

Задача А. НОПроблемо с восстановлением

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

Даны две последовательности. Найдите их наибольшую общую подпоследовательность (подпоследовательность — это то, что можно получить из данной последовательности вычёркиванием некоторых элементов).

Формат входных данных

В первой строке входного файла записано число N — длина первой последовательности ($1 \leq N \leq 10^3$). Во второй строке записаны члены первой последовательности (через пробел) — целые числа, не превосходящие 10^4 по модулю. В третьей строке записано число M — длина второй последовательности ($1 \leq M \leq 10^3$). В четвёртой строке записаны члены второй последовательности (через пробел) — целые числа, не превосходящие 10^4 по модулю.

Формат выходных данных

В первой строке выведите длину искомой подпоследовательности

Во второй строке выведите через пробел её элементы. Если есть несколько ответов, выведите любой.

Примеры

стандартный ввод	стандартный вывод
3 1 2 3 4 2 1 3 5	2 1 3
3 1 2 3 3 1001 1002 1003	0

Задача В. 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 ACDEXGIH	3

Задача С. НВП с восстановлением

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

Дана последовательность, требуется найти её наибольшую возрастающую подпоследовательность.

Формат входных данных

В первой строке входных данных задано число N — длина последовательности ($1 \leq N \leq 1000$). Во второй строке задается сама последовательность (разделитель — пробел). Элементы последовательности — целые числа, не превосходящие 10000 по модулю.

Формат выходных данных

В первой строке выведите длину НВП. В следующей строке требуется вывести наибольшую возрастающую подпоследовательность данной последовательности. Если таких подпоследовательностей несколько, необходимо вывести одну (любую) из них.

Пример

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

Задача D. НВП за \log

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

Числовая последовательность задана рекуррентной формулой: $a_{i+1} = (ka_i + b) \bmod m$. Найдите длину её наибольшей возрастающей подпоследовательности.

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

Выведите одно число — длину наибольшей возрастающей подпоследовательности.

Пример

стандартный ввод	стандартный вывод
5 41 2 1 100	3

Задача E. 0-1 рюкзак: наибольший вес

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

Дано n золотых слитков массой m_1, \dots, m_n . Ими наполняют рюкзак, который выдерживает вес не более w . Какую наибольшую массу золота можно унести в таком рюкзаке?

Формат входных данных

В первой строке вводится натуральное число n , не превышающее 100, и натуральное число w , не превышающее 10000.

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

Формат выходных данных

Выведите одно целое число — наибольшую возможную массу золота, которую можно унести в данном рюкзаке.

Примеры

стандартный ввод	стандартный вывод
2 3195 38 41	79
5 100 2 71 8 28 18	99

Задача F. Рюкзак максимальной стоимости

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 и стоимостью c_1, \dots, c_n соответственно. Ими наполняют рюкзак, который выдерживает вес не более w . Определите набор предметов, который можно унести в рюкзаке, имеющий наибольшую стоимость.

Формат входных данных

В первой строке вводится натуральное число n , не превышающее 100, и натуральное число w , не превышающее 10000.

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

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

Формат выходных данных

В первой строке выведите число k — количество элементов в рюкзаке наибольшей стоимости. Во второй строке выведите k различных чисел от 1 до n — номера предметов.

Пример

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

Задача G. Гирьки

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

Дан набор гирек массой m_1, \dots, m_n . Можно ли их разложить на две чаши весов таким образом, чтобы они оказались в равновесии?

Формат входных данных

Первая строка входных данных содержит натуральное число n , не превышающее 100. Далее идет n натуральных чисел m_i , не превышающих 100.

Формат выходных данных

Программа должна вывести «yes», если гирьки можно разложить на две кучки равной массы, или «no» в противном случае.

Пример

стандартный ввод	стандартный вывод
4 4 2 3 1	YES

Задача Н. Банкомат

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

Программа должна найти представление числа s в виде суммы слагаемых из множества x_i , содержащее минимальное число слагаемых, и вывести это представление на экран (в виде последовательности чисел, разделённых пробелами). Если таких представлений существует несколько, то программа должна вывести любое (одно) из них. Если такое представление не существует, то программа должна вывести -1 .

