

Сортировка.

А. Сортировка выбором максимума

Требуется отсортировать массив по неубыванию методом "выбор максимума" (сортировка исключениями).

В первой строке вводится одно натуральное число N , не превосходящее 1000 — размер массива. Во второй строке задаются N чисел — элементы массива (целые числа, не превосходящие по модулю 1000).

Требуется вывести получившийся массив.

| Input | Output |
|----------|--------|
| 2 3 1 | 1 3 |

В. Сортировка вставками

Требуется отсортировать массив по неубыванию методом «вставок» (сортировка включениями). В алгоритме сортировки вставкой в каждый произвольный момент начальная часть списка уже отсортирована. В решении имеется цикл `for i in range(1, len(A))`, внутри которого в предположении, что элементы списка $A[0], A[1], \dots, A[k-1]$ уже отсортированы, элемент $A[k]$ добавляется в отсортированную часть списка.

В первой строке вводится одно натуральное число N , не превосходящее 1000 — размер массива. Во второй строке задаются N чисел — элементы массива (целые числа, не превосходящие по модулю 1000).

Требуется вывести получившийся массив.

| Input | Output |
|----------------|-----------|
| 5 5 4 3 2 1 | 1 2 3 4 5 |

С. Сортировка вставками — иллюстрация

Продемонстрируйте работу метода сортировки вставками по возрастанию. Для этого выведите состояние данного массива после каждой вставки на отдельных строках. Если массив упорядочен изначально, то следует не выводить ничего.

На первой строке дано число N ($1 \leq N \leq 100$) — количество элементов в массиве. На второй строке задан сам массив: последовательность натуральных чисел, не превышающих 10^9 .

В выходной файл выведите строки (по количеству вставок) по N чисел каждая.

| Input | Output |
|--------------|--------------------|
| 2 2 1 | 1 2 |
| 4 2 1 5 3 | 1 2 5 3 1 2 3 5 |

Д. Сортировка пузырьком

Требуется отсортировать массив по неубыванию методом "пузырька".

В первой строке вводится одно натуральное число N , не превосходящее 1000 — размер массива. Во второй строке задаются N чисел — элементы массива (целые числа, не превосходящие по модулю 1000).

Требуется вывести получившийся массив.

| Input | Output |
|----------------|-----------|
| 5 5 4 3 2 1 | 1 2 3 4 5 |

Е. Сортировка пузырьком — количество обменов

Определите, сколько обменов сделает алгоритм пузырьковой сортировки по возрастанию для данного массива.

На первой строке дано число N ($1 \leq N \leq 1000$) — количество элементов в массиве. На второй строке — сам массив. Гарантируется, что все элементы массива различны и не превышают по модулю 10^9 .

Выведите одно число — количество обменов пузырьковой сортировки.

| Input | Output |
|----------------|--------|
| 5 1 2 3 4 5 | 0 |
| 5 5 4 3 2 1 | 10 |

F. Поразрядная сортировка – иллюстрация

Поразрядная сортировка является одним из видов сортировки, которые работают за линейное от размера сортируемого массива время. Такая скорость достигается за счет того, что эта сортировка использует внутреннюю структуру сортируемых объектов. Изначально этот алгоритм использовался для сортировки перфокарт. Первая его компьютерная реализация была создана в университете MIT Гарольдом Сьюардом (Harold H. Seward). Опишем алгоритм подробнее.

Пусть задан массив строк s_1, \dots, s_i причем все строки имеют одинаковую длину m . Работа алгоритма состоит из m фаз. На i -ой фазе строки сортируются по i -ой с конца букве. Происходит это следующим образом. В этой задаче рассматриваются строки из цифр от 0 до 9. Для каждой цифры создается “корзина” (“bucket”), после чего строки s_i распределяются по “корзинам” в соответствии с i -ой с конца цифрой. Строки, у которых i -ая с конца цифра равна j попадают в j -ую корзину (например, строка 123 на первой фазе попадет в третью корзину, на второй — во вторую, на третьей — в первую). После этого элементы извлекаются из корзин в порядке увеличения номера корзины.

