

Задача А. Производство кексов

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

Компания БерКекс производит кексы. В компании есть ровно одно устройство для заливки формочек для кексов, а также два конвейера с формочками.

Известно, что на первом конвейере есть всего n формочек, и они будут появляться около устройства, заливающего их, в моменты времени $a_1, a_2, \dots, a_n$. На втором же конвейере есть m формочек, и они будут появляться около устройства в моменты времени $b_1, b_2, \dots, b_m$. Все моменты времени целочисленные. Гарантируется, что последовательность $a_1, a_2, \dots, a_n$ не содержит повторяющихся элементов и задана в строго возрастающем порядке ($a_1 < a_2 < \dots < a_n$). Также гарантируется, что последовательность $b_1, b_2, \dots, b_m$ не содержит повторяющихся элементов и задана в строго возрастающем порядке ($b_1 < b_2 < \dots < b_m$).

Когда рядом с устройством в один момент времени оказываются сразу две формочки, устройство само выбирает ровно **одну** формочку и заливает её, другая формочка останется незалитой. Залить обе формочки в один и тот же момент нельзя. Если же в какой-то момент времени рядом с устройством оказывается только одна формочка, устройство обязательно заливает ее. Считайте, что формочки заливаются мгновенно.

Определите количество кексов, которые будут приготовлены в компании БерКекс. Кекс считается приготовленными, если устройство зальет формочку для него.

Формат входных данных

В первой строке следует целое число n ($1 \leq n \leq 100$) — количество формочек на первом конвейере.

В каждой из следующих n строк следуют моменты времени появления формочек с первого конвейера у устройства, заливающего их. В i -й строке следует одно целое число a_i ($1 \leq a_i \leq 100$) — момент появления i -й формочки с первого конвейера у устройства, заливающего их. Гарантируется, что значения a_i различны и заданы в строго возрастающем порядке (то есть $a_1 < a_2 < \dots < a_n$).

В следующей строке следует целое число m ($1 \leq m \leq 100$) — количество формочек на втором конвейере.

В каждой из следующих m строк следуют моменты времени появления формочек со второго конвейера у устройства, заливающего их. В j -й строке следует одно целое число b_j ($1 \leq b_j \leq 100$) — момент появления j -й формочки со второго конвейера у устройства, заливающего их. Гарантируется, что значения b_j различны и заданы в строго возрастающем порядке (то есть $b_1 < b_2 < \dots < b_m$).

Формат выходных данных

Выведите количество кексов, которые будут приготовлены в компании БерКекс. Кекс считается приготовленными, если устройство зальет формочку для него.

Примеры

стандартный ввод	стандартный вывод
5 1 3 7 10 11 4 2 3 10 11	6
2 1 50 2 50 100	3

Замечание

В первом примере у устройства окажутся 5 формочек с первого конвейера в моменты времени [1, 3, 7, 10, 11] и 4 формочки со второго конвейера в моменты времени [2, 3, 10, 11]. Таким образом, устройство зальет 6 формочек, например, первую, вторую и третью формочки с первого конвейера, и первую, третью и четвертую формочки со второго конвейера.

Во втором примере конвейер зальет первую формочку с первого конвейера, вторую формочку со второго конвейера, а также либо вторую формочку с первого конвейера, либо первую формочку со второго конвейера. Таким образом, суммарно будет залито 3 формочки.

Задача В. Контрольная

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

Учитель проверил контрольную работу, которую писали n учеников, причём i -й ученик получил a_i баллов.

Учитель хочет преобразовать баллы в оценки таким образом, чтобы ученик, набравший наибольшее количество баллов, получил оценку 100, при этом баллы всех остальных учеников учитель будет преобразовывать пропорционально, всегда округляя вниз. Если какой-то из учеников наберет 0 баллов, то его оценка должна быть равна 0.

Определите, какую оценку получит каждый из учеников.

Формат входных данных

В первой строке следует целое число n ($1 \leq n \leq 50$) — количество учеников, писавших контрольную работу.

