Задача А. Наибольшая общая подпоследовательность

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 3 секунды
Ограничение по памяти: 256 мегабайт

Даны две последовательности, требуется найти и вывести их наибольшую общую подпоследовательность.

Формат входных данных

В первой строке входных данных содержится целое число n — длина первой последовательности ($1 \leq n \leq 2000$). Во второй строке заданы члены первой последовательности (через пробел) — целые числа, не превосходящие 10^9 по модулю. В третьей строке записано целое число m — длина второй последовательности ($1 \leq m \leq 2000$). В четвертой строке задаются члены второй последовательности (через пробел) — целые числа, не превосходящие 10^9 по модулю.

Формат выходных данных

В первой строке выведите длину наибольшей общей подпоследовательности, а во второй строке выведите через пробел самую наибольшую общую подпоследовательность данных последовательностей. Если ответов несколько — выведите любой.

Пример

стандартный ввод	стандартный вывод
3	2
1 2 3	2 3
4	
2 3 1 5	

Задача В. Наибольшая возрастающая подпоследовательность

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дана последовательность, требуется найти её наибольшую возрастающую подпоследовательность.

Формат входных данных

В первой строке входных данных задано целое число n — длина последовательности ($1 \leq n \leq 1000$). Во второй строке задается сама последовательность. Числа разделяются пробелом. Элементы последовательности — целые числа, не превосходящие 10^9 по абсолютной величине.

Формат выходных данных

В первой строке выведите длину наибольшей возрастающей подпоследовательности, а во второй строке выведите через пробел саму наибольшую возрастающую подпоследовательность данной последовательности. Если ответов несколько — выведите любой.

Пример

стандартный ввод	стандартный вывод
6	3
3 29 5 5 28 6	3 5 28

Задача С. Кино

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Трое друзей летом посмотрели несколько фильмов. Для каждого школьника известно, какие фильмы и в каком порядке он посмотрел (естественно, если фильм кому-то из ребят сильно понравился, он мог его пересмотреть несколько раз). Так как друзья любят смотреть фильмы вместе, втроем они посмотрели максимальное возможное количество. Сколько же раз они встречались вместе?

Формат входных данных

В первой строке входного файла заданы три числа: N , M и K — количества фильмов, просмотренных каждым из друзей ($1 \leq N, M, K \leq 300$). В следующих трех строках выписаны номера фильмов, просмотренных ими. У друзей огромные планы на будущее, поэтому фильмы нумеруются числами между 1 и 10^9 .

Формат выходных данных

В первой строке выведите единственное число — максимальное кол-во просмотренных фильмов.
В следующей строке выведите через пробел номера просмотренных фильмов.

Примеры

<code>stdin</code>	<code>stdout</code>
3 3 4	2
1 2 3	1 3
1 3 10	
3 1 10 3	

Задача D. Расстояние по Левенштейну

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дана текстовая строка. С ней можно выполнять следующие операции:

1. Заменить один символ строки на другой символ.
2. Удалить один произвольный символ.
3. Вставить произвольный символ в произвольное место строки.

Например, при помощи первой операции из строки «СОК» можно получить строку «ТОК», при помощи второй операции — строку «ОК», при помощи третьей операции — строку «СТОК».

Минимальное количество таких операций, при помощи которых можно из одной строки получить другую, называется стоимостью редактирования или расстоянием Левенштейна.

Определите расстояние Левенштейна для двух данных строк.

Формат входных данных

Программа получает на вход две строки, длина каждой из которых не превосходит 1000 символов, строки состоят только из заглавных латинских букв.

Формат выходных данных

Требуется вывести одно число — расстояние Левенштейна для данных строк.

Пример

стандартный ввод	стандартный вывод
ABCDEFGH ACDEXGHI	3

Задача Е. Пилообразная последовательность

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 4 мегабайта

Назовем последовательность пилообразной, если каждый ее элемент либо строго больше, либо строго меньше своих соседей. По данным числам n и k определите число пилообразных последовательностей длины n , составленных из чисел $1, \dots, k$.

Формат входных данных

Программа получает на вход два натуральных числа n и k , $1 \leq n \leq 1000$, $1 \leq k \leq 1000$.

Формат выходных данных

Необходимо вывести остаток от деления количества искомым последовательностей на $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
3 3	10
20 3	35422

Задача F. Гладиолус

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

64 мегабайта

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

Формат входных данных

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 выходных данных

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

Пример

<code>gladiolus.in</code>	<code>gladiolus.out</code>
5 41 2 1 100	41 67 71

Замечание

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 G. Рюкзак

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Дано n предметов массой $m_1, \dots, m_n$ и стоимостью $c_1, \dots, c_n$ соответственно.

Ими наполняют рюкзак, который выдерживает вес не более t . Определите набор предметов, который можно унести в рюкзаке, имеющий наибольшую стоимость.

Формат входных данных

В первой строке вводится натуральное число n , не превышающее 1000 и натуральное число t , не превышающее 10000.

Во второй строке вводятся n натуральных чисел m_i , не превышающих 100.

Во третьей строке вводятся n натуральных чисел c_i , не превышающих 100.

Формат выходных данных

В первой строке выведите количество предметов, которые нужно взять. Во второй строке выведите номера предметов (числа от 1 до n), которые войдут в рюкзак наибольшей стоимости.

Пример

стандартный ввод	стандартный вывод
4 6	3
2 4 1 2	1 3 4
7 2 5 1	

Задача N. Получите в сумме ноль

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
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.

Пример

стандартный ввод	стандартный вывод
4 2 3 0 1	-+--

Задача I. Банкомат

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

64 мегабайта

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.

Пример

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

Задача J. Рюкзак 0-1: минимум предметов

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дано N предметов массой $m_1, \dots, m_N$. Ими наполняют рюкзак, который выдерживает вес не более M . Как набрать вес в точности M , используя как можно меньше предметов?

Формат входных данных

В первой строке вводится натуральное число N , не превышающее 100 и натуральное число M , не превышающее 10000.

Во второй строке вводятся N натуральных чисел m_i , не превышающих 100.

Формат выходных данных

Выведите наименьшее необходимое число предметов или 0, если набрать данный вес невозможно.

Пример

стандартный ввод	стандартный вывод
1 5968 18	0

Задача К. ЗОРП 2

Имя входного файла: knapsack-2.in
Имя выходного файла: knapsack-2.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 ($1 \leq n \leq 100$, $1 \leq S \leq 10^9$). Следующие n строк содержат по два целых числа w_i и c_i — вес и мимимишность i -го котика ($1 \leq w_i \leq 10^7$, $0 \leq c_i \leq 10^4$). Гарантируется, что сумма всех c_i не превосходит 10^4 .

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.

Пример

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