Пример

стандартный ввод	стандартный вывод
5 1 3 7 12 32 40	1 7 32

Задача I. Дизайнерский лифт

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

Дизайн-студия Артемия Индюкова получила заказ на разработку очень пафосного лифта для нового небоскреба. За работу взялся сам Артемий, отличающийся редкой неадекватностью. У него есть идея-фикс: для управления лифтом должно быть достаточно следующих четырёх кнопок:

- Подняться на A этажей вверх;
- Подняться на B этажей вверх;
- Подняться на C этажей вверх;
- Спуститься на первый этаж.

Изначально лифт находится на первом этаже. Пассажир лифта использует первые три кнопки, чтобы попасть на тот этаж, на который он хочет. Если пассажир пытается подняться вверх на A , B или C этажей, а такого этажа в здании не существует (то есть пассажир хочет подняться выше N -го, последнего, этажа), то лифт никуда не едет.

Заказчики проекта оказались с юмором и вместе с отказом от футуристичного дизайна решили оценить адекватность Артемия по шкале от 1 до N . Оценка адекватности равна количеству этажей, на которые можно попасть с первого с помощью такого лифта. Помогите им в этом.

Формат входных данных

Первая строка содержит число N — высоту небоскреба ($1 \leq N \leq 500000$).

Вторая строка содержит три числа A , B и C , задающие параметры кнопок ($1 \leq A, B, C \leq 100000$).

Формат выходных данных

Выведите единственное число — оценку адекватности Артемия Индюкова.

Пример

стандартный ввод	стандартный вывод
15 4 7 9	9

Задача J. Почти палиндромы

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

Слово называется *палиндромом*, если его первая буква совпадает с последней, вторая – с предпоследней и т.д. Например, слова «abba», «madam», «x» являются палиндромами.

Для заданного числа K слово называется *почти палиндромом*, если в нём можно изменить не более K букв так, чтобы получился палиндром. Например, при $K = 2$ слова «reactor», «kolobok», «madam» являются почти палиндромами, так как могут быть преобразованы в «reacaer», «kololok», «madam» заменой двух, одной и нуля букв соответственно.

Подсловом данного слова являются все непустые слова, получающиеся путем вычеркивания из данного нескольких (возможно, нуля) первых букв и нескольких последних. Например, подсловами слова «cat» являются слова «с», «а», «t», «са», «at» и само слово «cat» (а «ct» подсловом слова «cat» не является).

Требуется для данного числа K определить, сколько подслов данного слова S являются почти палиндромами.

Формат входных данных

В первой строке вводятся два натуральных числа: N ($1 \leq N \leq 5 \cdot 10^3$) – длина слова и K ($0 \leq K \leq N$).

Во второй строке содержится слово S , состоящее из N строчных латинских букв.

Формат выходных данных

Требуется вывести одно число – количество подслов слова S , являющихся почти палиндромами (для данного K).

Примеры

стандартный ввод	стандартный вывод
5 1 abcde	12
3 3 aaa	6

Задача К. НОВП

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

Найдите наибольшую общую возрастающую подпоследовательность двух последовательностей. В качестве ответа предъявите и длину этой подпоследовательности, и её элементы.

Формат входных данных

Во входном файле записаны две последовательности. Каждая последовательность описывается двумя строками следующим образом: в первой строке идет длина последовательности M ($1 \leq M \leq 500$), во второй идут M целых чисел a_i ($-2^{31} \leq a_i \leq 2^{31}$) — члены последовательности.

Формат выходных данных

В первой строке выходного файла выведите N — длину наибольшей возрастающей подпоследовательности. Во второй строке выходного файла выведите саму подпоследовательность.