Во второй строке следует последовательность целых чисел $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 10\,000$), где a_i равно количеству баллов, которые i -й ученик получил за контрольную. Гарантируется, что хотя бы один ученик набрал **положительное** количество баллов.

Формат выходных данных

Выведите n целых чисел — оценки, которые получили ученики за контрольную. Оценки учеников должны следовать в том же порядке, в котором заданы их баллы во входных данных.

Примеры

стандартный ввод	стандартный вывод
6 15 27 8 33 19 50	30 54 16 66 38 100
4 0 0 0 3	0 0 0 100
6 67 89 72 100 95 88	67 89 72 100 95 88
7 1234 3483 234 5738 3421 5832 4433	21 59 4 98 58 100 76
1 8765	100

Замечание

В первом примере максимальное количество баллов равно 50, и этот ученик получил оценку 100, поэтому каждый ученик получил оценку, которая в два раза больше, чем количество набранных ими баллов.

Во втором примере все, кроме четвёртого ученика, получили оценку 0, так как набрали 0 баллов.

В третьем примере максимальное количество баллов равно 100, поэтому все ученики получили оценку, равную набранным ими баллам.

Задача С. Палиндромные слагаемые

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

Контрольная работа из задачи В содержала в себе задания только лишь одного типа.

Задано **двузначное** число x в системе счисления по основанию k . Число x в системе счисления по основанию k — это такая последовательность чисел $d_0, d_1, \dots$ ($0 \leq d_i < k$), что $d_0 + d_1 \cdot k + d_2 \cdot k^2 + \dots = x$. Легко показать, что для любого x существует ровно одна такая последовательность.

Число называется *k -палиндромным*, если оно неотрицательное, и его запись в системе счисления по основанию k читается одинаково как слева направо, так и справа налево. Иными словами, если s -значное число в системе счисления k записывается как $d_{c-1} d_{c-2} \dots d_0$ (в порядке от старших цифр к младшим), то для всех i от 0 до $c-1$ должно выполняться $d_i = d_{c-i-1}$.

Например, трехзначное число в системе счисления по основанию 100, старший разряд которого равен 45, средний разряд которого равен 65 и младший разряд которого равен 45, является палиндромным, а двухзначное число в системе счисления по основанию 50, старший разряд которого равен 11 и младший разряд которого равен 1, не является палиндромным.

Контрольная работа состоит из t заданий (задание характеризуется двумя целыми числами k и x), каждое из которых должно быть решено независимо от других. Для каждого задания необходимо найти два *k -палиндромных* числа a и b таких, что $a + b = x$. Если какое-то задание не может быть решено, сообщите об этом.

Рассмотрите примеры для лучшего понимания условия.

Формат входных данных

В первой строке следует целое число t ($1 \leq t \leq 100$) — количество заданий в контрольной работе.

В каждой из следующих t строк следуют по три целых числа k, d_1, d_0 ($2 \leq k \leq 10^9, 1 \leq d_1 < k, 0 \leq d_0 < k$) — система счисления для текущего задания, а также старшая и младшая цифры числа x для текущего задания ($x = d_0 + d_1 \cdot k$).

Формат выходных данных

Выведите ответы на все t заданий в том же порядке, в котором задания даны во входных данных.

Если ответа на задание не существует, то в первой строке ответа на задание выведите «NO» (без кавычек).

В противном случае в первую строку ответа на задание выведите «YES» (без кавычек). Во второй и третьей строках должен быть записан ответ на задание — *k -палиндромные* числа a и b в системе счисления k . Каждое число записывается в формате $cnt\ d_{cnt-1}\ d_{cnt-2} \dots d_0$, где cnt — количество цифр в числе, а далее следуют цифры числа в порядке от старшей к младшей. Каждое число не должно содержать лидирующих нулей, кроме случая, когда оно равно нулю. $A + B$ должно быть равно X из соответствующего задания.

Если существует несколько правильных ответов на какое-либо из заданий, то выведите любой из них.

