

## Задача А. Побег Майлза

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

Майлз опять сбегает от Мигеля О'Хары, но в этот раз он смог захватить с собой устройство для перемещения между двумя мирами. К сожалению, его перемещения ограничены одним городом в каждом из миров. Но эти два города очень похожи: в каждом из них ровно  $n$  небоскребов, располагающихся в одних и тех же точках пространства.

Некоторые пары небоскребов, расположенных в одном мире, пригодны для того, чтобы протянуть между ними паутину и переместиться с одного на другой (в любом из двух направлений). В первом мире есть ровно  $m_1$  таких пар небоскребов, а во втором — ровно  $m_2$ . Известно, за какое время можно переместиться между доступными парами небоскребов в каждом мире. Помимо этого Майлз может, находясь на  $i$ -м небоскребе в первом мире, переместиться на  $i$ -й небоскреб во втором, и наоборот, за  $x$  секунд.

Майлз собирается встретиться со своей командой на небоскребе номер  $t$  второго мира, при этом начинает он побег с небоскреба номер  $s$  первого мира. Помогите Майлзу и скажите, как быстро он сможет встретиться со своей командой, чтобы иметь шансы против Мигеля.

### Формат входных данных

Первая строка ввода содержит два целых числа  $n$  и  $x$  — количество небоскребов в каждом из двух миров и время перемещения между соответствующими небоскребами разных миров ( $1 \leq n \leq 10^5$ ;  $1 \leq x \leq 10^6$ ).

Вторая строка содержит число  $m_1$  — количество пар небоскребов, между которыми можно перемещаться в городе первого мира ( $0 \leq m_1 \leq 10^6$ ).

Следующие  $m_1$  строк содержат по три числа  $u_i$ ,  $v_i$  и  $c_i$ , означающих, что между небоскребами  $u_i$  и  $v_i$  в первом мире можно переместиться в любом направлении за  $c_i$  секунд ( $1 \leq u_i, v_i \leq n$ ;  $1 \leq c_i \leq 10^6$ ).

В следующих строках в таком же формате содержится информация о возможных перемещениях между небоскребами второго мира: в первой из этих строк дано число  $m_2$ , а следующие  $m_2$  строк содержат сами описания перемещений (в виде троек чисел  $u_i$ ,  $v_i$  и  $c_i$ ).

Последняя строка содержит два целых числа  $s$  и  $t$  — номер стартового небоскреба в первом мире и конечного во втором ( $1 \leq s, t \leq n$ ).

### Формат выходных данных

Выведите единственное целое число — минимальное время путешествия между небоскребом  $s$  первого мира и небоскребом  $t$  второго мира, или  $-1$ , если между ними нет пути.

## Пример

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

## Задача В. Ловля пауков

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

Вы вступили на сторону зла и теперь работаете на «Alchemax»!

В лаборатории «Alchemax» начали отлавливать пауков из других вселенных. Но это оказалось непростой задачей, так как каждого паука еще нужно кормить. Для этого в лабораториях изготавливают специальный корм, но изготовление корма для пауков из различных вселенных — это весьма затратное занятие.

Ученые выяснили, что для успеха операции:

1. сначала надо заранее заготовить некоторое количество порций корма  $m$ , после чего отдать одну порцию этого корма в лабораторию на анализ;
2. затем **поровну** распределить оставшиеся  $m - 1$  порций корма между пауками (каждый паук при этом должен получить целое число порций корма);
3. в конце корма не должно остаться, так как его утилизация привлечет ненужное внимание.

При этом, чтобы максимизировать комфорт пауков, вы хотите выделить каждому пауку как можно больше порций корма. Но из-за ограниченного бюджета не представляется возможным заготовить больше  $n$  порций.

Ловля пауков не всегда проходит удачно, поэтому вам известно только что после сегодняшней операции в лаборатории появится от 1 до  $k$  пауков. Определите максимальное количество корма, не превышающее  $n$ , чтобы при любом количестве пойманных сегодня пауков, вы могли отдать одну порцию в лабораторию, а оставшиеся распределить между пауками поровну. Разрешается не кормить пауков вообще (но только если не кормить ни одного из них).

### Формат входных данных

Единственная строка ввода содержит два целых числа  $n$  и  $k$  — максимальное количество корма и пойманных пауков, соответственно ( $1 \leq n, k \leq 10^{18}$ ).