Таким образом, после первой фазы строки отсортированы по последней цифре, после двух фаз — по двум последним, ..., после m фаз — по всем. При этом важно, чтобы элементы в корзинах сохраняли тот же порядок, что и в исходном массиве (до начала этой фазы). Например, если массив до первой фазы имеет вид: 111, 112, 211, 311, то элементы по корзинам распределятся следующим образом: в первой корзине будет 111, 211, 311, а второй: 112. Ваша задача состоит в написании программы, детально показывающей работу этого алгоритма на заданном массиве. Первая строка входного файла содержит целое число n ($1 \leq n \leq 1000$). Последующие n строк содержат каждая по одной строке s_i . Длины всех s_i одинаковы и не превосходят 20. Все s_i состоят только из цифр от 0 до 9.

В выходной файл выведите исходный массив строк v , состояние “корзин” после распределения элементов по ним для каждой фазы и отсортированный массив. Следуйте формату, приведенному в примере.

| Input | Output |
|-------|------------------------------------|
| 9 | Initial array: |
| 12 | 12, 32, 45, 67, 98, 29, 61, 35, 09 |
| 32 | ***** |
| 45 | Phase 1 |
| 67 | Bucket 0: empty |
| 98 | Bucket 1: 61 |
| 29 | Bucket 2: 12, 32 |
| 61 | Bucket 3: empty |
| 35 | Bucket 4: empty |
| 09 | Bucket 5: 45, 35 |
| | Bucket 6: empty |
| | Bucket 7: 67 |
| | Bucket 8: 98 |
| | Bucket 9: 29, 09 |
| | ***** |
| | Phase 2 |
| | Bucket 0: 09 |
| | Bucket 1: 12 |
| | Bucket 2: 29 |
| | Bucket 3: 32, 35 |
| | Bucket 4: 45 |
| | Bucket 5: empty |
| | Bucket 6: 61, 67 |
| | Bucket 7: empty |
| | Bucket 8: empty |
| | Bucket 9: 98 |
| | ***** |
| | Sorted array: |
| | 09, 12, 29, 32, 35, 45, 61, 67, 98 |

G. Mergesort

Отсортируйте данный массив, используя сортировку слиянием. Можете использовать рекурсивную функцию или написать не рекурсивный алгоритм.

В этой задаче рекомендуется реализовать функцию `merge(x, a, b, c)` для слияния двух упорядоченных частей массива, в которой разрешается использовать вспомогательный массив. Здесь x - данный массив, который будет изменён в ходе работы функции, a, b, c — границы соседних упорядоченных частей массива.

| Input | Output |
|-------|--------|
| 2 | 1 3 |
| 3 1 | |

Н. Quicksort

Идея быстрой сортировки заключается в выборе некоторого элемента x массива A и последующем разбиении исходного массива на три части: первая состоит из элементов, меньших x , вторая — равных x и третья — больших x . Эту часть алгоритма лучше реализовать в виде функции. Затем рекурсивно вызывать функцию сортировки от первой и третьей частей.

Элемент, относительно которого происходит разбиение называется *опорным* (*pivot*). Задача была в листочке про массивы (доп. листок, задача С).

Скорость работы алгоритма существенно зависит от метода выбора опорного элемента.

В программе запрещается использовать вспомогательные массивы.

| Input | Output |
|----------|--------|
| 2 3 1 | 1 3 |

I. Сортировка подсчётом

Имеется список целых чисел размера N , сами числа по модулю не превосходят 10000. Отсортировать их за время $O(N)$. Разрешается использовать дополнительную память, объём которой пропорционален максимально возможному значению элемента массива.

| Input | Output |
|----------------|-----------|
| 5 1 3 4 2 5 | 1 2 3 4 5 |

В следующих задачах время работы программы должно быть пропорционально $N \cdot \log N$ или лучше.

Разрешается и поощряется использовать встроенную сортировку.

J. Результаты олимпиады

Во время проведения олимпиады каждый из участников получил свой идентификационный номер — целое неотрицательное число. Необходимо отсортировать список участников олимпиады по количеству набранных ими баллов.

На первой строке дано число N ($1 \leq N \leq 1000$) — количество участников. В следующих N строках даны идентификационный номер и набранное число баллов соответствующего участника. Во входных данных могут встретиться строки с совпадающими идентификационными номерами. Все числа во входном файле не превышают 10^5 .