Примеры

стандартный ввод	стандартный вывод
3 10 9 4 2 1 0 50 49 48	YES 2 8 8 1 6 YES 1 1 1 1 NO
2 1456432 453234 123453 2342 1231 1857	YES 2 453233 453233 1 1126652 YES 2 1231 1231 1 626

Замечание

Рассмотрим первый пример подробнее.

В первом задании используется десятичная система счисления. Два палиндромных числа, которые в сумме дадут 94 — это, например, двузначное число 88 в 10-чной системе счисления и однозначное число 6 в 10-чной системе счисления.

Во втором задании используется двоичная система счисления. Два палиндромных числа, которые в сумме дадут 10, совпадают — это однозначные числа 1 в двоичной системе счисления.

В третьем задании не существует двух палиндромных чисел, которые в сумме дадут число $48 + 49 \cdot 50$ в системе счисления по основанию 50.

Задача D. Кегельный магазин

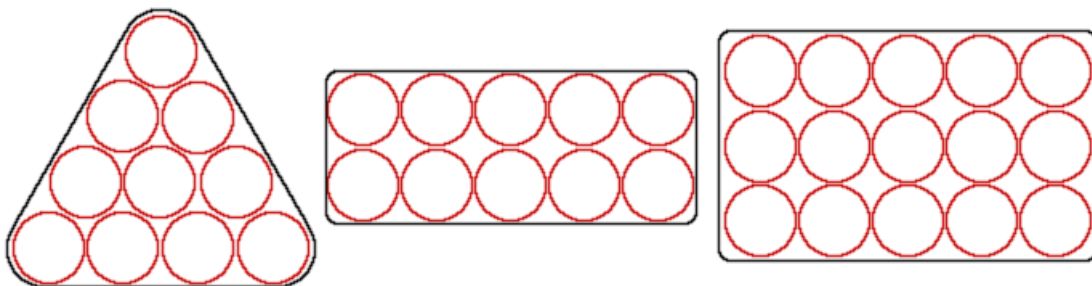
Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Петя решил открыть свой боулинг-клуб с необычными упражнениями. В данный момент ему не хватает лишь кегель.

В боулинге есть два типа упражнений, которые задаются следующими расстановками кегель:

1. треугольник: в первом ряду стоит 1 кегля, в каждом следующем ряду стоит на одну кеглю больше;
2. прямоугольник: стоящие кегли образуют **заполненный** прямоугольник со длинами сторон **строго больше 1**.

Примеры для треугольника из 4 рядов и прямоугольников из 2 рядов по 5 кегель и 3 рядов по 5 кегель:



Петя хочет купить такое количество кегель, что из них можно будет составить **как треугольник, так и прямоугольник**, и ни в одном из упражнений не останется лишних кегель. Длины сторон прямоугольника должны быть строго больше 1.

Определите максимальное количество кегель, которые Петя может купить, если его бюджета хватит не более чем на s кегель. Если Петя не может купить столько кегель, что и треугольник, и прямоугольник с длинами сторон больше 1, собираются без лишних кегель, то выведите 0.

Во входных данных будут заданы несколько тестовых наборов. Для каждого из них необходимо решить задачу независимо.

Формат входных данных

В первой строке следует целое число t ($1 \leq t \leq 1000$) — количество тестовых наборов.

В каждой из следующих t строк следует по одному целому числу s ($1 \leq s \leq 10^{18}$).

Формат выходных данных

Выведите t целых чисел, причем i -е число должно быть равно максимальному количеству кегель, которые может купить Петя в i -м тестовом наборе.

Если для некоторого тестового набора Петя не может купить столько кегель, что и треугольник, и прямоугольник можно собрать без лишних кегель, то выведите 0 для этого тестового набора.

Пример

стандартный ввод	стандартный вывод
3	10
10	15
16	0
2	

Замечание

Подробно рассмотрим пример, в котором заданы 3 тестовых набора.

В первом тестовом наборе $s = 10$. На первых двух картинках в условии изображено, что можно собрать и треугольник, и прямоугольник, которые состоят ровно из 10 кегель.