### Формат выходных данных

Выведите единственное число  $m$  — наибольшее количество корма, которое вы можете приготовить, чтобы выполнить все условия.

### Примеры

стандартный ввод	стандартный вывод
5 2	5
10 3	7

## Задача С. Прыжки между вселенными

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

Для путешествий между вселенными в *Паучьих мирах* существуют специальные порталы. Сеть из  $m$  порталов для  $n$  вселенных представляет из себя граф из  $n$  вершин и  $m$  ребер. Каждый портал связывает какие-то две вселенные, причем никакие два портала не связывают одну и ту же пару вселенных.

Для перемещения Мигель О'Хара использует свои высокотехнологичные порталные часы. Но самые первые версии этих часов были не настолько развиты, и не позволяли сразу путешествовать в произвольные вселенные. А именно, у часов был параметр *энергетического уровня*, который изначально был равен 0.

Каждый портал характеризуется некоторой характеристикой  $w$  — минимальным значением энергетического уровня, необходимым для использования этого портала. Если значение энергетического уровня часов меньше  $w$ , то воспользоваться этим порталом сейчас нельзя. К счастью, есть способ увеличить энергетический уровень: **при первом попадании** во вселенную  $i$  энергетический уровень часов **навсегда** увеличивается на  $a_i$ .

Мигель задумался, какое максимальное значение энергетического уровня он может получить на старом прототипе часов, если посетит все вселенные, в которые сможет попасть, начав свой путь из вселенной номер  $s$ ? Помогите ему ответить на этот вопрос.

### Формат входных данных

В первой строке входных данных даны три целых числа  $n$ ,  $m$  и  $s$  — количество вселенных, количество порталов между этими вселенными, и номер вселенной, из которой Мигель начинает свои перемещения ( $1 \leq s \leq n \leq 10^5$ ;  $1 \leq m \leq 2 \cdot 10^5$ ).

Во второй строке входных данных через пробел перечислены  $n$  целых чисел  $a_i$  — значения, на которые увеличивается энергетический уровень часов при первом посещении каждой вселенной ( $1 \leq a_i \leq 10^9$ ).

В следующих  $m$  строках дано описание порталов. Описание портала под номером  $i$  содержит три целых числа  $u_i$ ,  $v_i$  и  $w_i$ , и определяет портал между вселенными  $u_i$  и  $v_i$ , для использования которого нужно иметь силу часов не меньше  $w_i$  ( $1 \leq u_i, v_i \leq n$ ;  $0 \leq w_i \leq 10^9$ ).

Граф связей между вселенными не содержит кратных ребер и петель, однако не обязательно связан.

### Формат выходных данных

В единственной строке выведите одно целое число — максимальное значения энергетического уровня часов, которое может получить Мигель.

### Примеры

стандартный ввод	стандартный вывод
5 4 1 1 1 1 1 1 1 2 2 1 3 1 1 4 3 1 5 5	4
4 3 1 3 2 1 10 1 2 3 2 3 5 1 3 4	6

## Задача D. Тест на интеллект

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

В «Паучьем сообществе» есть специальная машина, которая отправляет попутанцев из других миров в их родные измерения. Однако далеко не все знают, как она работает: даже ее настройка — это чудовищно сложное занятие.

Для того, чтобы корректно настроить эту машину, сначала надо подтвердить, что у тебя достаточно навыков для выполнения настройки. И для выяснения этого, перед настройкой после каждого перезапуска эта машина предлагает вам пройти небольшой тест на интеллект.

Правила теста очень простые: вам дается целое положительное число  $n$ , после чего вы должны назвать строку  $s$  из строчных латинских букв от 'a' до 'z' длины ровно  $n$ . Как только вы назвали эту строку, машина выполняет следующие действия:

1. Сначала вычисляется количество уникальных букв в строке  $s$ , обозначим его за  $k$ .
2. Затем каждой использованной в  $s$  букве алфавита сопоставляется «вес» — уникальное целое число от 1 до  $k$ . Обозначим этот «вес» буквы  $c$  за  $w(c)$ .
3. После чего вычисляется следующая величина:  $T = \sum_{c \in s} w(c)$ , иными словами — суммируются веса всех букв в названной строке.

Разумеется, машина назначает веса буквам так, чтобы итоговая сумма получилась как можно меньше. Например, если вы назвали строку «aba», то машина назначит вес 1 букве 'a' и вес 2 букве 'b', после чего конечное значение получится равным  $1 + 2 + 1 = 4$ .

