

Задача А. Лунтик и концерты

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

Лунтик решил попробовать себя в пении. У него есть a одноминутных песен, b двухминутных и c трёхминутных. Он хочет распределить все песни на два концерта, при этом каждая песня должна быть ровно в одном концерте.

Он хочет, чтобы разница между длительностями концертов была как можно меньше. Длительность концерта — это сумма длительностей всех песен в концерте.

Помогите Лунтику найти наименьшую возможную разницу в минутах между длительностями концертов.

Формат входных данных

В первой строке находится единственное целое число t ($1 \leq t \leq 1000$) — количество наборов входных данных.

Каждый набор входных данных состоит из одной строки, содержащей три целых числа a, b, c ($1 \leq a, b, c \leq 10^9$) — количество одноминутных, двухминутных и трёхминутных песен.

Формат выходных данных

Для каждого набора входных данных выведите наименьшую возможную разницу в минутах между длительностями концертов.

Пример

стандартный ввод	стандартный вывод
4	0
1 1 1	1
2 1 3	0
5 5 5	1
1 1 2	

Замечание

В первом наборе входных данных Лунтик может на первом концерте исполнить одноминутную и двухминутную песни, а на втором — трехминутную. Тогда разница между длительностями будет равна 0.

Задача В. Составить число

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

Лунтик построил себе новый дом и теперь хочет, чтобы у этого дома был настоящий адрес. Дом располагается далеко от всех улиц города, поэтому мэр разрешил Лунтику самому выбрать себе номер дома. Это предложение очень понравилось Лунтику, и он решил выбрать в качестве номера наибольшее число, которое сможет.

Чтобы друзья могли легко найти домик Лунтика, он должен нанести номер дома на стену. Для этого нужно использовать цифры, которые продаются в магазине рядом. Всего в продаже имеются k цифр, $0, 1 \dots k-1$, и для каждой цифры в этом магазине есть своя цена, цифра i стоит a_i монет. У Лунтика имеется x монет, поэтому он должен выбрать такой номер дома, чтобы ему хватило монет для покупки всех необходимых цифр, присутствующих в этом числе.

Номер дома может получиться очень длинным, поэтому сейчас Лунтику интересно сколько цифр будет в этом номере, а также первые и последние 50 цифр этого числа. Номером дома является целое неотрицательное число. Помогите Лунтику.

Формат входных данных

В первой строке дано натуральное число k ($1 \leq k \leq 10$) — количество имеющихся в продаже цифр. Следующие k строк содержат по одному натуральному числу a_i ($1 \leq a_i \leq 10^{18}$) — цене цифры $0, 1 \dots k-1$ соответственно.

В следующей строке дано целое число x ($0 \leq x \leq 10^{18}$) — количество монет, имеющихся у Лунтика.

Формат выходных данных

В первой строке выведите количество цифр в наибольшем номере дома, который может выбрать Лунтик.

Во второй строке выведите первые 50 цифр номера дома Лунтика. Если в числе меньше цифр, выведите только число, без ведущих нулей.

В третьей строке выведите последние 50 цифр номера дома Лунтика. Если в числе меньше цифр, выведите только число, без ведущих нулей.

Если монет не хватит ни на одну цифру, в единственной строке выведите ноль.

Примеры

стандартный ввод	стандартный вывод
3 6 7 8 21	3 210 210
3 1 7 8 21	15 1000000000000000 1000000000000000

Задача С. Баба Капа вяжет шарф

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

Баба Капа решила связать шарф и попросила Деда Шера сделать для него шаблон, представляющий собой строку из строчных букв латинского алфавита. Дед Шер написал строку s длины n .

Баба Капа хочет связать красивый шарф, а по ее мнению красивый шарф можно связать только из строки, являющейся палиндромом. Она хочет изменить шаблон, написанный Дедой Шером, но, чтобы его сильно не обидеть, она выберет одну любую строчную букву латинского алфавита и уберет какие-то (на свой выбор, возможно никакие или все) вхождения этой буквы в строку s .