Во втором тестовом наборе $s = 16$. В этом случае можно собрать корректный прямоугольник из 16 кегель (4 ряда по 4 кегли или же 2 ряда по 8 кегель), однако треугольник из 16 кегель собрать невозможно. Прямоугольник для 15 кегель изображен на третьей картинке в условии, треугольник из 15 кегель собрать также можно, он будет состоять из 5 рядов.

В третьем тестовом наборе $s = 2$. В этом случае невозможно собрать как корректный прямоугольник (одна из его сторон будет равна 1), так и корректный треугольник, поэтому ответ 0.

Задача Е. Торт

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

Катя решила приготовить торт из коржей. У нее есть n типов коржей, при этом про i -й тип известно целое число a_i — количество коржей этого типа, которые есть у Кати.

Катя будет выкладывать коржи один на другой, при этом она **не хочет**, чтобы два коржа **одного** типа шли **подряд** в ее торте. Начинать выкладывать коржи Катя может с любого типа.

Определите максимальное количество коржей, которые Катя может использовать для приготовления своего торта.

Формат входных данных

В первой строке следует целое число n ($1 \leq n \leq 200\,000$) — количество типов коржей, которые есть у Кати.

Во второй строке следует последовательность целых чисел $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^{12}$), где a_i равно количеству коржей i -го типа, которые есть у Кати.

Следует обратить внимание, что входные данные в этой задаче не помещаются в стандартный 32-битный тип данных. Необходимо использовать 64-битный тип данных (`long long` в C++, `int64` в Паскале, `long` в Java).

Формат выходных данных

Выведите максимальное количество коржей, которые Катя может использовать для приготовления своего торта.

Примеры

стандартный ввод	стандартный вывод
3 3 1 3	7
5 1 10 1 1 1	9
2 1000000000000 1000000000000	2000000000000

Замечание

В первом примере Катя сможет использовать все коржи, которые у неё есть, для приготовления торта. Например, она может их класть в следующем порядке:

1. корж 2-го типа,
2. корж 1-го типа,
3. корж 3-го типа,
4. корж 1-го типа,
5. корж 3-го типа,
6. корж 1-го типа,
7. корж 3-го типа.

Во втором примере Катя сможет использовать 9 коржей для приготовления торта. Например, она может их класть в следующем порядке:

1. корж 2-го типа,

2. корж 3-го типа,
3. корж 2-го типа,
4. корж 5-го типа,
5. корж 2-го типа,
6. корж 4-го типа,
7. корж 2-го типа,
8. корж 1-го типа,
9. корж 2-го типа.

Задача F. Друг или враг

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

В городе живут n человек. Про некоторые пары людей известно, дружат они или враждуют.

Скоро состоятся выборы, и поэтому каждый житель города обязательно должен проголосовать **за одну из двух** партий. При этом если два жителя дружат, то они должны проголосовать за одну и ту же партию, а если враждуют — за разные.

Определите такой способ голосования, при котором будут выполняться следующие условия:

- для каждой пары друзей (a, b) верно, что человек a и человек b проголосовали за одну и ту же партию;
- для каждой пары врагов (c, d) верно, что человек c и человек d проголосовали за разные партии.

Если подходящего способа голосования не существует, сообщите об этом.

Формат входных данных

В первой строке следуют два целых числа n и m ($1 \leq n \leq 2 \cdot 10^5$, $0 \leq m \leq \min(2 \cdot 10^5, \frac{n(n-1)}{2})$) — количество человек, живущих в городе, а также количество пар людей, про которых известно, дружат они или враждуют.

В следующих m строках следуют по три целых числа x , y и t ($1 \leq x, y \leq n$, $1 \leq t \leq 2$, $x \neq y$). Если $t = 1$, то люди с номерами x и y дружат между собой (а значит, должны проголосовать за одну и ту же партию). Если $t = 2$, то люди с номерами x и y враждуют между собой (а значит, должны проголосовать за разные партии). Гарантируется, что каждая **неупорядоченная** пара людей x и y встречается во входных данных не более одного раза.

Формат выходных данных

Если способа голосования, удовлетворяющего описанным требованиям, не существует, выведите «NO» (без кавычек).