По заданному  $n$  назовите любую строку длины  $n$ , которая позволяет получить максимальное возможное значение указанной величины  $T$ .

### Формат входных данных

В единственной строке ввода дано целое число  $n$  — длина строки, которую вам предстоит назвать ( $1 \leq n \leq 10^5$ ).

### Формат выходных данных

Выведите строку из  $n$  букв латинского алфавита от 'a' до 'z', которая позволяет получить максимальное возможное значение  $T$  после оценки машиной. Если возможных вариантов ответа несколько, выведите любой из них.

### Пример

стандартный ввод	стандартный вывод
3	abc

## Задача Е. Прямоугольное Пятно

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

Мало кто знает, но в Лего-вселенной есть свой суперзлодей по кличке Пятно. Только он не настолько известный, потому что он может создавать не красивые овальные «дыры» в ткани реальности, а только скучные прямоугольные.

Тем не менее, он, как и Пятно, которого мы знаем, стремится увеличить свою силу. Для этого он пытается создать как можно больше «дыр», попарно вложенных друг в друга, чтобы затем с помощью получившегося мега-портала впитать в себя энергию сразу большого числа коллайдеров из других вселенных.

Сейчас у него уже есть  $n$  прямоугольных «дыр», расположенных в одной плоскости. Их стороны параллельны осям координат, и эти «дыры» можно

- перемещать и накладывать друг на друга;
- поворачивать на  $90^\circ$  в рамках их общей плоскости (то есть менять местами их высоту и ширину).

Определите, какую максимальную по количеству «дыр» последовательность можно составить, чтобы каждая следующая «дыра» была вложена в предыдущую. Мы считаем, что «дыру» размера  $(h_1, w_1)$  можно вложить в «дыру» размера  $(h_2, w_2)$ , если  $h_1 \leq h_2$  и  $w_1 \leq w_2$ .

### Формат входных данных

В первой строке ввода дано целое число  $n$  — количество прямоугольных «дыр», из которых мы хотим составить последовательность вложенных ( $1 \leq n \leq 10^5$ ).

В  $i$ -й из следующих  $n$  строк через пробел записаны два целых числа  $h_i$  и  $w_i$  — высота и ширина  $i$ -й «дыры» ( $1 \leq h_i, w_i \leq 10^9$ ).

### Формат выходных данных

В первой строке выведите максимальное количество «дыр», которые можно последовательно вложить друг в друга.

Во второй строке выведите через пробел их номера в 1-нумерации, в порядке от самого маленького до самого большого.

Если существует несколько вариантов ответа, выведите любой из них.

### Примеры

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

## Замечание

В примере можно повернуть четвертый прямоугольник на  $90^\circ$ , и получится последовательность прямоугольников с размерами  $(1, 1)$ ,  $(1, 4)$ ,  $(2, 5)$  и  $(3, 5)$ . Нетрудно заметить, что каждый предыдущий можно вложить в следующий, а получить последовательность большей длины с тем же свойством не получится.

## Задача F. Гвен отдыхает

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

Гвен опять путешествует между мирами, но, так как она уже не состоит в команде Мигеля, то она использует порталные часы, полученные от Хоби. При этом ее старые порталные часы заблокированы, но это не значит, что они бесполезны — на них есть забавная игра, за которой можно убить время, когда нечего делать.

Игра начинается с прямоугольника размера  $n \times m$  и проходит следующим образом:

- Первой ходит Гвен. Она может выбрать любую сторону прямоугольника и увеличить ее на 1. После этого действия она получит очки, равные изменению площади прямоугольника.
- После этого по таким же правилам ходит встроенный в часы компьютер.

Всего Гвен и алгоритм в часах совершают по  $k$  ходов, т.е. общее количество ходов равно  $2k$ . И Гвен, и компьютер всегда делают свои ходы оптимально.

Но все же скоро Гвен придется отправиться на поиски Майлза, а оставлять игру недоигранной не хочется. Помогите Гвен и посчитайте, кто победит в такой игре, и на сколько очков больше будет у победителя.

### Формат входных данных

В первой строке ввода даны два целых числа  $n$  и  $m$  — изначальные размеры прямоугольника ( $1 \leq n, m \leq 10^9$ ).

Во второй строке находится единственное число  $k$  — количество доступных ходов у каждой стороны ( $1 \leq k \leq 10^9$ ).