Пример

стандартный ввод	стандартный вывод
5	2
1 4 2 5 -12	1 4
4	
-12 1 2 4	

Задача L. Лестница

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

Археологи раскопали Древний Храм, ко входу в который ведет лестница, шириной в 1 (один) метр, из ступенек различной длины и высоты. Лестница построена из каменных блоков $1 \times 1 \times 1$ метр. Археологи хотят для удобства туристов, чтобы лестница состояла из меньшего количества ступенек N . Для этого они могут также устанавливать каменные блоки $1 \times 1 \times 1$. Какое минимальное количество блоков необходимо, чтобы сделать лестницу в N ступенек, если известны начальная длина и высота каждой ступеньки? (То есть требуется выровнять какие-то ступеньки с помощью блоков, чтобы уменьшить их число). Высоты и длины ступенек новой лестницы могут различаться.

Формат входных данных

В первой строке через пробел заданы два целых числа M и N ($1 \leq N < M \leq 100$). Далее идут M строк, содержащих пару целых чисел L и H — длина и высота i -й ступеньки соответственно ($1 \leq L, H \leq 101$). Ступеньки нумеруются снизу вверх.

Формат выходных данных

Выведите единственное число — минимальное число блоков.

Пример

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

Задача М. 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 её элемент либо строго больше, либо строго меньше своих соседей. По данным числам n и k определите количество 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 искомых последовательностей на $10^9 + 7$.

Примеры

стандартный ввод	стандартный вывод
3 3	10
20 3	35422

Задача N. НВП с восстановлением за \log

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

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

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
5 41 2 1 100	3 41 67 71

Задача О. Максимальная по сумме возрастающая подпоследовательность

Имя входного файла: `stdin`
Имя выходного файла: `stdout`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Дана последовательность. Найти ее максимальную по сумме возрастающую подпоследовательность.

Формат входных данных

В первой строке дано одно число n ($1 \leq n \leq 5000$) — количество элементов последовательности. Далее во второй строке перечислены элементы последовательности (натуральные числа не превосходящие 10^9).

Формат выходных данных

Выведите максимальную по сумме подпоследовательность данной последовательности.

Примеры

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

Задача Р. Прямоугольное Пятно

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

Мало кто знает, но в Лего-вселенной есть свой суперзлодей по кличке Пятно. Только он не настолько известный, потому что он может создавать не красивые овальные «дыры» в ткани реальности, а только скучные прямоугольные.

Тем не менее, он, как и Пятно, которого мы знаем, стремится увеличить свою силу. Для этого он пытается создать как можно больше «дыр», попарно вложенных друг в друга, чтобы затем с помощью получившегося мега-портала впитать в себя энергию сразу большого числа коллайдеров из других вселенных.

Сейчас у него уже есть n прямоугольных «дыр», расположенных в одной плоскости. Их стороны параллельны осям координат, и эти «дыры» можно

- перемещать и накладывать друг на друга;
- поворачивать на 90° в рамках их общей плоскости (то есть менять местами их высоту и ширину).

Определите, какую максимальную по количеству «дыр» последовательность можно составить, чтобы каждая следующая «дыра» была вложена в предыдущую. Мы считаем, что «дыру» размера (h_1, w_1) можно вложить в «дыру» размера (h_2, w_2) , если $h_1 \leq h_2$ и $w_1 \leq w_2$.

Формат входных данных

В первой строке ввода дано целое число n — количество прямоугольных «дыр», из которых мы хотим составить последовательность вложенных ($1 \leq n \leq 10^5$).

В i -й из следующих n строк через пробел записаны два целых числа h_i и w_i — высота и ширина i -й «дыры» ($1 \leq h_i, w_i \leq 10^9$).

Формат выходных данных

В первой строке выведите максимальное количество «дыр», которые можно последовательно вложить друг в друга.

Во второй строке выведите через пробел их номера в 1-нумерации, в порядке от самого маленького до самого большого.

Если существует несколько вариантов ответа, выведите любой из них.

Примеры

стандартный ввод	стандартный вывод
5 1 1 3 2 2 5 4 1 3 5	4 1 4 3 5
5 1 10 2 9 3 8 4 7 5 6	1 1

Замечание

В примере можно повернуть четвертый прямоугольник на 90° , и получится последовательность прямоугольников с размерами $(1, 1)$, $(1, 4)$, $(2, 5)$ и $(3, 5)$. Нетрудно заметить, что каждый предыдущий можно вложить в следующий, а получить последовательность большей длины с тем же свойством не получится.