В противном случае в первую строку выведите «YES» (без кавычек), а во вторую — n целых чисел, равных 1 или 2, причем если i -е число равно 1, то человек с номером i должен проголосовать за первую партию, а если i -е число равно 2 — за вторую. Если подходящих способов голосования несколько, вы можете вывести любой из них.

Примеры

стандартный ввод	стандартный вывод
5 4 1 5 1 2 4 1 4 3 2 1 3 1	YES 1 2 1 2 1
3 3 1 2 1 2 3 1 3 1 2	NO
13 10 12 7 2 9 11 1 12 8 1 6 10 1 4 3 2 2 13 1 1 5 1 9 7 1 13 1 2 3 8 1	YES 1 2 1 2 1 1 2 1 2 1 2 1 2

Замечание

В первом примере люди с номерами 1, 3 и 5 должны проголосовать за одну партию, а люди с номерами 2 и 4 за другую, так как человек 1 дружит с 3 и 5, а 2 дружит с 4, при этом 3 и 4 являются врагами.

Во втором примере невозможно проголосовать способом, который описан в условии, так как человек 2 дружит с 1 и 3, поэтому все они должны голосовать за одну и ту же партию. Но 1 и 3 — враги, поэтому должны голосовать за разные партии. Получаем противоречие, поэтому ответ на этот пример «NO».

Задача G. Пароль

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

Катя решила поменять пароль от своей страницы в социальной сети «Вразрыве». Она нашла в Интернете большую строку s , состоящую из строчных букв латинского алфавита, и решила, что ее новым паролем будет подстрока строки s .

Подстрока строки s — это последовательность подряд идущих букв строки s произвольной (возможно нулевой) длины.

Кате нужно ввести на сайт новый пароль, чтобы он обновился, но ее клавиатура очень старая. Про каждую клавишу с буквой латинского алфавита известно количество раз, которые ее еще можно нажать. После этого клавиша сломается. Например, если букву «b» можно нажать еще три раза, а букву «e» можно нажать еще два раза, то Катя сможет набрать строки «bebeb», «ee», «bbeb», но не сможет набрать строки «bbbb», «eebbe».

Катя считает, что надежность пароля определяется его длиной — чем длиннее пароль, тем он надежнее. По заданной строке s найдите длину наибольшей ее подстроки, которая может стать паролем Кати, и которую она сможет набрать на своей клавиатуре.

Формат входных данных

В первой строке следует непустая строка s , длина которой не превышает 100 000. Строка s состоит из строчных букв латинского алфавита.

Во второй строке следуют 26 целых чисел $cnt_1, cnt_2, \dots, cnt_{26}$ ($0 \leq cnt_i \leq 100\,000$), где cnt_i — количество раз, которое можно набрать на клавиатуре i -ю букву латинского алфавита. То есть, cnt_1 — количество раз, которое можно набрать букву «a», cnt_2 — количество раз, которое можно набрать букву «b», и так далее.

Формат выходных данных

Выведите длину наибольшей подстроки строки s , которая может стать паролем Кати, и которую она сможет набрать на своей клавиатуре. Обратите внимание, что эта подстрока может быть пустой. В этом случае ее длина равна 0.

Примеры

стандартный ввод
abacaba
1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
стандартный вывод
3

стандартный ввод
aaazazbzaaa
5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3
стандартный вывод
6
стандартный ввод
aaaaaaaaa
0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
стандартный вывод
0

Замечание

В первом примере, например, паролем может стать подстрока «bac» длины 3, так как каждую из букв «a», «b» и «c» можно нажать не более одного раза.

Во втором примере паролем будет строка «aaazaz» длины 6, так как букву «b» нельзя нажимать ни одного раза.

В третьем примере пароль будет пустым, так как букву «a» нельзя нажимать ни одного раза.

Задача Н. Очередь в кассы

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

На почту пришли n посетителей, i -й из них хочет сделать a_i дел. Все n посетителей стоят в одной очереди друг за другом. Первый посетитель, стоящий в очереди, имеет номер 1, посетитель, стоящий за ним, имеет номер 2, и так далее.

