Задача A. Простые суффиксы

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

Будем называть *суффиксом* числа x число y, которое получается из десятичной записи x откидыванием любого числа первых цифр. Если в десятичной записи x не встречается нулей, то все суффиксы x не содержат ведущих нулей. Например, суффиксами числа 283 являются числа 283, 83 и 3.

Число называется простым, если оно имеет ровно два натуральных делителя. Заметим, что число 1 простым не является — у него только один натуральный делитель.

Сене нравятся простые числа, не содержащие нулей в десятичной записи, все суффиксы которых также являются простыми числами.

Заданы целые числа a и b. Помогите Сене подсчитать, сколько целых чисел между a и b включительно ему нравится.

Формат входных данных

Входные данные содержат два целых числа a и b ($1 \leqslant a \leqslant b \leqslant 10^{11}$).

Формат выходных данных

Выведите количество простых чисел, не содержащих нулей, от a до b включительно, таких, что если откинуть сколько угодно первых цифр числа, то оставшееся число всё ещё будет простым.

Примеры

стандартный ввод	стандартный вывод
4 13	3
101 109	0
281 286	1

Замечание

В первом примере подходят числа 5, 7 и 13.

Во втором примере ни одно число не подходит, так как все числа в диапазоне содержат 0.

В третьем примере число 283 подходит, так как числа 283, 83 и 3 — простые.

Задача B. Простая последовательность цифр

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

На перемене перед уроком математики Рома решил поупражняться в определении простоты числа. Напомним, что простым называется натуральное число, имеющее ровно два различных натуральных делителя — единицу и самого себя. Сначала он написал на доске первое простое число, после чего справа приписал к нему второе, затем третье и так далее. Всего Рома выписал на доску первые n простых чисел. В результате действий Ромы на доске появилось одно длинное число, которое начинается так: «$23571113171923\ldots$».

Когда в кабинет вошла Елена Евгеньевна, учительница Ромы, она предложила классу решить следующую задачку: вычеркнуть из написанного на доске числа k цифр так, чтобы оставшееся на доске число было максимальным.

Помогите Роме и одноклассникам решить предложенную задачу, чтобы не получить двойку от строгой учительницы.

Формат входных данных

Входной файл к этой задаче содержит несколько наборов тестовых данных. В первой строке входного файла задано число T — количество наборов в файле.

В следующих T строках идут описания наборов, каждое из которых состоит из двух целых положительных чисел n и k. Гарантируется, что первые n простых чисел содержат в себе хотя бы $k+1$ цифру суммарно.

Сумма всех n во входном файле не превосходит 400 000.

Формат выходных данных

Для каждого из тестовых наборов в отдельной строке выведите искомое максимальное число для соответствующих n и k.

Пример

стандартный ввод	стандартный вывод
2 4 2 5 3	57 711

Замечание

Пояснение к примеру В первом тесте Рома выписал число 2357. Максимальное число, которое может получиться после вычеркивания из него двух цифр: 57.

Во втором тесте Рома выписал число 235711. Максимальное число, которое может получиться после вычеркивания из него трех цифр: 711.

Задача C. Диофантово уравнение

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.25 секунд
Ограничение по памяти:	256 мегабайт

Даны натуральные числа a, b и c. Решите в целых числах уравнение $ax+by=c$. Среди множества решений следует выбрать такое, где x имеет наименьшее неотрицательное значение.

Формат входных данных

Первая строка содержит три целых числа a и b и c ($1 \leqslant a, b, c \leqslant 10^9$).

Формат выходных данных

Выведите искомые x и y через пробел. Если решения не существует, выведите одну строку «`Impossible`».

Примеры

стандартный ввод	стандартный вывод
1 2 3	1 1
10 6 8	2 -2

Задача D. Кто не будет решать математику — пойдёт красить забор

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.25 секунд
Ограничение по памяти:	256 мегабайт

Миша не любит математику. Из-за этого он не смог решить сложную задачу на Всероссе, не стал призёром и не получил 150000 руб. от Москвы. Чтобы хоть как-то сводить концы с концами Мише приходится подрабатывать, а именно — красить заборы.

Мише очень нравятся зебры, поэтому он пытается найти их везде где только можно. Миша должен покрасить забор на даче и ему выдали неограниченное количество белой и чёрной краски. Забор является последовательностью досок, некоторые из которых уже покрашены в белый или чёрный цвет, а остальные ещё нет. Менять цвета уже покрашенных досок запрещается, а для остальных Миша может выбрать цвета по своему усмотрению. В данной задаче забор представляется строкой, состоящей из символов «`0`», «`1`» и «`?`», означающих белую доску, чёрную доску и ещё не окрашенную доску соответственно.