При этом она хочет, чтобы количество удаленных из шаблона символов было как можно меньше. Помогите ей и скажите, какое минимальное количество символов она может удалить, чтобы строка s стала палиндромом, или скажите, что это невозможно. Заметьте, что она может удалять только символы, равные **одной** букве, которую она выбрала.

Строка называется палиндромом, если она одинаково читается слева направо и справа налево. Например, строки 'kek', 'abacaba', 'r' и 'papicipap' — палиндромы, а строки 'abb' и 'iq' — нет.

Формат входных данных

Первая строка содержит единственное целое число t ($1 \leq t \leq 100$) — количество наборов входных данных. Следующие $2 \cdot t$ строк содержат описание наборов входных данных. Описание каждого набора входных данных состоит из двух строк.

Первая строка описания каждого набора входных данных содержит единственное целое число n ($1 \leq n \leq 10^5$) — длину строки.

Вторая строка описания каждого набора входных данных содержит строку s из n строчных букв латинского алфавита.

Гарантируется, что сумма значений n по всем наборам входных данных не превосходит $2 \cdot 10^5$.

Формат выходных данных

Для каждого набора входных данных выведите минимальное количество символов, которое потребуется удалить Бабе Капе, чтобы строка стала палиндромом, если это возможно, а если невозможно, выведите -1 .

Пример

стандартный ввод	стандартный вывод
5	2
8	-1
abcaacab	0
6	3
xuzxyz	2
4	
abba	
8	
rprarlarp	
10	
khyyhhyhky	

Замечание

В первом наборе входных данных можно выбрать букву 'a' и удалить ее первое и последнее вхождение, получится строка 'bcaacb', являющаяся палиндромом. Также можно выбрать букву 'b' и удалить все ее вхождения, получится строка 'acaaca', также являющаяся палиндромом.

Во втором наборе входных данных можно показать, что нельзя выбрать букву и удалить какие-то ее вхождения так, чтобы получилась строка-палиндром.

В третьем наборе входных данных можно ничего не удалять, строка и так является палиндромом.

Задача D. Игра на цикле

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

Это интерактивная задача

Лунтик и Кузя играют в игру. Они нарисовали на песке граф из n вершин, пронумеровали их от 1 до n и соединили вершину 1 с вершиной 2, вершину 2 с вершиной 3, ... вершину $n - 1$ с вершиной n и вершину n с вершиной 1. Таким образом, на песке получился неориентированный простой цикл на n вершинах.

Кузя придумал вес для каждого из n ребер этого цикла — натуральное число, не превосходящее 10^6 . Задача Лунтика — угадать вес каждого ребра. Для этого он может делать следующие запросы к Кузе:

- Узнать длину кратчайшего пути между двумя произвольными различными вершинами u и v
- Узнать длину длиннейшего простого пути между двумя произвольными различными вершинами u и v

Лунтик может сделать не больше чем $n + 1$ запрос. Кузя играет честно, поэтому не будет менять веса ребер в процессе игры. Помогите Лунтику выиграть — сыграйте вместо него с Кузей!

Протокол взаимодействия

Сначала на вход подаётся одно целое число n ($3 \leq n \leq 1000$).

Чтобы сделать запрос первого типа (узнать длину кратчайшего пути), выведите строку вида «`min u v`», где u и v — интересующие вас вершины ($u \neq v$).

Чтобы сделать запрос второго типа (узнать длину длиннейшего простого пути), выведите строку вида «`max u v`», где u и v — интересующие вас вершины ($u \neq v$).

После каждого запроса считайте одно целое число s — ответ на ваш запрос.

Когда будете готовы угадать вес каждого ребра, выведите строку, содержащую восклицательный знак («!») и затем n натуральных чисел $w_1, w_2, \dots, w_n$ ($1 \leq w_i \leq 10^6$) через пробел — веса ребер между вершинами 1 и 2, 2 и 3, ... n и 1 соответственно. После вывода этой строки ваша программа должна завершиться.

В случае, если ваш запрос имеет неверный формат, или вы сделали более $n + 1$ запроса, вы получите вердикт «Неправильный ответ».