На почте есть m касс, в каждой из которых сидит ровно по одному кассиру. Обслуживание i -го посетителя в кассе проходит следующим образом. Сначала проходит x_j секунд, в течение которых кассир знакомится с посетителем. После этого каждое из дел посетителя будет выполнено кассиром за y_j секунд. Посетитель не покинет кассу, пока не сделает все свои a_i дел. Таким образом, на обслуживание i -го посетителя в j -й кассе потребуется $x_j + y_j \cdot a_i$ секунд.

В каждый момент времени человек, стоящий первым в очереди, идёт в свободную кассу, если таковая имеется. Если таких касс несколько, то он выбирает кассу с минимальным номером. Если в один момент времени есть более одной кассы и более одного человека в очереди, считайте, что они подходят к кассам одновременно, но выбор кассы в первую очередь осуществляет тот, кто раньше стоит в очереди. Изначально все кассы свободны.

Перед вами стоит задача промоделировать описанный процесс и вывести время, по истечении которого будут обслужены все посетители. Также определите номера касс, которыми будут пользоваться посетители.

Формат входных данных

В первой строке следуют два целых числа n и m ($1 \leq n, m \leq 2 \cdot 10^5$) — количество посетителей и количество касс.

Во второй строке следует n целых чисел $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 100$), где a_i равно количеству дел i -го посетителя.

В следующих m строках следует по два целых числа x_j и y_j ($1 \leq x_j, y_j \leq 100$) — время, которое нужно j -му кассиру на знакомство с посетителем, а также время, которое он тратит на выполнение одного дела посетителя.

Формат выходных данных

В первую строку выведите время, по истечении которого будут обслужены все посетители.

Во вторую строку выведите n чисел, где i -е число должно быть равно номеру кассы, в которой будет обслуживаться i -й посетитель.

Примеры

стандартный ввод	стандартный вывод
3 3 1 2 3 3 4 3 2 2 1	7 1 2 3
5 1 4 3 2 4 6 5 3	82 1 1 1 1 1

Замечание

В первом примере 3 посетителя и 3 свободные кассы. Первый посетитель сразу же пойдёт в первую кассу и будет обслужен через $3 + 4 \cdot 1 = 7$ секунд, второй посетитель — во вторую кассу и будет обслужен через $3 + 2 \cdot 2 = 7$ секунд, а третий посетитель — в третью кассу и будет обслужен через $2 + 1 \cdot 3 = 6$ секунд. Таким образом, все посетители будут обслужены через 7 секунд.

Во втором примере всего одна касса, поэтому все посетители будут обслужены в ней по очереди. Суммарно на это уйдет 82 секунды.

Задача I. Тандемная строка

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

В этой задаче мы будем использовать понятие *тандемная строка*. Строка s называется тандемной, если она имеет четную длину и ее левая половина равна правой половине. Например, строки «abab», «zz», «ttbbnnttbbnn» являются тандемными, а строки «g», «abba», «zpppz» нет.

Дано n строк $s_1, s_2, \dots, s_n$, состоящих из строчных букв латинского алфавита. Определите, есть ли такая пара номеров i и j ($i \neq j$), что строка, полученная путем конкатенации строк s_i и s_j (то есть строка $s_i + s_j$), является тандемной строкой.

Формат входных данных

В первой строке следует целое число n ($2 \leq n \leq 30\,000$) — количество строк.

В i -й из следующих n строк следует непустая строка s_i , состоящая из строчных букв латинского алфавита. Длина каждой строки не превосходит 20. Строки нумеруются в том порядке, в котором они заданы во входных данных. Гарантируется, что все заданные строки **различны**.

Формат выходных данных

Если нет двух строк, конкатенация которых является тандемной строкой, выведите «NO» (без кавычек).

В противном случае в первую строку выведите «YES» (без кавычек), а во вторую строку выведите такие номера i и j ($i \neq j$), что строка $s_i + s_j$ является тандемной. Если ответов несколько, разрешается вывести любой из них.