Миша считает, что забор похож на зебру, если существуют целые числа a и b ($a > 0, b \geqslant 0$), такие что первые a досок забора являются белыми, следующие b досок являются чёрными, затем снова идут a белых досок, далее опять b чёрных и так далее, при этом последний блок может быть не полным. Например, заборы, описываемые строками «`01101`» ($a = 1, b = 2$), «`000`» ($b = 0$, a может быть любым целым положительным числом) и «`00110011`» ($a = 2, b = 2$) являются зебрами, а «`01001`» и «`101010`» — нет.

Помогите Мише раскрасить оставшиеся доски таким образом, чтобы забор являлся зеброй для каких-нибудь чисел a и b ($a > 0, b \geqslant 0$). Поскольку Миша мечтает покрасить в чёрный цвет всё что он видит, то если подходящих раскрасок забора несколько, выберите среди них ту, в которой как можно больше чёрных досок. Среди таких раскрасок разрешается выбрать любую.

Формат входных данных

Входные данные содержат единственную строку s ($1 \leqslant |s| \leqslant 300000$), состоящую из символов «`0`», «`1`» и «`?`».

Формат выходных данных

Если невозможно раскрасить ещё не покрашенные доски забора таким образом, чтобы он был похож на зебру, то выведите -1 в единственной строке выходных данных. В противном случае выведите какое-нибудь решение с максимальным возможным количеством чёрных досок. Решение выводите как строку из символов «`0`» и «`1`», означающих белую и чёрную доску соответственно.

Примеры

стандартный ввод	стандартный вывод
0?	01
0110?	01101
10?	-1
011011	011011
101	-1

Задача E. Сколько простых?

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

Найдите количество простых чисел от n^2 до $n^2 + n$ включительно.

Формат входных данных

Первая строка содержит число n ($1 \leqslant n \leqslant 10^7$).

Формат выходных данных

Выведите количество простых чисел от n^2 до $n^2 + n$ включительно.

Примеры

стандартный ввод	стандартный вывод
2	1
5	1

Задача F. Все обратные по модулю

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	3.5 секунд
Ограничение по памяти:	512 мегабайт

Дано простое число p. Найдите обратные по модулю p ко всем числам от 1 до $p-1$.

Формат входных данных

Первая строка содержит число p ($1 \leqslant p \leqslant 10^8$).

Формат выходных данных

Для каждого числа от 1 до $p-1$ требуется посчитать обратное по модулю $p-1$. Так как чисел очень много, сначала выведите сумму обратных для первых 100 чисел по модулю p, потом для вторых 100 чисел по модулю p, потом для третьих 100 чисел и так далее. Если $p-1$ не делится на 100, последнее из выведенных вами чисел будет состоять из суммы меньше, чем 100 слагаемых.

Примеры

стандартный ввод	стандартный вывод
2	1
5	0

Замечание

Обратите внимание, что сумма 100 чисел тоже берется по модулю, так что все числа, которые вы выводите не должны превышать $p-1$.

Задача G. Странная функция

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	0.5 секунд
Ограничение по памяти:	256 мегабайт

Дан массив из n целых чисел $a_1, a_2, \ldots, a_n$, определим

$$f(l, r) = \gcd(a_l, a_{l+1}, ..., a_r) \cdot \left(\left(\sum_{i=l}^{r} a_i \right) - \max(a_l, a_{l+1}, \ldots, a_r) \right).$$

Формат входных данных

Первая строка содержит целое число n ($1 \leqslant n \leqslant 50000$).

Вторая строка содержит n целых чисел $a_1, a_2, \ldots a_n$ ($-10^6 \leqslant a_i \leqslant 10^6$).

Формат выходных данных

Выведите $\max\limits_{1 \leqslant l \leqslant r \leqslant n} f(l, r)$.

Примеры

стандартный ввод	стандартный вывод
4 10 4 5 6	15
5 7 12 24 6 5	144

Задача H. Функция Эйлера

Имя входного файла:	`стандартный ввод`
Имя выходного файла:	`стандартный вывод`
Ограничение по времени:	3 секунды
Ограничение по памяти:	1024 мегабайта

Красить забор — не очень. Вернёмся к математике.