После вывода запроса не забудьте вывести перевод строки и сбросить буфер вывода. В противном случае вы получите вердикт Решение «зависло». Для сброса буфера используйте:

- `fflush(stdout)` или `cout.flush()` в C++;
- `System.out.flush()` в Java;
- `flush(output)` в Pascal;
- `stdout.flush()` в Python;
- смотрите документацию для других языков.

Гарантируется, что веса ребер зафиксированы и не будут меняться в процессе взаимодействия.

Примеры

стандартный ввод	стандартный вывод
4	max 1 3
7	min 1 3
3	max 2 4
5	min 2 4
5	max 2 3
8	! 1 2 3 4
3	min 1 3
7	max 1 3
14	! 7 7 7

Задача Е. Лунтик и подпоследовательности

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

Лунтик вышел на утреннюю прогулку и нашел массив a длины n . Он посчитал сумму s элементов массива ($s = \sum_{i=1}^n a_i$). Лунтик называет подпоследовательность массива a *почти полной*, если сумма чисел в этой подпоследовательности равна $s - 1$.

Лунтик очень хочет узнать количество *почти полных* подпоследовательностей массива a . Но ему надо возвращаться домой, поэтому он просит вас решить эту задачу!

Напоминаем, что последовательность x является подпоследовательностью y , если x может быть получена из y удалением нескольких (возможно, ни одного или всех) элементов.

Формат входных данных

В первой строке находится единственное целое число t ($1 \leq t \leq 1000$) — количество наборов входных данных. Следующие $2 \cdot t$ строк содержат описания наборов входных данных. Описание каждого набора состоит из двух строк.

Первая строка описания каждого набора входных данных содержит единственное целое число n ($1 \leq n \leq 60$) — длину массива.

Во второй строке находится n целых чисел $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 10^9$) — элементы массива a .

Формат выходных данных

Для каждого набора входных данных выведите количество *почти полных* подпоследовательностей массива.

Пример

стандартный ввод	стандартный вывод
5	1
5	0
1 2 3 4 5	2
2	4
1000 1000	4
2	
1 0	
5	
3 0 2 1 1	
5	
2 1 0 3 0	

Замечание

В первом наборе входных данных $s = 1 + 2 + 3 + 4 + 5 = 15$, из всех подпоследовательностей почти полной будет только $(2, 3, 4, 5)$, сумма в ней равна $2 + 3 + 4 + 5 = 14 = 15 - 1$.

Во втором наборе входных данных нет почти полных подпоследовательностей.

В третьем наборе входных данных $s = 1 + 0 = 1$, почти полными будут подпоследовательности (0) и $()$ (пустая подпоследовательность имеет сумму 0).

Задача F. Кузя и конфеты

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

Все друзья Лунтика очень любят конфеты. Но есть их в большом количестве вредно, поэтому они предпочитают играть с ними в такую игру: пусть есть x конфет, игроку требуется разложить их по кучкам так, чтобы размеры кучек были последовательными натуральными числами. Например, 15 конфет можно разложить на кучки (4, 5, 6) или (7, 8), можно всё сложить и в одну кучку (15), но на кучки (3, 5, 7) разложить нельзя.

У Лунтика намечается день рождения. Кузя планирует подарить ему коробку конфет, но понимает, что как только он вручит имениннику подарок, все присутствующие гости сразу сядут играть с конфетами. Гости будут раскладывать конфеты на кучки, причем, разложив один раз, они захотят разложить их еще несколько раз другими способами. Когда они разложат конфеты на кучки всеми возможными способами, им станет скучно, и они уйдут.

Дома у Кузи есть n коробок конфет. Он хочет, чтобы веселье затянулось на подольше, поэтому выберет в качестве подарка ту из них, которая имеет наибольшее количество разбиений, удовлетворяющих правилам игры. Если же несколько коробок конфет будут иметь максимальное количество разбиений, то Кузя выберет ту, где меньше всего конфет (потому что нести подарок до дома Лунтика тяжело). Помогите Кузе определить, сколько конфет он подарит Лунтику и количество разбиений на кучки, которое может получиться из этого количества конфет.