### Формат выходных данных

Если победит Гвен, то выведите число 1; если же больше очков наберет компьютер, то выведите число 2. При ничьей выведите число 0.

После этого через пробел выведите неотрицательное целое число — разницу в очках между победителем и проигравшим.

### Примеры

стандартный ввод	стандартный вывод
3 3 5	2 5
10 3 2	0 0

## Задача G. Человек-паук Нуар и кубик Рубика

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

Немного отвлечемся от второй части истории и вспомним первую. Человек-паук Нуар оказался очень заинтересован кубиком Рубика, и решил во что бы то ни стало разгадать все тайны этого загадочного предмета. Но вот незадача — до сих пор он так и не смог выяснить, что это за странное устройство, и в чем его предназначение.

Поэтому он решил потренироваться на черно-белой версии кубика Рубика, при чем еще и двумерной, а не трехмерной. Для этого он взял клетчатую табличку размера  $n \times m$ , каждая клетка которого закрашена либо в черный, либо в белый цвет. Как и в кубике Рубика, цвета элементов можно менять, но вместо вращения с этой таблицей можно делать одну из двух операций:

- инвертировать все цвета в любом столбце;
- инвертировать все цвета в любой строке.

Конечная цель — добиться того, чтобы существовал «путь» по черным клеткам из левого верхнего угла таблицы в правый нижний с шагами на один вправо и шагами на один вниз. Иными словами, должна существовать такая последовательность черных клеток  $(r_1, c_1), (r_2, c_2), \dots, (r_{n+m-1}, c_{n+m-1})$ , что

- $(r_1, c_1)$  равно  $(1, 1)$ , то есть левой верхней клетке;
- $(r_{n+m-1}, c_{n+m-1})$  равно  $(n, m)$ , то есть правой нижней клетке;
- и для любых двух соседних в этой последовательности клеток  $(r_i, c_i)$  и  $(r_{i+1}, c_{i+1})$  либо  $r_i = r_{i+1}$  и  $c_{i+1} = c_i + 1$ , либо, наоборот,  $c_i = c_{i+1}$  и  $r_{i+1} = r_i + 1$ .

Определите минимальное число действий, необходимое для получения такого пути, или скажите, что получить такой путь из черных клеток невозможно. Человек-паук Нуар очень рассчитывает на вашу помощь.

### Формат входных данных

В первой строке ввода через пробел даны два целых числа  $n$  и  $m$  — высота и ширина таблицы ( $1 \leq n, m \leq 2000$ ).

В  $i$ -й из следующих  $n$  строк записаны  $m$  символов, каждый из которых равен '0' или '1' — обозначения цветов клеток в  $i$ -й строке таблицы. Символ '0' означает белый цвет, а '1' — черный.

### Формат выходных данных

Если невозможно, инвертируя строки и столбцы, получить черный путь из левой верхней клетки таблицы до правой нижней, выведите единственное число  $-1$ .

Иначе, в первой строке выведите через пробел два целых числа  $r$  и  $c$  — количество инвертируемых строк и столбцов, соответственно. После чего во второй строке выведите через пробел  $r$  номеров инвертируемых строк, а в третьей строке —  $c$  номеров инвертируемых столбцов.

Строки и столбцы нумеруются с 1 (строки — сверху вниз, столбцы — слева направо). Выводить номера строк и столбцов можно в любом порядке. Среди всех ответов, минимизирующих  $r + c$ , можно выбрать любой.

## Примеры

стандартный ввод	стандартный вывод
2 2 10 01	1 1 1 1
4 4 1111 0001 0001 0000	1 0 4
3 5 10000 01010 00001	2 1 2 1 1

## Задача Н. Квантовая дыра

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

Квантовые дыры, разрушающие вселенные, иногда можно «залатать», тем самым остановив разрушение мира. Но это удастся далеко не всегда.

Мигель придумал способ оценивать опасность квантовой дыры целым числом. Для этого сначала информация о квантовой дыре представляется в виде битовой строки длины ровно  $n$ . Каждая ее подстрока  $t$  (последовательность подряд идущих бит) длины  $k$  вносит свой независимый вклад  $\text{danger}(t) = d_t$  в суммарную опасность дыры, а именно:

$$\text{danger}(s) = \sum_{i=1}^{n-k+1} \text{danger}(s_{i,\dots,i+k-1}),$$

где  $s_{i,\dots,i+k-1}$  означает подстроку  $s$  длины  $k$  с началом в  $i$ -м бите.