Примеры

стандартный ввод	стандартный вывод
3 aa a aaa	YES 3 2
4 ba h abac znhzn	YES 2 4
2 ptp tp	NO

Замечание

В первом примере можно, например, соединить третью и вторую строки. Получится строка «aaaa», которая является тандемной.

Во втором примере можно, например, соединить вторую и четвертую строки. Получится строка «hznhzn», которая является тандемной.

В третьем варианте возможны всего два варианта соединений строк, в результате которых могут получиться строки «ptptp» и «tpptp». Ни одна из этих строк не является тандемной.

Задача J. Минимальное произведение

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2.5 секунд
Ограничение по памяти:	256 мегабайт

Вам заданы две перестановки одинаковой длины $a_1, a_2, \dots, a_n$ и $b_1, b_2, \dots, b_n$. Перестановка — это последовательность целых чисел, состоящая из n элементов, причем каждое число от 1 до n встречается в этой последовательности ровно один раз.

Циклическим сдвигом перестановки p на k элементов вправо называется такая перестановка q , что при $1 \leq i \leq k$ выполняется $q_i = p_{i+n-k}$, а при $k+1 \leq i \leq n$ выполняется $q_i = p_{i-k}$. Например, циклическим сдвигом перестановки $\{1, 2, 3, 4, 5\}$ на 3 элемента вправо является перестановка $\{3, 4, 5, 1, 2\}$.

Пусть функция $f(a, b)$ — это минимальное значение $a_i \cdot b_i$ среди всех i от 1 до n (формально, $f(a, b) = \min_{i=1}^n a_i b_i$). Например, если $a = \{4, 2, 1, 3\}$, $b = \{2, 1, 3, 4\}$, то $f(a, b) = 2$, так как для $i = 2$ $a_i \cdot b_i = 2$.

Определите значение описанной функции для зафиксированной перестановки a и для всех циклических сдвигов вправо перестановки b . Отметим, что циклических сдвигов вправо перестановки b , состоящей из n элементов, всего n : сдвиг на 0 элементов вправо, сдвиг на 1 элемент вправо, сдвиг на 2 элемента вправо, ..., сдвиг на $(n-1)$ элемент вправо.

Формат входных данных

В первой строке следует целое число n ($2 \leq n \leq 10^6$) — размер перестановок a и b .

Во второй строке следует последовательность $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq n$) — описание перестановки a .

В третьей строке следует последовательность $b_1, b_2, \dots, b_n$ ($1 \leq b_i \leq n$) — описание перестановки b .

Гарантируется, что последовательности a и b задают перестановки из n элементов.

Формат выходных данных

Выведите n целых чисел через пробел: i -е число должно быть равно значению описанной функции для перестановки a и сдвинутой вправо на $(i-1)$ элемент перестановки b .

Примеры

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

Замечание

Рассмотрим первый пример. Сдвиг на 0 элементов вправо перестановки b равен $\{1, 3, 4, 2, 5\}$, $f(a, b) = 2$. Сдвиг на 1 элемент вправо перестановки b равен $\{5, 1, 3, 4, 2\}$, $f(a, b) = 4$. Сдвиг на 2 элемента вправо перестановки b равен $\{2, 5, 1, 3, 4\}$, $f(a, b) = 2$. Сдвиг на 3 элемента вправо перестановки b равен $\{4, 2, 5, 1, 3\}$, $f(a, b) = 1$. Сдвиг на 4 элемента вправо перестановки b равен $\{3, 4, 2, 5, 1\}$, $f(a, b) = 4$.

Задача К. Равномерное деление дерева

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 4 секунды
Ограничение по памяти: 512 мегабайт

Дано неориентированное невзвешенное дерево, состоящее из n вершин. Дерево из n вершин — это связный ациклический граф, состоящий из $n - 1$ ребра.

Необходимо удалить из своего дерева ровно 2 ребра. Это нужно сделать таким образом, чтобы разность между размерами **максимального** и **минимального** деревьев, получившихся после удаления, была минимальна. Очевидно, что после удаления ровно 2 ребер дерево разобьется на ровно 3 не связанных между собой дерева. Под размером дерева подразумевается количество вершин в нем.