Формат входных данных

В первой строке входных данных записано целое число n ($2 \leq n \leq 1000$) — количество коробок конфет у Кузи дома.

Во второй строке записаны n целых чисел $a_1, a_2, \dots, a_n$, где a_i ($1 \leq a_i \leq 2 \cdot 10^9$) обозначает количество конфет в i -й коробке.

Формат выходных данных

В единственной строке выведите два числа: количество конфет, которое Кузя возьмёт к Лунтику, и количество разбиений на кучки, которое может получиться из этого количества конфет.

Примеры

стандартный ввод	стандартный вывод
2 15 30	15 4
3 20 30 20	30 4

Замечание

Первый пример. 15 конфет можно разбить на кучки четырьмя способами: (1, 2, 3, 4, 5), (4, 5, 6), (7, 8), (15). 30 конфет тоже можно разбить на кучки четырьмя способами: (4, 5, 6, 7, 8), (6, 7, 8, 9), (9, 10, 11), (30). Но Кузя подарит Лунтику первую коробку, потому что в ней меньше конфет.

Второй пример. 30 конфет можно разбить на кучки четырьмя способами, как описано выше. 20 конфет можно разбить на кучки двумя способами: (2, 3, 4, 5, 6), (20). Поэтому Кузя подарит Лунтику вторую коробку.

Задача G. Корней Корнеевич и XOR

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1.5 секунд
Ограничение по памяти: 512 мегабайт

Корней Корнеевич откопал массив a длины n . Корней Корнеевич недавно прочитал в книге про операцию побитового исключающего «ИЛИ» (XOR), поэтому он захотел поэкспериментировать с ней. Для этого он решил найти все целые $x \geq 0$ такие, что существует **возрастающая** подпоследовательность массива a , в которой побитовое исключающее «ИЛИ» ее чисел равно x .

Корней Корнеевич довольно быстро нашел все такие x , и хочет проверить свой результат. Для этого он попросил вас решить эту задачу!

Последовательность s является подпоследовательностью b , если s может быть получена из b удалением нескольких (возможно, ни одного или всех) элементов.

Последовательность $s_1, s_2, \dots, s_m$ называется возрастающей, если выполняется неравенство: $s_1 < s_2 < \dots < s_m$.

Формат входных данных

Первая строка входных данных содержит одно целое число n ($1 \leq n \leq 10^6$).

Вторая строка содержит n натуральных чисел $a_1, a_2, \dots, a_n$ ($0 \leq a_i \leq 5000$) — элементы массива a .

Формат выходных данных

В первой строке выходных данных выведите единственное число k — количество найденных вами x .

Во второй строке выходных данных выведите k целых неотрицательных чисел в **возрастающем** порядке $x_1, x_2, \dots, x_k$ ($0 \leq x_1 < \dots < x_k$) — найденные вами x .

Примеры

стандартный ввод	стандартный вывод
4 4 2 2 4	4 0 2 4 6
8 1 0 1 7 12 5 3 2	12 0 1 2 3 4 5 6 7 10 11 12 13

Замечание

Напомним, что XOR или «Исключающим ИЛИ», называется бинарная операция, которая применяется к каждой паре битов, стоящих на одинаковых позициях в двоичных представлениях чисел так, что, если биты равны, то в результате на этой позиции стоит 0, если же биты различны, то 1. Например, $3 \text{ XOR } 5 = 6$, потому что $3_{10} = 011_2$, $5_{10} = 101_2$, поэтому после применения операции второй и третий бит становятся равными 1, а первый бит — 0, таким образом получается $110_2 = 6_{10}$.

Задача Н. Пчеленок и подотрезки

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 2 секунды
Ограничение по памяти: 512 мегабайт

Пчеленок решил сделать Миле подарок. Пчеленок уже купил массив a длины n , но дарить просто массив слишком банально. Вместо этого он решил подарить Миле подотрезки этого массива!