Формат входных данных

Дано число n ($1 \leqslant n \leqslant 10^8$).

Формат выходных данных

Для каждого числа от 1 до n требуется посчитать функцию Эйлера от него. Так как чисел очень много, сначала выведите сумму функций Эйлера для первых 100 чисел, потом для вторых 100 чисел, потом для третьих 100 чисел и так далее. Если n не делится на 100, последнее из выведенных вами чисел будет состоять из суммы меньше, чем 100 слагаемых.

Примеры

стандартный ввод	стандартный вывод
10	32
200	3044 9188

Замечание

Для чисел от 1 до 10 функция Эйлера будет равна соответственно $1, 1, 2, 2, 4, 2, 6, 4, 6, 4$, что в сумме даёт 32.

Задача I. Грустные танцы

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

Во Флатляндии проводится ежегодный турнир по танцам!

Из города NN приехала команда, состоящая из n танцоров, и вот настал день соревнований.

Состязания проходят в таком формате: танцоры пронумерованы от 1 до n, и изначально i-й танцор стоит на i-м месте. После этого они начинают танцевать по заранее согласованной программе выступления a : каждую минуту танцор с a_i-го места передвигается на i-е место, при этом все a_i различны. От команды требуется выстроиться так, чтобы i-й танцор оказался на b_i-м месте (аналогично, все b_i различны). После этого выступление завершается, и жюри оценивает его техничность и артистизм. При этом выступление должно продлиться хотя бы одну минуту, иначе оценивать будет просто нечего.

Но в этом году участники заподозрили жюри в подлоге: к ним пришла мысль, что, возможно, следуя программе a , они никогда не смогут занять требуемое положение b, что приводит к автоматическому поражению в турнире.

Так как они не программисты по образованию, команда города NN решила обратиться к вам за помощью: проверьте по их программе выступления a и требуемому положению b, существует ли такое положительное количество минут k, что через k минут после начала выступления i-й танцор будет находиться на b_i-м месте.

Формат входных данных

В первой строке входного файла содержится целое число $n (1 \leqslant n \leqslant 10^6)$ — количество участников команды, приехавшей из города NN.

Во второй строке вводится a — перестановка чисел от 1 до n.

Во второй строке вводится b — перестановка чисел от 1 до n.

Формат выходных данных

Для каждого тестового примера выведите «Yes» (без кавычек), если существует такое количество минут k, что спустя k минут после начала выступления все танцоры будут в требуемом от них положении, или «No» (без кавычек), если такого k не существует.

Примеры

стандартный ввод	стандартный вывод
4 2 3 4 1 1 2 3 4	Yes
4 1 2 3 4 2 1 4 3	No

Задача J. Шоппинг

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

Вчера был день рожденья Дени, и она получила много подарков от своих друзей. Можно сказать, что благодаря этим подаркам у нее теперь есть неограниченное количество всех типов товаров, которые продаются в магазинах торгового центра. Дени решила продать какие-то из них, чтобы получить некоторое количество денег. Конечно, с этими деньгами она отправится за покупками в торговый центр со своими друзьями, но она будет покупать только те типы товаров, которые отличаются от тех, которые она продала. После всего этого Дени хочет остаться с определенным количество денег (если это возможно сделать только путем продажи некоторых из ее подарков, то она не будет делать покупки). Так как имеется множество типов товаров с различными ценами, то ей предстоит трудный выбор — какие товары продавать, а какие покупать, чтобы в конце концов получить то количество денег, которые она хочет.

Пусть в магазинах имеется k типов товаров с ценами $a_1, a_2, \ldots, a_k$ левов (валюта Болгарии) соответственно, и Дени хочет в итоге иметь ровно n левов. Вы должны вывести сколько раз она должна покупать или продавать каждый тип товаров (покупка обозначается отрицательным числом, а продажа — положительным), чтобы в итоге у Дени было n левов. Ваша программа должна обработать t тестов в одном наборе входных данных. Так как выводимые числа могут быть очень большими, каждое число должно быть выведено как произведение не более чем 100 целых чисел. Если задача имеет более одного решения, вы можете вывести любое из них. Если решения нет, выведите текст «No solutions» (без кавычек).

Формат входных данных

В первой строке находится одно положительное целое число t — количество тестов, которые Ваша программа должна обработать ($1 \leqslant t \leqslant 2$). Далее идет описание тестов.