Значения опасности всех двоичных строк длины  $k$  уже изучены и известны, всего таких значений  $2^k$ . Например, для  $k = 2$  вам известны 4 значения:  $d_{00}$ ,  $d_{01}$ ,  $d_{10}$ ,  $d_{11}$ .

Сегодня Марго Кесс от скуки пришла в голову идея найти описание самой безопасной квантовой дыры. Найдите битовую строку длины ровно  $n$ , опасность которой минимальна.

### Формат входных данных

В первой строке входных данных даны два числа  $n$  и  $k$  — длина искомой строки и длина подстрок, соответственно ( $1 \leq n \leq 1000$ ,  $1 \leq k \leq 10$ ).

Во второй строке через пробел перечислены  $2^k$  чисел:  $d_{00\dots 00}$ ,  $d_{00\dots 01}$ ,  $d_{00\dots 10}$ ,  $\dots$ ,  $d_{11\dots 10}$  и  $d_{11\dots 11}$  — уровни опасности всех возможных битовых строк длины  $k$  в лексикографическом порядке ( $1 \leq d_t \leq 1000$ ).

### Формат выходных данных

В качестве ответа выведите наименее опасную строку длины  $n$ . Если таких строк несколько, вы можете вывести любую из них.

### Примеры

стандартный ввод	стандартный вывод
7 2 4 2 1 3	1010101
5 3 8 5 4 6 3 5 6 7	01001
5 3 486 750 753 40 798 644 599 56	01111
5 2 358 906 7 859	10000

### Замечание

В первом примере опасность получившейся строки равна  $d_{10} + d_{01} + d_{10} + d_{01} + d_{10} + d_{01}$ , то есть  $3 \cdot (d_{10} + d_{01}) = 3 \cdot (1 + 2) = 9$ . Обратите внимание, что это не единственный возможный ответ с таким уровнем опасности.

Во втором примере стоимость получившейся строки равна  $d_{010} + d_{100} + d_{001} = 4 + 3 + 5 = 12$ .

В четвертом примере стоимость получившейся строки равна  $c_{10} + 3 \cdot c_{00} = 1081$ .

## Задача I. Стабилизация мультивселенной

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

Аномалии очень опасны тем, что могут изменять естественный ход событий, и даже вмешиваться в *ключевые события*, что может повлечь за собой необратимые последствия для многих вселенных. После того, как Пятно появился на *Земле-2099*, он нарушил столько ключевых событий, что аж  $n$  соседних вселенных под угрозой: паутина мультивселенной расплетается. И вам предстоит это исправить!

Для того, чтобы стабилизировать вселенные, в которых открылись квантовые дыры, необходимо заново связать их вместе, пустив между ними потоки космической энергии. Известно, что:

1. некоторые пары вселенных связаны **односторонними** энергетическими каналами;
2. из каждой из  $n$  вселенных выходят ровно два канала;
3. в каждую из  $n$  вселенных ведут ровно два канала;
4. каждый канал характеризуется парой чисел  $a$  и  $b$ , которые означают, что мощность энергии, пущенной по этому каналу, должна лежать между  $a$  и  $b$  включительно.

Вам необходимо выбрать ровно  $n$  каналов так, чтобы они образовывали циркуляцию энергии, затрагивающую каждую из  $n$  вселенных. Иными словами, эти  $n$  каналов должны образовывать несколько (один или более) циклов, проходящих в совокупности по всем  $n$  вселенным.

При этом мощность энергии, пускаемой по каждому выбранному каналу, должна быть одинакова (даже если каналы находятся на разных циклах). Более формально, должно существовать такое значение мощности  $w$ , что на каждом выбранном канале  $a_i \leq w \leq b_i$ .

Определите, можно ли выбрать такие каналы, и такое значение мощности, чтобы все требуемые условия были выполнены. Спасите мультивселенную!

### Формат входных данных

В первой строке ввода дано единственное целое число  $n$  — количество рассматриваемых вселенных ( $3 \leq n \leq 10^5$ ).

В следующих  $2n$  строках перечислены энергетические каналы: сначала даны два канала, исходящих из вселенной номер 1, затем два канала, исходящих из вселенной номер 2, и так далее. Описание каждого канала задается тремя целыми числами  $t_i$ ,  $a_i$  и  $b_i$  — номером вселенной, в которую он ведет, и ограничениями на мощность пускаемой по нему энергии ( $1 \leq t_i \leq n$ ;  $1 \leq a_i \leq b_i \leq 10^5$ ).