В выходной файл выведите исходный список в порядке убывания баллов. Если у некоторых участников одинаковые баллы, то их между собой нужно упорядочить в порядке возрастания идентификационного номера.

| Input | Output |
|---------------------------------|----------------------------|
| 3 101 80 305 90 200 14 | 305 90 101 80 200 14 |
| 3 20 80 30 90 25 90 | 25 90 30 90 20 80 |

К. Похожие массивы

Назовём два массива похожими, если они состоят из одних и тех же элементов (без учёта кратности). По двум данным массивам выясните, похожие они или нет.

В первой строке содержится число N ($1 \leq N \leq 100000$) — размер первого массива. Во второй строке идет N целых чисел, не превосходящих по модулю 10^9 — элементы массива. Далее аналогично задаётся второй массив.

В решении запрещается использовать вспомогательные массивы, а также иные структуры данных с объёмом памяти, зависящем от размера исходных массивов.

Программа должна вывести слово YES, если массивы похожи, и слово NO в противном случае.

| Input | Output |
|--------------------------|--------|
| 3 5 7 5 3 7 5 7 | YES |

L. *Несамопересекающаяся незамкнутая ломаная*

Дано N различных точек на плоскости. Построить $(N - 1)$ - звенную несамопересекающуюся ломаную, проходящую через все эти точки.

В первой строчке программе подаётся на вход натуральное число N ($1 < N < 10^3$). Затем в N строках даётся по два целых числа, разделённых пробелом: абсцисса и ордината точки.

Требуется вывести N чисел от 0 до $N - 1$ (номера заданных точек) в таком порядке, чтобы соединённые в соответствующей последовательности точки образовывали несамопересекающуюся ломаную. Если возможных порядков построения ломаной несколько, можно вывести любой.

| Input | Output |
|-------|-------------|
| 6 | 3 4 2 5 0 1 |
| 1 2 | |
| 3 4 | |
| 4 1 | |
| 2 1 | |
| 3 2 | |
| 2 -1 | |

M. *Обувной магазин*

В обувном магазине продается обувь разного размера. Известно, что одну пару обуви можно надеть на другую, если она хотя бы на три размера больше. В магазин пришел покупатель. Требуется определить, какое наибольшее количество пар обуви сможет предложить ему продавец так, чтобы он смог надеть их все одновременно.

Сначала вводится размер ноги покупателя (обувь меньшего размера он надеть не сможет), затем количество пар обуви в магазине и размер каждой пары. Размер — натуральное число, не превосходящее 100, количество пар обуви в магазине не превосходит 1000.

Выведите единственное число — максимальное количество пар обуви.

| Input | Output |
|----------------|--------|
| 60 | 2 |
| 2 | |
| 60 63 | |
| 26 | 3 |
| 5 | |
| 30 35 40 41 42 | |

N. *Числа*

Саша и Катя учатся в начальной школе. Для изучения арифметики при этом используются карточки, на которых написаны цифры (на каждой карточке написана ровно одна цифра). Однажды они пришли на урок математики, и Саша, используя все свои карточки, показал число A , а Катя показала число B . Учитель тогда захотел дать им такую задачу, чтобы ответ на нее смогли показать и Саша, и Катя, каждый используя только свои карточки. При этом учитель хочет, чтобы искомое число было максимально возможным.

Во входном файле записано два целых неотрицательных числа A и B (каждое число в одной строке). Длина каждого из чисел не превосходит 100000 цифр.

Выведите одно число — максимальное целое число, которое можно составить используя как цифры первого числа, так и цифры второго числа. Если же ни одного такого числа составить нельзя, выведите -1 .

| Input | Output |
|------------------|--------|
| 280138 798081 | 8810 |
| 123 456 | -1 |
| 13579 4321 | 31 |

О. Анаграммы

Слово называется анаграммой другого слова, если оно может быть получено перестановкой его букв.

Даны два слова на отдельных строках. Слова состоят из строчных латинских букв и цифр. Длины слов не превышают 10^6 .

Требуется вывести YES — если введенные слова являются анаграммами друг друга, NO — если нет.

| Input | Output |
|----------------|--------|
| sharm marsh | YES |
| anas nnaass | NO |

Р. Клавиатура

Всем известно, что со временем клавиатура изнашивается, и клавиши на ней начинают залипать. Конечно, некоторое время такую клавиатуру еще можно использовать, но для нажатий клавиш приходится использовать большую силу.