Для каждого теста в первой строке находится число k, на следующей строке — k натуральных чисел $a_1, a_2, \ldots, a_k$ — цены для каждого типа товара в магазинах супермаркета ($4 \leqslant k \leqslant 10^5$, $1 \leqslant a_i \leqslant 10^9$). В последней строке описания теста находится натуральное число n — количество левов, которое Дени хочет иметь в итоге ($1 \leqslant n \leqslant 10^9$).

Формат выходных данных

Для каждого теста необходимо вывести текст «No solutions» (без кавычек), если задача не имеет решения. Иначе нужно вывести k чисел (каждое в формате $num_1 * num_2 * \ldots * num_p$, $1 \leqslant p \leqslant 100$, $-10^9 \leqslant num_1 \leqslant 10^9$, $0 \leqslant num_i \leqslant 10^9$ для $2 \leqslant i \leqslant p$), которые определяют, сколько раз товар каждого типа продается или покупается Дени. Если число отрицательное, это означает, что она покупает этот товар, если положительное — она его продает. Если это ноль, то это означает, что этот тип товара она не покупала и не продавала. Если необходимо вывести, например, 1000000002, то оно может быть выведено в виде $2 * 500000001$, но не 1000000002, потому что оно больше 10^9. .

Примеры

стандартный ввод	стандартный вывод
1 2 3 5 11	2 1
1 4 30 42 70 105 413	7 3*3 5 -1*5

Задача K. Увеличить НОД

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

У Mr. F есть n положительных целых чисел, $a_1, a_2, \ldots, a_n$.

Он считает, что наибольший общий делитель этих чисел слишком маленький, и хочет увеличить его, удалив некоторые из чисел.

Но эта задача показалась ему слишком простой, поэтому он не хочет решать ее сам. Если вы ему поможете, он даст вам несколько баллов в качестве вознаграждения.

Ваша задача найти минимальное количество чисел, удалив которые, наибольший делитель оставшихся будет строго больше чем наибольший общий делитель всех исходных чисел.

Формат входных данных

В первой строке входного файла записано единственное целое число n ($2 \leqslant n \leqslant 3 \cdot 10^5$) — количество чисел у Mr. F.

Во второй строке записаны n целых чисел, $a_1, a_2, \ldots, a_n$ ($1 \leqslant a_i \leqslant 1.5 \cdot 10^7$).

Формат выходных данных

Выведите одно целое положительное число — минимальное количество чисел, удалив которые, наибольший делитель оставшихся будет строго больше чем наибольший общий делитель всех исходных чисел.

Вы не можете удалить все числа.

Если решения не существует, выведите «`-1`» (без кавычек).

Примеры

стандартный ввод	стандартный вывод
3 1 2 4	1
4 6 9 15 30	2
3 1 1 1	-1

Замечание

В первом примере, НОД изначально 1. Вы можете удалить 1, НОД увеличится, и станет равным 2. Таким образом, ответ 1.

Во втором примере, НОД изначально 3. Вы можете удалить два числа, 6 и 9, НОД увеличится, и станет равным 15. Можно показать, что удалив одно число, увеличить НОД невозможно. Таким образом, ответ 2.

В третьем примере, невозможно удалить числа, чтобы НОД увеличился. Таким образом, ответ -1.

Задача L. Система линейных уравнений

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

Дана система из двух линейных сравнений: $x \equiv a\,(mod\,n)$ и $x \equiv b\,(mod\,m)$, где числа n и m не обязательно взаимно простые. Решите эту систему или определите, что она не имеет решений.

Формат входных данных

В первой строке входного файла записано единсвенное число $1 \leqslant t \leqslant 100\,000$. В следующих t строках содержатся по четыре целых числа a, b, n, m, задающих одну систему сравнений. Все числа не превосходят по модулю 10^4, $n > 1, m > 1$.

Формат выходных данных

Программа должна вывести t строк, по одной на каждую систему.

В случае, если система не имеет решений, выведите строку «NO» (без кавычек).

В случае, если решение есть, то необходимо вывести слово «YES» (без кавычек) и два таких целых числа x_0 и p, $0 \leqslant x < p$, такие, что множество чисел $x = x_0 + kp$ (где k — произвольное целое число) является решением данной системы.

Пример

стандартный ввод	стандартный вывод
3 3 2 5 9 1 1 5 9 7 13 20 24	YES 38 45 YES 1 45 NO