Гарантируется, что в каждую вселенную ведет ровно два канала. Никакой канал не соединяет вселенную с самой собой, но может быть два канала между одними и теми же вселенными.

### Формат выходных данных

В первой строке выведите единственное число  $-1$ , если способа выбрать циркуляцию одной мощности не существует.

Иначе — выведите в первой строке искомое значение мощности энергии  $w$ . После чего во второй строке выведите ровно  $n$  целых чисел через пробел — номера выбранных каналов (от 1 до  $2n$ ) в порядке их следования во вводе.

Если подходящих ответов несколько, выведите любой из них.

## Примеры

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

## Замечание

В первом примере выбираются каналы под номерам 1, 3 и 5. Первый ведет из вселенной номер 1 во вселенную номер 2 и имеет ограничения на энергию [1, 3]. Аналогично, третий идет из вселенной 2 во вселенную 3 с ограничениями [2, 4], и пятый ведет из вселенной 3 во вселенную 1 с ограничениями [3, 5]. Получаем цикл  $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$ , проходящий по всем вселенным, по каждому каналу которого можно пустить энергию с мощностью 3.

Во втором примере получается два цикла  $1 \rightarrow 2 \rightarrow 5 \rightarrow 1$  и  $3 \rightarrow 4 \rightarrow 3$ . По каждому из выбранных каналов можно пустить энергию мощности 6.

## Задача J. Паутина во все стороны

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

Несколько Пауков пытаются остановить обрушение «Alchemax», но Пятно им мешает. Действительно, чем больше хаоса, тем проще ему будет скрыться, ведь Гвен, Майлз, Хоби и Павитр будут заняты спасением окружающих людей.

Чтобы представить себе ситуацию получше, рассмотрим вид сверху, и скажем, что Пауки располагаются на плоскости в точке с координатами  $(0, 0)$  — прямо в сердце «Alchemax», а стены здания «Alchemax» представляют из себя окружность радиуса  $10^6$  с центром в той же точке  $(0, 0)$ .

Для того, чтобы стены «Alchemax» после очередного запуска коллайдера не обрушились наружу, Пауки используют для ее сдерживания новый вид паутины: непрерывную паутину. Она распространяется из точки  $(0, 0)$  как свет — в каждом направлении на произвольно далекое расстояние. И чтобы помешать им, Пятно расположил  $n$  своих «дыр в пространстве» вокруг. Каждая дыра — это «абсолютно черный» круглый объект, то есть каждая такая дыра полностью поглощает все, что на нее попадает.

Другими словами, из точки  $(0, 0)$  во все стороны выходят лучи паутины, но если на пути луча встречается дыра в пространстве, дальше нее луч не пойдет. Ваша задача — определить, до какой части окружности стен «Alchemax» паутина все же дотянется.

### Формат входных данных

В первой строке дано целое число  $n$  — количество дыр в пространстве вокруг команды Пауков ( $1 \leq n \leq 10^5$ ).

В следующих  $n$  строках дано их описание. Каждая дыра представляется кругом на плоскости и задается тремя целыми числами  $x_i$ ,  $y_i$  и  $r_i$  — координатами его центра и радиусом ( $1 \leq |x_i|, |y_i|, r_i \leq 10^5$ ). Дыры могут накладываться друг на друга.