Пчеленок хочет, чтобы его подарок был красивым, поэтому он решил выбрать k непересекающихся подотрезков массива $[l_1, r_1], [l_2, r_2], \dots, [l_k, r_k]$ так, что:

- длина первого отрезка $[l_1, r_1]$ равна k , длина второго отрезка $[l_2, r_2]$ равна $k-1, \dots$, длина k -го отрезка $[l_k, r_k]$ равна 1
- для любых $i < j$ i -й подотрезок встречается в массиве раньше j -го (т.е. $r_i < l_j$)
- суммы в этих подотрезках строго возрастают (т.е. пусть $sum(l \dots r) = \sum_{i=l}^r a_i$ — сумма чисел на подотрезке массива $[l, r]$, тогда $sum(l_1 \dots r_1) < sum(l_2 \dots r_2) < \dots < sum(l_k \dots r_k)$).

Пчеленок также хочет, чтобы его подарок был как можно больше, поэтому он просит вас узнать, при каком максимальном k он сможет сделать подарок Миле!

Формат входных данных

Первая строка содержит одно целое число t ($1 \leq t \leq 100$) — количество наборов входных данных. Следующие $2 \cdot t$ строк содержат описания наборов входных данных. Описание каждого набора состоит из двух строк.

Первая строка описания каждого набора входных данных содержит единственное целое число n ($1 \leq n \leq 10^5$) — длину массива.

Вторая строка содержит n целых чисел $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$) — элементы массива a .

Гарантируется, что сумма значений n по всем наборам входных данных не превосходит 10^5 .

Формат выходных данных

Для каждого набора входных данных выведите максимальное возможное значение k .

Пример

стандартный ввод	стандартный вывод
5	1
1	1
1	2
3	3
1 2 3	1
5	
1 1 2 2 3	
7	
1 2 1 1 3 2 6	
5	
9 6 7 9 7	

Задача I. День рождения Милы

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

У Милы сегодня день рождения! На её праздник пришли n гостей (для удобства пронумеруем их от 1 до n), каждый из которых принёс одну конфетку. Также на праздник пришел Лунтик, который принёс огромный мешок конфет. Собравшись, все гости вместе с Милой начинают играть в следующую игру.

Мила начинает по очереди называть номера гостей (Мила может повторяться). Когда Мила называет число x , Лунтик считает сколько конфет у всех гостей кроме гостя x (пусть их s). Затем он достаёт из своего мешка s конфет и отдаёт их гостю с номером x .

Мила хочет с помощью этой игры добиться того, чтобы у i -го гостя было a_i конфет. Помогите ей определить план действий, чтобы достичь своей цели, либо определите, что это невозможно.

Формат входных данных

В первой строке входных данных содержится одно натуральное число n ($1 \leq n \leq 1000$) — число гостей Милы. Во второй строке содержится n натуральных чисел $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^9$).

Формат выходных данных

Если у Милы есть последовательность действий, выведите в первой строке «Yes» без кавычек. Во второй строке выведите число k — сколько раз Миле предстоит называть номер гостя. Если k не превышает 100 000, то в третьей строке выведите k натуральных чисел через пробел — номера гостей, в том порядке, в котором их должна называть Мила. Если же $k > 100\,000$, то ничего выводить не нужно.

Если у Милы не получится добиться того, чтобы у i -го гостя было a_i конфет в конце игры, выведите «No» без кавычек.

Примеры

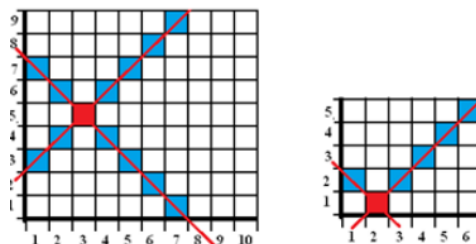
стандартный ввод	стандартный вывод
3 1 1 3	Yes 1 3
2 5 2	Yes 3 2 1 1
5 1 2 3 4 5	No

Задача J. Диагонали

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

Лунтику дали прямоугольную сетку, состоящую из n строк и m столбцов. Строки пронумерованы снизу вверх числами от 1 до n , а столбцы — слева направо числами от 1 до m .