При изготовлении клавиатуры изначально для каждой клавиши задается количество нажатий, которое она должна выдерживать. Если знать эти величины для используемой клавиатуры, то для определенной последовательности нажатых клавиш можно определить, какие клавиши в процессе их использования сломаются, а какие — нет.

Требуется написать программу, определяющую, какие клавиши сломаются в процессе заданного варианта эксплуатации клавиатуры.

Первая строка входного файла содержит целое число n ($1 \leq n \leq 100$) — количество клавиш на клавиатуре. Вторая строка содержит n целых чисел — c_1, c_2, \dots, c_n , где i ($1 \leq c_i \leq 100000$) — количество нажатий, выдерживаемых i -ой клавишей. Третья строка содержит целое число k ($1 \leq k \leq 100000$) — общее количество нажатий клавиш, и последняя строка содержит k целых чисел p_j ($1 \leq p_j \leq n$) — последовательность нажатых клавиш.

В выходной файл необходимо вывести n строк, содержащих информацию об исправности клавиш. Если i -ая клавиша сломалась, то i -ая строка должна содержать слово `yes` (без кавычек), если же клавиша работоспособна — слово `no`.

| Input | Output |
|-------------------------------|--------|
| 5 | yes |
| 1 50 3 4 3 | no |
| 16 | no |
| 1 2 3 4 5 1 3 3 4 5 5 5 5 4 5 | no |
| | yes |

Q. *Форум*

Клуб Юных Хакеров организовал на своем сайте форум. Форум имеет следующую структуру: каждое сообщение либо начинает новую тему, либо является ответом на какое-либо предыдущее сообщение и принадлежит той же теме.

После нескольких месяцев использования своего форума юных хакеров заинтересовал вопрос — какая тема на их форуме наиболее популярна. Помогите им выяснить это.

В первой строке вводится целое число N — количество сообщений в форуме ($1 \leq N \leq 1000$). Следующие строки содержат описание сообщений в хронологическом порядке.

Описание сообщения, которое представляет собой начало новой темы, состоит из трех строк. Первая строка содержит число 0. Вторая строка содержит название темы. Длина названия не превышает 30 символов. Третья строка содержит текст сообщения.

Описание сообщения, которое является ответом на другое сообщение, состоит из двух строк. Первая строка содержит целое число — номер сообщения, ответом на которое оно является. Сообщения нумеруются, начиная с единицы. Ответ всегда появляется позже, чем сообщение, ответом на которое он является. Вторая строка содержит текст сообщения.

При подсчёте ответов на сообщения надо учитывать ответы всех уровней, не только первого (ответы на ответы и т.п.).

Длина каждого из сообщений не превышает 100 символов.

Выведите название темы, к которой относится наибольшее количество сообщений. Если таких тем несколько, то выведите первую в хронологическом порядке

| Input | Output |
|-----------------------------------------------------------------------------|---------|
| 2 0 topic 1 body of message 1 0 topic 2 body of message 2 | topic 1 |

R. *Иннокентий решает задачи*

Мальчик Иннокентий решает вступительную работу в летний математический лагерь. В ней N заданий, которые можно выполнять в произвольном порядке. Разные задачи требуют разного времени для решения. При этом известно, что если задание с номером i выполнять j -м по счету, Иннокентию потребуется $T_i \times j$ времени: чем больше думаешь, тем больше устаешь. Например, если начать с первой задачи, а затем выполнить вторую, то потребуется $T_1 \times 1 + T_2 \times 2$ времени, а если выполнить сначала вторую задачу, а затем первую — то $T_2 \times 1 + T_1 \times 2$.

Подскажите Иннокентию, в каком порядке нужно решать задачи, чтобы на выполнение всей работы ушло как можно меньше времени.

В первой строке вводится число N , во второй строке — N чисел через пробел T_1, T_2, \dots, T_N , разделённые пробелами. Все числа целые и удовлетворяют следующим ограничениям: $0 < N \leq 10, 0 < T_i \leq 100$.