Обратите внимание, что при удалении ребра (u, v) **концы ребра не удаляются** (то есть вершины u и v остаются в дереве), удаляется только само ребро.

Формат входных данных

В первой строке следует целое число n ($3 \leq n \leq 2 \cdot 10^5$) — количество вершин в дереве.

В каждой из следующих $(n - 1)$ строк следуют по два целых числа a_i, b_i ($1 \leq a_i, b_i \leq n, a_i \neq b_i$) — описание ребер дерева. Гарантируется, что заданный граф является деревом.

Формат выходных данных

Выведите минимальную разность между размерами **максимального** и **минимального** деревьев, получившихся после удаления Катей ровно двух рёбер из заданного дерева.

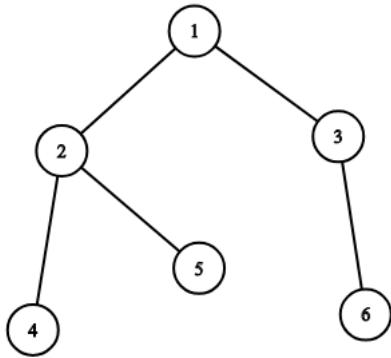
Обратите внимание, что при удалении ребра (u, v) **концы ребра не удаляются** (то есть вершины u и v остаются в дереве), удаляется только само ребро.

Примеры

стандартный ввод	стандартный вывод
6 1 3 1 2 2 5 2 4 3 6	2
10 1 2 1 3 3 8 8 9 8 10 2 4 4 7 2 5 5 6	2
9 1 2 1 3 3 4 3 7 7 8 8 9 2 5 5 6	0

Замечание

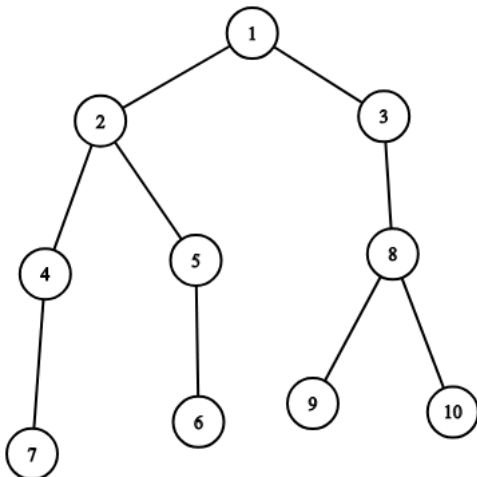
Картинка, соответствующая дереву из первого тестового примера:



Ответ 2 можно получить, удалив любую из следующих пар ребер:

- (1, 2), (1, 3);
- (1, 2), (2, 4);
- (1, 3), (2, 4);
- (1, 2), (2, 5);
- (1, 3), (2, 5);
- (1, 2), (3, 6).

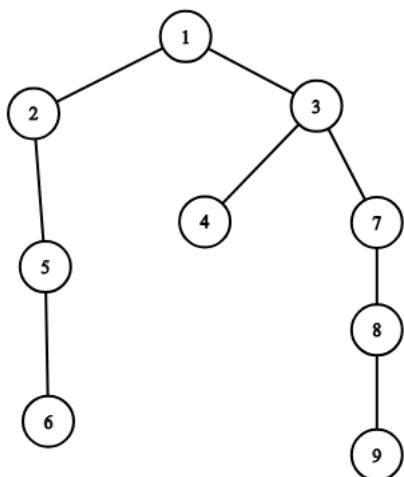
Картинка, соответствующая дереву из второго тестового примера:



Ответ 2 можно получить, удалив любую из следующих пар ребер:

- (1, 3), (2, 4);
- (1, 3), (2, 5).

Картинка, соответствующая дереву из третьего тестового примера:



Ответ 0 можно получить, удалив следующую пару ребер:

- $(1, 2), (3, 7)$.