Вначале все клетки были белыми. Затем Лунтик взял квадрат в столбце c строки r и закрасил его красным. Потом от этого квадрата во все четыре стороны провёл диагонали через центр красного квадрата и его углы. И все квадраты, через которые прошли диагонали, закрасил синим.



На рисунке показаны два примера: на левой сетке значения $n = 9$, $m = 10$, $r = 5$, $c = 3$ на правой $n = 5$, $m = 6$, $r = 1$, $c = 2$.

Цветные квадраты делят сетку максимум на 4 области соседних по стороне белых квадратов. В левом примере областей 4, а в правом примере 3. Вычислите, сколько белых квадратов в каждой из четырех областей.

Формат входных данных

На вход подаются четыре натуральных числа n, m, r и c ($5 \leq n, m \leq 10^9$, $1 \leq r \leq n$, $1 \leq c \leq m$).

Формат выходных данных

Выведите четыре числа в порядке возрастания — количество белых квадратов в каждой области. Если область отсутствует, количество квадратов в ней равно нулю.

Примеры

стандартный ввод	стандартный вывод
9 10 5 3	4 15 15 43
5 6 1 2	0 1 10 13

Задача К. Обход дерева

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

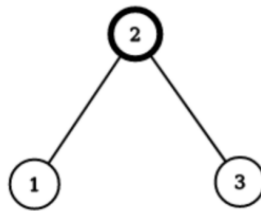
На очередном уроке информатики Лунтик рассказывал о том, как обходить подвешенное дерево.

Дерево — это связный неориентированный граф без циклов. *Подвешенное дерево* — дерево, у которого одна из вершин выбрана в качестве корня. *Родителем* вершины называется соседняя вершина, лежащая на кратчайшем пути к корню. У корня родителя нет. Соседние вершины, не являющиеся родителем, называются *детьми*. Все вершины дерева пронумерованы числами от 1 до n , где n — количество вершин.

Лунтик обходит дерево, посещая все его вершины в определённом порядке. Сначала посещается корень. После этого для каждой посещённой вершины выполняется алгоритм:

- Если у вершины есть дети, которые ещё не были посещены, то выбираем из них ребёнка с **минимальным номером** и посещаем его;
- Если у вершины нет непосещённых детей, то посещаем её родителя, или же заканчиваем обход, когда родителя нет.

Обходом называется последовательность посещённых вершин.



Например, обход данного дерева — 2, 1, 2, 3, 2.

Вупсень и Пупсень, присутствовавшие на уроке информатики, решили попрактиковаться в новой теме. Они выбрали какое-то дерево и начали его обходить, выписывая посещённые вершины. Но вот только Вупсень добавлял вершину в свой обход лишь тогда, когда она была посещена в первый раз, а Пупсень добавлял, когда вершина была посещена в последний раз. Т.е. для представленного выше дерева их обходы получились бы такими: 2, 1, 3 у Вупсения и 1, 3, 2 у Пупсения.

Лунтик увидел, что они наделали, и заинтересовался — можно ли по обходам Вупсения и Пупсения установить, какое дерево они выбрали для обхода? Эту задачу вам и предстоит решить.

Формат входных данных

В первой строке дано целое число n — размер обходов Вупсения и Пупсения ($1 \leq n \leq 3000$).

Во второй строке даны n различных целых чисел a_i — обход Вупсения ($1 \leq a_i \leq n$).

В третьей строке даны n различных целых чисел b_i — обход Пупсения ($1 \leq b_i \leq n$).

Гарантируется, что дерево с такими обходами Вупсения и Пупсения существует.

Формат выходных данных

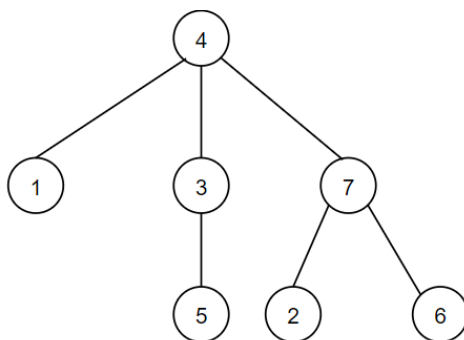
Выведите n целых чисел. i -е число должно быть равно номеру родителя вершины с номером i в искомом дереве, или равно 0, если у i -й вершины нет родителя. Если существует несколько деревьев, подходящих под обходы Вупсения и Пупсения, выведите информацию о любом.

