Задача A. Фильтр

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

I'll be back.

T-800

После стычки с Кайлом Ризом у Терминатора сломался клапан, отвечающий за движение суставов. Теперь ему необходимо ежедневно наливать себе в фильтр x литров масла.

Но, поскольку некоторые участки процессора Терминатора повреждены, он не может точно отмерить дозу и каждый i-й день вливает a_i литров масла в фильтр размером r литров. Если масла в фильтре становится больше, чем фильтр способен в себя вместить, то лишнее масло выливается через край. После этого масло вытекает из фильтра в другие системы. Если в данный момент масла в фильтре не более x литров, то оно полностью вытекает, и фильтр становится пуст. В противном случае количество масла в фильтре уменьшается на x литров, утекших в системы Терминатора. Все оставшееся масло остается в фильтре до следующего дня и далее.

Помогите Терминатору определить суммарное количество масла, вытекшего в его системы, после n дней. В первый день до залива масла в фильтр он пуст.

Формат входного файла

В первой строке входного файла заданы три числа n, r и x ($1 \le n \le 10^3, 1 \le r, x \le 10^6$), где n — количество дней, в течении которых Терминатор вливает масло в фильтр. В следующей строке находятся n чисел a_i ($0 \le a_i \le 10^6$) — количество литров масла, которые он вливает в i-й день.

Формат выходного файла

В единственной строке выведите суммарное количество литров масла, которое вытекло из фильтра в другие системы Терминатора за n дней.

Примеры

filter.in	filter.out
2 6 3 4 1	5
2 6 3 7 2	6

Комментарий

Решения, верно работающие в случаях, когда фильтр не переполняется, будут оцениваться в 50 баллов.

Обратите внимание на возможность узнать результат проверки вашего решения, нажав на ссылку «`Request feedback`» на вкладке «`Runs`».

Задача B. Тестирование

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

— Доктор Сильберман: Ты мне руку сломала!
— Сара Коннор: В человеческом организме 215 костей. Это была только одна.

«Терминатор 2: Судный день»

Чтобы добиться возможности видеть своего сына, Саре необходимо доказать доктору Сильберману, главному врачу клиники, свою адекватность и дееспособность. В качестве одного из тестов доктор предложил Саре следующую задачу.

Доктор выкладывает на стол n карточек, на каждой из которых написано по два натуральных числа, каждое из которых находится в интервале от одного до $n+1$. В этом наборе ни на одной карточке не написаны одинаковые числа. Кроме этого, не существует двух карточек с одинаковой парой чисел.

Сара может выбрать несколько карточек и изменить написанные на них числа. После изменений, внесенных Сарой, должно существовать число, написанное на всех карточках. Кроме этого должны быть выполнены условия, описанные выше — уникальность карточек и несовпадение чисел на любой из них.

Задача Сары состоит в том, чтобы сделать это, изменив как можно меньше карточек. Помогите ей сделать это.

Формат входного файла

В первой строке дано одно целое число n ($1 \le n \le 100\,000$) — количество карточек, выложенных на стол доктором. Следующие n строк содержат по два различных числа от одного до $n+1$ — числа, написанные на карточках.

Гарантируется, что не существует двух карточек с одинаковой парой чисел, или двух карточек, числа на которых отличаются только порядком их записи.

Формат выходного файла

В единственной строке выведите число x — минимальное количество карточек, которые требуется изменить Саре Коннор для прохождения теста.

Примеры

<table>
<tr><th>psycho.in</th><th>psycho.out</th></tr>
<tr><td><pre>4
1 2
2 3
3 4
4 5</pre></td><td><pre>2</pre></td></tr>
</table>

Комментарий

Решения, работающие для $n \le 100$, будут оцениваться в 60 баллов.

Обратите внимание на возможность узнать результат проверки вашего решения, нажав на ссылку «`Request feedback`» на вкладке «`Runs`».

Задача C. T-X против T-850

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

I'm back!

T-850

Терминатор T-850 вернулся, чтобы снова противостоять компьютерной сети Skynet. Теперь Skynet выставила против него свое новейшее изобретение — терминатора T-X, призванного уничтожить T-850 и начать Судный День.

