых чисел, задающих первую последовательность. Третья строка содержит  $m$  целых чисел, задающих вторую последовательность. Все элементы последовательностей — целые неотрицательные числа, не превосходящие  $10^9$ .

### Формат выходных данных

Выведите одно целое число — длину наибольшей общей возрастающей подпоследовательности.

### Примеры

<code>lcis.in</code>	<code>lcis.out</code>
6 5 1 2 1 2 1 3 2 1 3 2 1	2

## Задача Е. Учебная программа в ЛКШ

Имя входного файла: `lcs.in`  
Имя выходного файла: `lcs.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 8 мегабайта

Как известно, учебные программы третьей и четвертой параллели в ЛКШ очень похожи. Учебная программа параллели представляет из себя строку  $s$  из строчных латинских букв, где символ  $s_i$  записанный на  $i$ -й позиции кодирует тему, которая будет рассказана в  $i$ -й день. Будем считать, что учебная программа имеет размер, равный длине её строки. Например, учебная программа `aboba` размера 5 может обозначать, что в первый и пятый день будет рассказана *Геометрия*, во второй и четвертый день *FFT*, а в третий день *Дерево Фенвика*.

Если строку  $t$  можно получить удалением некоторого (возможно пустого) множества символов из строки  $s$ , то будем говорить, что  $t$  является *учебной подпрограммой* для учебной программы  $s$ . Например, для учебной программы `aboba`, учебными подпрограммами могут быть: `a`, `ab`, `aba`, и так далее.

Маленький мальчик Саша решил измерить *похожесть* учебных программ третьей и четвертой параллели. Для этого он сперва хочет определить максимальный размер некоторой учебной подпрограммы  $w$ , которая является *общей* для третьей и четвертой параллели (то есть учебная подпрограмма  $w$  может быть получена как из учебной программы третьей параллели, так и из учебной программы четвертой параллели, путем удаления некоторого множества символов). После этого, он определяет *похожесть* как два числа:

$$\frac{k}{n} \cdot 100, \quad \frac{k}{m} \cdot 100,$$

где  $k$  — размер максимальной общей учебной подпрограммы, а  $n$  и  $m$  — размеры учебных программ третьей и четвертой параллели соответственно.

### Формат входных данных

В первой и второй строке входных данных записаны строки из строчных латинских букв длины не более  $5 \cdot 10^3$  — учебные программы третьей и четвертой параллели. Так как в ЛКШ не могут быть только выходные дни, гарантируется, что строки не пусты.

### Формат выходных данных

Выведите через пробел два числа, характеризующие *похожесть* учебных программ третьей и четвертой параллели. Каждое число необходимо вывести с точностью **ровно** два знака после запятой.

### Примеры

<code>lcs.in</code>	<code>lcs.out</code>
<code>abc</code>	66.67
<code>bacd</code>	50.00
<code>abacabadabacaba</code>	26.67
<code>dbdccbdb</code>	50.00

## Задача F. Получите в сумме ноль

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 146 мегабайт

Даны целые неотрицательные числа  $a_1, a_2, \dots, a_n$ . Определите, какой знак следует поставить перед каждым из них, чтобы результат сложения данных чисел с учётом выбранных знаков равнялся нулю. Гарантируется, что способ расставить знаки таким образом существует.

### Формат входных данных

В первой строке записано целое число  $n$  ( $1 \leq n \leq 100$ ). Во второй строке через пробел записаны числа  $a_i$  ( $0 \leq a_i \leq 500$ ).

### Формат выходных данных

Выведите единственную строку из  $n$  символов.  $i$ -й символ строки должен соответствовать знаку, который требуется поставить в сумме перед  $i$ -м числом, чтобы получить в результате ноль.

### Примеры

stdin	stdout
4 2 3 0 1	-+--

## Задача G. Банкомат

Имя входного файла: `atm.in`  
Имя выходного файла: `atm.out`  
Ограничение по времени: 1 секунда  
Ограничение по памяти: 6 мегабайт

В некотором государстве в обращении находятся банкноты определенных номиналов. Национальный банк хочет, чтобы банкомат выдавал любую запрошенную сумму при помощи минимального числа банкнот, считая, что запас банкнот каждого номинала неограничен. Помогите Национальному банку решить эту задачу.

### Формат входных данных

Первая строка входных данных содержит натуральное число  $N$  не превосходящее 100 — количество номиналов банкнот в обращении. Вторая строка входных данных содержит  $N$  различных натуральных чисел  $x_1, x_2, \dots, x_N$ , не превосходящих  $10^6$  — номиналы банкнот. Третья строка содержит натуральное число  $S$ , не превосходящее  $10^6$  — сумму, которую необходимо выдать.

### Формат выходных данных

В первую строку выходного файла выведите минимальное число слагаемых (или -1, если такого представления не существует). Во вторую строку выведите это представление в любом порядке.

### Примеры

<code>atm.in</code>	<code>atm.out</code>
5 1 3 7 12 32 40	3 1 7 32

## Задача N. Разбиения на K слагаемых

Имя входного файла: стандартный ввод  
Имя выходного файла: стандартный вывод  
Ограничение по времени: 2 секунды  
Ограничение по памяти: 128 мегабайт

Для данных натуральных чисел  $N$  и  $K$  определите количество способов представить число  $N$  в виде суммы  $K$  натуральных слагаемых, если способы, отличающиеся только порядком слагаемых считать одинаковыми.

### Формат входных данных

Программа получает на вход два натуральных числа  $N$  и  $K$ , не превосходящих 400. Гарантируется, что ответ не превосходит  $2^{64} - 1$ .

### Формат выходных данных

Выведите ответ на задачу.

### Примеры

стандартный ввод	стандартный вывод
6 3	3
5 2	2

## Задача I. Число возрастающих подпоследовательностей

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

Задана последовательность из  $n$  чисел  $a_1, a_2, \dots, a_n$ . Подпоследовательностью длины  $k$  этой последовательности называется набор индексов  $i_1, i_2, \dots, i_k$ , удовлетворяющий неравенствам  $1 \leq i_1 < i_2 < \dots < i_k \leq n$ . Подпоследовательность называется возрастающей, если выполняются неравенства  $a_{i_1} < a_{i_2} < \dots < a_{i_k}$ .

Необходимо найти число возрастающих подпоследовательностей наибольшей длины заданной последовательности  $a_1, \dots, a_n$ . Так как это число может быть достаточно большим, необходимо найти остаток от его деления на  $10^9 + 7$ .

### Формат входных данных

Первая строка входного файла содержит целое число  $n$  ( $1 \leq n \leq 1000$ ). Вторая строка входного файла содержит  $n$  целых чисел:  $a_1, a_2, \dots, a_n$ . Все  $a_i$  не превосходят  $10^9$  по абсолютной величине.

### Формат выходных данных

В выходной файл выведите ответ на задачу.

### Примеры

subseq.in	subseq.out
5 1 2 3 4 5	1
6 1 1 2 2 3 3	8