Примеры

стандартный ввод	стандартный вывод
3 2 1 3 1 3 2	2 0 2
7 4 1 3 5 7 2 6 1 5 3 2 6 7 4	4 7 4 0 3 7 4

Замечание

Во втором примере Вупсень и Пупсень обходили такое дерево с корнем в вершине 4. Обход дерева получился 4, 1, 4, 3, 5, 3, 4, 7, 2, 7, 6, 7, 4, поэтому Вупсень выписал 4, 1, 3, 5, 7, 2, 6, а Пупсень 1, 5, 3, 2, 6, 7, 4.



Задача L. Лунтик, Луна и игра

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

Лунтик со своей подружкой решили поиграть в увлекательную игру. Луна выбирает на плоскости несколько целочисленных точек, после этого Лунтик тоже должен выбрать некоторую целочисленную точку A , которая возможно совпадает с одной из выбранных Луной. Задачей Лунтика является выбрать точку A таким образом, чтобы нашлись две точки из выбранных Луной таких, что точка A лежит на отрезке с концами в этих точках, причем строго внутри.

Лунтик понимает, что эта задача непростая, особенно когда точек становится много, поэтому просит вас помочь ему.

Формат входных данных

В первой строке вводится натуральное число n ($1 \leq n \leq 10^5$) — число выбранных Луной точек. В последующих n строках даны координаты этих точек, пары целых чисел не превосходящих по модулю 10^9 . Все точки попарно различны.

Формат выходных данных

Если такая точка существует, в первой строке выведите «YES», а во второй координаты точки A , и координаты двух точек из множества Луны, на отрезке между которыми лежит точка A .

Если же такой точки нет, выведите «NO»

Примеры

стандартный ввод	стандартный вывод
2 0 0 3 -3	YES 1 -1 0 0 3 -3
3 -8 -1 4 6 5 -7	NO

Замечание

В первом примере так же подходит точка $(2; -2)$. Других подходящих точек нет.

Задача М. Лунтик и квадраты

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

Однажды Лунтик захотел проверить теорию о том, что лунные камни со временем теряют в весе, но в его распоряжении были только большие чашечные весы. В магазине неподалёку продаются 10^6 квадратных гирек, вес i -й из которых равен i^2 грамм. Однако не всё так просто, из-за причудливой акции в этом магазине гирьки продаются только наборами вида «первые k гирек» для любого $k \leq 10^6$. Таким образом покупатель может выбрать некоторое k , и он получит гирьки весами $1, 4, 9, \dots, k^2$.

Лунтик знает, что имеющийся у него лунный камень раньше весил w грамм, и теперь он хочет выбрать некоторое k , чтобы проверить действительно ли камень по-прежнему весит w грамм. После того как Лунтик купит какой-то набор, он хочет использовать для взвешивания все гирьки оттуда, так как иначе окажется, что он зря покупал и нёс домой какую-то гирьку, что его очень расстроит.

При взвешивании Лунтик может класть гирьки на любую чашу весов.

Формат входных данных

Вводится одно натуральное число w ($1 \leq w \leq 10^5$) — вес, который нужно набрать.

Формат выходных данных

Выведите строку, длина которой равна числу гирек, которые нужно купить. i -й символ строки должен быть «+» или «-», в зависимости от того, на какую чашу нужно ставить i -ю гирьку. После такой расстановки, разность сумм всех гирек на чашках «+» и «-» должна равняться w . На какой из чашек вес будет больше не имеет значения.

Длина строки не должна превышать 10^6 . Можно показать, что ответ всегда существует. Обратите внимание, что вам не нужно минимизировать длину строки.

Пример

стандартный ввод	стандартный вывод
3	-+

Задача N. Вупсень, Пупсень и 0

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

Вупсеню и Пупсеню подарили массив целых чисел. Так как Вупсень не любит число 0, он выкинул все числа, равные 0 из массива. В итоге получился массив a длины n .