Очередная схватка T-850 и T-X проходит на открытой местности, где из строений присутствует только одно здание. Будем считать, что здание имеет форму окружности радиуса r с центром в координатах (x_c, y_c). T-850 находится в точке (x_1, y_1), а T-X — в точке (x_2, y_2). T-850 хочет выстрелить в T-X, но может это сделать только, если видит её.

Считается, что T-850 видит T-X только если отрезок, их соединяющий, не пересекается со стеной здания. Помогите программистам из будущего, которые будут писать программное обеспечение T-850, предусмотреть даже такой вариант схватки и определить, может ли T-850 в данный момент выстрелить в T-X.

Формат входного файла

В первой строке задано число t ($1 \le t \le 5$) — количество запросов. Каждый запрос состоит из трёх строк в следующем формате.

В первой строке заданы числа x_c, y_c, r — координаты и радиус здания. Во второй строке заданы числа x_1, y_1 — координаты T-850. В третьей строке заданы числа x_2, y_2 — координаты T-X.

Все числа во входном файле целые и не превышают по модулю 100. Гарантируется, что расстояние от прямой, проходящей через (x_1, y_1) и (x_2, y_2), до точки (x_c, y_c) отстоит от r более, чем на 10^{-5}. Гарантируется, что ни T-X, ни T-850 не стоят на границе здания.

Формат выходного файла

Для каждого запроса выведите «`YES`» или «`NO`» — может ли T-850 попасть в T-X.

Примеры

tx.in	tx.out
3 0 0 1 0 0 2 0 0 0 1 -2 0 2 0 0 0 1 2 0 2 1	NO NO YES

Комментарий

Решения, в которых хотя бы кто-нибудь из T-850 или T-X находится внутри здания, будут оцениваться из 50 баллов.

Обратите внимание на возможность узнать результат проверки вашего решения, нажав на ссылку «`Request feedback`» на вкладке «`Runs`».

Задача D. Начало конца

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

> Это начало конца.
>
> Теглайн фильма «Терминатор: Да придет спаситель»

Джон Коннор и Маркус Райт добрались до главной сборочной базы терминаторов Т-800. Теперь они готовы взорвать эту базу и положить конец одной из битв за будущее человечества. Однако, они заметили странный спецэффект — топливные элементы терминаторов, которые они хотят взорвать, были переставлены за время их битвы с одним из активированных Т-800. Джон и Маркус решили, во избежание непредвиденных последствий, переставить их назад перед взрывом.

На каждом из n топливных элементов написано число. До перестановки все они были выставлены в ряд. Джон и Маркус знают, что перестановка выполнялась по следующему алгоритму:

- разворачивается отрезок из k топливных элементов, начинающийся в первом слева элементе
- разворачивается отрезок из k топливных элементов, начинающийся во втором слева элементе
- ...
- разворачивается отрезок из k топливных элементов, заканчивающийся в самом правом элементе

Таким образом, если исходно в ряд были выставлены элементы с написанными числами `[1, 2, 3, 1, 2]`, а число k было равно трем, то после перестановки порядок чисел на элементах будет таким: `[3, 1, 2, 2, 1]`.

Единственное, чего Джон и Маркус не знают — число k. Они помнят порядок чисел, записанных на элементах до их перестановки, и видят порядок чисел на элементах после нее. Помогите им восстановить возможные значения числа k.

Формат входного файла

В первой строке входного файла задано число n ($1 \le n \le 100\,000$) — количество топливных элементов.

Далее идут две строки по n чисел a_i ($1 \le a_i \le 100\,000$) в каждой — порядок чисел на элементах до и после перестановки соответственно.

Гарантируется, что количество вхождений любого числа в данные последовательности совпадает.

Формат выходного файла

В единственной строке выведите число x — количество возможных чисел k. В следующей строке выведите x различных чисел, разделенных пробелами — сами варианты чисел.

Варианты выводить в порядке возрастания.

Примеры

beginend.in	beginend.out
5 1 2 3 1 2 3 1 2 2 1	1 3

Комментарий

Решения, работающие для $n \leq 100$, будут оцениваться в 40 баллов.

Решения, работающие для $n \leq 5\,000$, будут оцениваться в 60 баллов.

Обратите внимание на возможность узнать результат проверки вашего решения, нажав на ссылку «`Request feedback`» на вкладке «`Runs`».