Требуется вывести сначала минимальное время, за которое можно решить все задачи, а затем — номера задач в том порядке, в котором их нужно решать, чтобы уложиться в это время. Все числа разделяются пробелами. Если решений несколько, нужно выдать любое из них.

| Input | Output |
|----------|----------|
| 2 2 3 | 7 2 1 |

S. *Такси*

После затянувшегося совещания директор фирмы решил заказать такси, чтобы развезти сотрудников по домам. Он заказал N машин — ровно столько, сколько у него сотрудников. Однако когда они подъехали, оказалось, что у каждого водителя такси свой тариф за 1 километр.

Директор знает, какому сотруднику сколько километров от работы до дома (к сожалению, все сотрудники живут в разных направлениях, поэтому нельзя отправить двух сотрудников на одной машине). Теперь директор хочет определить, какой из сотрудников на каком такси должен поехать домой, чтобы суммарные затраты на такси (а их несет фирма) были минимальны.

Первая строка входных данных содержит натуральное число N ($1 \leq N \leq 1000$) — количество сотрудников компании (совпадающее с количеством вызванных машин такси). Далее записано N чисел, задающих расстояния в километрах от работы до домов сотрудников компании (первое число — для первого сотрудника, второе — для второго и т.д.). Все расстояния — положительные целые числа, не превышающие 1000. Далее записано еще N чисел — тарифы за проезд одного километра в такси (первое число — в первой машине такси, второе — во второй и т.д.). Тарифы выражаются положительными целыми числами, не превышающими 10000.

В первой строке программа должна вывести минимальную стоимость доставки всех сотрудников. Во второй строке программа должна вывести N чисел. Первое число — номер такси, в которое должен сесть первый сотрудник, второе число — номер такси, в которое должен сесть второй и т.д., чтобы суммарные затраты на такси были минимальны. Если вариантов рассадки сотрудников, при которых затраты минимальны, несколько, выведите любой из них.

| Input | Output |
|---------------|-----------|
| 3 | 1700 |
| 10 20 30 | 1 3 2 |
| 50 20 30 | |
| 5 | 243 |
| 10 20 1 30 30 | 5 1 3 2 4 |
| 3 3 3 2 3 | |

T. *Большое число*

Маша написала на длинной полоске бумаги большое число и решил похвастаться своему старшему брату Серёже этим достижением. Но только она вышла из комнаты, чтобы позвать брата, как её сестра Дуня вбежала в комнату и разрезала полоску бумаги на несколько частей. В результате на каждой части оказалось одна или несколько идущих подряд цифр. Теперь Маша не может вспомнить, какое именно число она написала. Только помнит, что оно было очень большое. Чтобы утешить сестру, Серёжа решил выяснить, какое максимальное число могло быть написано на полоске бумаги перед разрезанием. Помогите ему.

Входные данные представляют собой одну или более строк, каждая из которых содержит последовательность цифр. Количество строк во входном файле не превышает 100, каждая строка содержит от 1 до 100 цифр. Гарантируется, что хотя бы в одной строке первая цифра отлична от нуля.

Выведите одну строку — максимальное число, которое могло быть написано на полоске перед разрезанием.

| Input | Output |
|-------|----------|
| 2 | 66220004 |
| 20 | |
| 004 | |
| 66 | |
| 3 | 3 |

U. *Решение задач*

Вася готовится к олимпиаде. Учитель дал ему N ($1 \leq N \leq 100000$) задач для тренировки. Для каждой из этих задач известно, каким умением a_i нужно обладать для её решения. Это означает, что если текущее умение Васи больше либо равно заданного умения для задачи, то он может ее решить.

Кроме того, после решения i -й задачи Васиного умение увеличивается на число b_i .

Исходное умение Васи равно A . Решать данные учителем задачи он может в произвольном порядке. Какое максимальное количество задач он сможет решить, если выберет самый лучший порядок их решения?

Сначала вводятся два целых числа N, A ($1 \leq N \leq 100000, 0 \leq A \leq 10^9$) — количество задач и исходное умение. Далее идут N пар целых чисел a_i, b_i ($1 \leq a_i \leq 10^9, 1 \leq b_i \leq 10^9$) — соответственно сколько умения нужно для решения i -й задачи и сколько умения прибавится после её решения. Выведите одно число — максимальное количество задач, которое Вася сможет решить.

| Input | Output |
|-------|--------|
| 3 2 | 3 |
| 3 1 | |
| 2 1 | |
| 1 1 | |

V. *Покрывтие отрезками*

Даны N отрезков на прямой и координаты M точек. Для каждой из точек отрезка $[1, L]$ определите — каким количеством данных отрезков они покрываются?