Пупсень, напротив, любит число 0 и расстроился, когда увидел массив без нулей. Чтобы подбодрить Пупсенья, Вупсень решил придумать другой массив b длины n такой, что $\sum_{i=1}^n a_i \cdot b_i = 0$. Так как Вупсень не любит число 0, **в массиве b не должно быть чисел, равных 0**. К тому же числа в этом массиве не должны быть слишком большими, поэтому **сумма их абсолютных значений не может превышать 10^9** . Помогите Вупсеню найти любой такой массив b !

Формат входных данных

В первой строке находится единственное целое число t ($1 \leq t \leq 100$) — количество наборов входных данных. Следующие $2 \cdot t$ строк содержат описания наборов входных данных. Описание каждого набора состоит из двух строк.

Первая строка описания каждого набора входных данных содержит единственное целое число n ($2 \leq n \leq 10^5$) — длину массива.

Вторая строка содержит n целых чисел $a_1, a_2, \dots, a_n$ ($-10^4 \leq a_i \leq 10^4$, $a_i \neq 0$) — элементы массива a .

Гарантируется, что сумма значений n по всем наборам входных данных не превосходит $2 \cdot 10^5$.

Формат выходных данных

Для каждого набора входных данных выведите n целых чисел $b_1, b_2, \dots, b_n$ — элементы массива b ($|b_1| + |b_2| + \dots + |b_n| \leq 10^9$, $b_i \neq 0$, $\sum_{i=1}^n a_i \cdot b_i = 0$).

Можно показать, что ответ всегда существует.

Пример

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

Замечание

В первом наборе входных данных $5 \cdot 1 + 5 \cdot (-1) = 5 - 5 = 0$. Также можно было, например, вывести 3 -3, так как $5 \cdot 3 + 5 \cdot (-3) = 15 - 15 = 0$

Во втором наборе входных данных $5 \cdot (-1) + (-2) \cdot 5 + 10 \cdot 1 + (-9) \cdot (-1) + 4 \cdot (-1) = -5 - 10 + 10 + 9 - 4 = 0$.

Задача О. Кузя и домашнее задание

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	512 мегабайт

Кузя начал ходить в школу. На уроке математики ему дали домашнее задание. Для этого задания ему выдали массив натуральных чисел a длины n и массив символов b длины n , состоящий только из символов $*$ или $/$.

Давайте определим *траекторию вычислений* для отрезка $[l; r]$ ($1 \leq l \leq r \leq n$) следующим образом:

- Пусть изначально $x = 1$. Для всех i от l до r последовательно выполним следующее: если $b_i = *$, $x = x * a_i$, а если $b_i = /$, то $x = \frac{x}{a_i}$. *Траекторией вычисления* для отрезка $[l; r]$ назовем список всех x , которые были получены в процессе вычислений (их будет ровно $r - l + 1$).

Например, пусть $a = [7, 12, 3, 5, 4, 10, 9]$, $b = [/ , *, / , / , *, *]$, $l = 2$, $r = 6$, тогда траектория вычислений для этого отрезка — это $[12, 4, 0.8, 0.2, 2]$.

Давайте назовем отрезок $[l; r]$ *простым*, если траектория вычислений для него содержит **только целые числа**.

Кузе нужно посчитать количество простых отрезков $[l; r]$ ($1 \leq l \leq r \leq n$). Так как у него явно нет ни времени, ни желания проводить вычисления для всех вариантов, он попросил вас написать программу, чтобы узнать это количество!

Формат входных данных

Первая строка содержит единственное целое число n ($2 \leq n \leq 10^6$).

Вторая строка содержит n целых чисел $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^6$).

Третья строка содержит n символов без пробелов между ними — массив $b_1, b_2 \dots b_n$ ($b_i = /$ или $b_i = *$ для всех $1 \leq i \leq n$).

Формат выходных данных

В единственной строке выведите одно целое число — количество простых отрезков $[l; r]$.

Примеры

стандартный ввод	стандартный вывод
3 1 2 3 */*	2
7 6 4 10 1 2 15 1 */**//	8