Во входном файле даны сначала L, N, M ($1 \leq L \leq 10000, 1 \leq N \leq 10000, 1 \leq M \leq 100000$).

Далее идут N пар чисел $l \leq r$ от 1 до L — левые и правые концы отрезков. Затем перечислены M чисел от 1 до L .

Выведите M чисел — количество отрезков, покрывающую каждую из указанных M точек.

| Input | Output |
|--------|--------|
| 39 4 7 | 4 |
| 3 21 | 3 |
| 3 15 | 0 |
| 2 20 | 4 |
| 3 17 | 4 |
| 4 | 0 |
| 17 | 0 |
| 33 | |
| 5 | |
| 9 | |
| 25 | |
| 37 | |

W. *Закраска прямой*

На числовой прямой окрасили N отрезков. Известны координаты левого и правого концов каждого отрезка (L_i и R_i). Найти длину окрашенной части числовой прямой.

В первой строке находится число N , в следующих N строках — пары L_i и R_i . L_i и R_i — целые, $-10^9 \leq L_i \leq R_i \leq 10^9, 1 \leq N \leq 15000$.

Вывести одно число — длину окрашенной части прямой.

| Input | Output |
|-------|--------|
| 1 | 0 |
| 10 10 | |
| 2 | 30 |
| 10 20 | |
| 20 40 | |

Х. Задача о минимальном покрытии

На прямой задано некоторое множество отрезков с целочисленными координатами концов $[L_i, R_i]$. Выберите среди данного множества подмножество отрезков, целиком покрывающее отрезок $[0, M]$, (M — натуральное число), содержащее наименьшее число отрезков.

В первой строке указано число M ($1 \leq M \leq 5000$). В каждой последующей строке записана пара чисел L_i и R_i ($|L_i|, |R_i| \leq 50000$), задающая координаты левого и правого концов отрезков. Список завершается парой нулей. Общее число отрезков не превышает 100000.

В первой строке выходного файла выведите минимальное число отрезков, необходимое для покрытия отрезка $[0, M]$. Далее выведите список покрывающего подмножества, упорядоченный по возрастанию координат левых концов отрезков. Список отрезков выводится в том же формате, что и во входе. Завершающие два нуля выводить не нужно. Если покрытие отрезка $[0, M]$ исходным множеством отрезков $[L_i, R_i]$ невозможно, то следует вывести единственную фразу **No solution.**

| Input | Output |
|----------------------------------|-------------|
| 1 -1 0 -5 -3 2 5 0 0 | No solution |
| 1 -1 0 0 1 0 0 | 1 0 1 |

У. Выравнивание забора

Забор на дачном участке Глеба состоит из деревянных планок, каждая высотой h_i . Для того чтобы использовать одну из сторон забора для прыжков в высоту, его необходимо выровнять — сделать все планки одной и той же высоты.

Приглашённые рабочие объявили, что изменение высоты i -ой планки на один метр в любую сторону (уменьшения или увеличения) будет стоить a_i рублей. При этом изменять высоту на нецелую величину рабочие не умеют.

Помогите Глебу выровнять забор, заменив все высоты на какую-то одну, так, чтобы общая плата за работы была минимальна.

Требуется написать программу, которая будет определять оптимальную высоту нового забора и сумму денег, в которую обойдется ее выравнивание.

В первой строке входного файла задано одно число n ($1 \leq n \leq 10^5$) — количество планок в заборе. Во второй строке задано n целых чисел h_i ($1 \leq h_i \leq 10^6$), где h_i — высота i -ой планки. В третьей строке так же заданы n целых чисел a_i ($1 \leq a_i \leq 10^6$) — цена изменения высоты i -ой планки на один метр.

Выведите в выходной файл два целых числа: конечную высоту забора и сумму денег, которую придется потратить на его выравнивание. Если вариантов выравнивания одинаковой стоимости несколько, то выведите минимально возможную высоту.

| Input | Output |
|----------------------------------|--------|
| 6 6 7 8 8 7 7 10 6 3 1 1 4 | 7 14 |
| 5 7 5 7 9 8 10 8 7 8 5 | 7 37 |
| 5 8 5 10 9 7 2 5 4 8 4 | 9 34 |

Z. *Белоснежка и N гномов*

У Белоснежки N гномов, и все они очень разные. Она знает, что для того, чтобы уложить спать i -го гнома нужно a_i минут, и после этого он будет спать ровно b_i минут. Помогите Белоснежке узнать, может ли она получить хотя бы минутку отдыха, когда все гномы будут спать, и если да, то в каком порядке для этого нужно укладывать гномов спать.

Например, пусть есть всего два гнома, $a_1 = 1, b_1 = 10, a_2 = 10, b_2 = 20$. Если Белоснежка сначала начнет укладывать первого гнома, то потом ей потребуется целых 10 минут, чтобы уложить второго, а за это время проснется первый. Если же она начнет со второго гнома, то затем она успеет уложить первого и получит целых 10 минут отдыха.

Первая строка входного файла содержит число N ($1 \leq N \leq 10^5$), вторая строка содержит числа a_1, a_2, \dots, a_n , третья — числа b_1, b_2, \dots, b_n ($1 \leq a_i, b_i \leq 10^9$).

Выведите в выходной файл N чисел — порядок, в котором нужно укладывать гномов спать. Если Белоснежке отдохнуть не удастся, выведите число -1 .

| Input | Output |
|---------------------|--------|
| 2 1 10 10 20 | 2 1 |
| 2 10 10 10 10 | -1 |

ZA. *Гражданская оборона*

Штаб гражданской обороны Тридесятой области решил обновить план спасения на случай ядерной атаки. Известно, что все N селений Тридесятой области находятся вдоль одной прямой дороги. Вдоль дороги также расположены M бомбоубежищ, в которых жители селений могут укрыться на случай ядерной атаки.

Чтобы спасение в случае ядерной тревоги проходило как можно эффективнее, необходимо для каждого селения определить ближайшее к нему бомбоубежище.

В первой строке вводится число n — количество селений ($1 \leq n \leq 100000$). Вторая строка содержит N различных целых чисел, i -е из этих чисел задает расстояние от начала дороги до i -го селения. В третьей строке входных данных задается число M — количество бомбоубежищ ($1 \leq M \leq 100000$). Четвертая строка содержит M различных целых чисел, i -е из этих чисел задает расстояние от начала дороги до i -го бомбоубежища. Все расстояния положительны и не превышают 10^9 . Селение и убежище могут располагаться в одной точке.

Выведите N чисел — для каждого селения выведите номер ближайшего к нему бомбоубежища. Бомбоубежища пронумерованы от 1 до M в том порядке, в котором они заданы во входных данных. Если два бомбоубежища равноудалены, можно вывести номер любого из них.

Входные данные

| Input | Output |
|---------------------------|---------|
| 4 1 2 6 10 2 7 3 | 2 2 1 1 |

ZB. Триангуляция

Вася нарисовал выпуклый N -угольник и провел в нем несколько диагоналей таким образом, что никакие две диагонали не пересекаются внутри N -угольника. Теперь он утверждает, что весь N -угольник оказался разбит на треугольники. Напишите программу, которая проверяет истинность Васиного утверждения.

Входные данные

В первой строке вводятся два числа N и M . N — количество вершин N -угольника ($3 \leq N \leq 1000$) и M — количество диагоналей, проведенных Васей. Далее на вход программы поступают M пар чисел, задающих диагонали (каждая диагональ задается парой номеров вершин, которые она соединяет). Гарантируется, что каждая пара чисел задает диагональ (то есть две вершины различны и не являются соседними), а также что никакие две пары не задают одну и ту же диагональ. Никакие две диагонали не пересекаются внутри N -угольника. Вершины N -угольника нумеруются числами от 1 до N .

Выходные данные

Если Васиное утверждение верно, то программа должна выводить единственное число 0. В противном случае необходимо вывести сначала число K — количество вершин в какой-нибудь не треугольной части. Далее должно быть выведено K чисел — номера вершин исходного N -угольника, которые являются вершинами этой K -угольной части в порядке обхода этой части.

| Input | Output |
|-------------------|-----------|
| 3 0 | 0 |
| 4 1 1 3 | 0 |
| 6 2 1 3 5 3 | 4 1 3 5 6 |