### Формат выходных данных

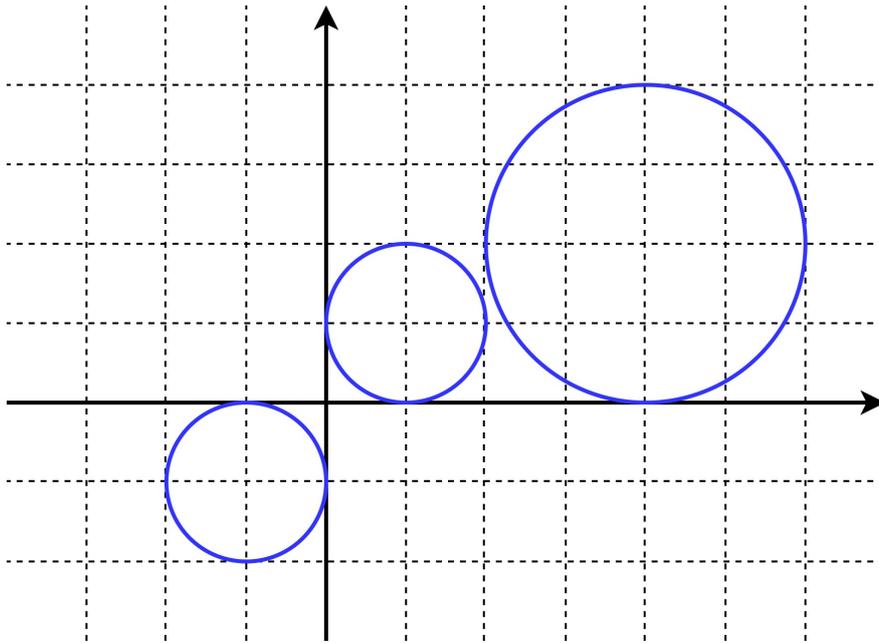
Выведите одно дробное число от 0 до 1 — долю большой окружности, до которой дотягивается паутина, не встретив по пути дыры. Ваш ответ будет засчитан при погрешности не более  $10^{-4}$ .

### Примеры

стандартный ввод	стандартный вывод
3 1 1 1 4 2 2 -1 -1 1	0.5000000
2 4 0 1 0 3 1	0.8113959

### Замечание

Иллюстрация для первого теста в примере. Лучи распространяются на большое расстояние только в левую верхнюю и правую нижнюю четверти, то есть ровно на половину далекой окружности.



## Задача К. Иерархия Паучьего сообщества

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

В «Паучьем сообществе» есть понятная иерархия того, кто кому подчиняется. Разумеется, это не означает, что кто-то из Людей-Пауков менее или более важен, чем другие, но во время проведения операций по спасению мультивселенной важно, чтобы кто-то отвечал за координацию участников операции.

Всего в сообществе  $n$  Пауков, и его иерархия представляет собой подвешенное дерево. Корень этого дерева — паук под номером 1, Мигель О'Хара. Сама иерархия задается  $n - 1$  связями непосредственного подчинения между членами сообщества. Будем говорить, что Паук номер  $u$  — «ментор» Паука  $v$ , если между ними есть связь непосредственного подчинения, и  $u$  располагается в иерархии ближе к Мигелю, чем  $v$ . В таком случае будем называть  $v$  «подчиненным»  $u$ .

После появления Майлза в сообществе, и его побега, члены сообщества разделились на две группы, у каждой из которых свое понимание того, как им следует действовать дальше. Назовем эти два имеющихся мнения А и В. *Беспорядком* в сообществе назовем количество таких пар  $u$  и  $v$ , что Паук номер  $u$  — ментор Паука номер  $v$  (а  $v$ , соответственно — подчиненный  $u$ ), и при этом у Паука номер  $u$  мнение А, а у паука номер  $v$  мнение В.

Мигель очень хочет знать, кто из его союзников придерживается какого мнения, но у него нет времени на проведение опроса. Помогите ему определить возможное распределение мнений в сообществе, если известно, что на данный момент **беспорядок в сообществе максимальный возможный**.

### Формат входных данных

В первой строке ввода дано единственное целое число  $n$  — количество Людей-Пауков в «Паучьем сообществе» ( $1 \leq n \leq 10^5$ ).

В следующих  $n - 1$  строках дано по два целых числа  $a_i$  и  $b_i$ , означающих, что между Пауками  $a_i$  и  $b_i$  есть связь непосредственного подчинения ( $1 \leq a_i, b_i \leq n$ ). При этом не обязательно, что  $a_i$  — ментор  $b_i$ , может быть и наоборот.

Гарантируется, что набор связей образует дерево.

### Формат выходных данных

В первой строке выведите через пробел два целых числа — максимальное возможное значение беспорядка  $d$  в такой иерархии, и  $k$  — количество Людей-Пауков, придерживающихся мнения А.

Во второй строке через пробел перечислите  $k$  различных чисел от 1 до  $n$  — номера Пауков, придерживающихся мнения А.

Если существует несколько возможных ответов с максимальным значением  $d$ , выведите любой из них.

## Примеры

стандартный ввод	стандартный вывод
3 1 2 2 3	1 1 2
4 1 2 1 3 1 4	3 1 1
8 1 2 1 3 2 8 3 4 3 5 3 6 5 7	4 2 2 3

## Задача L. Взломать коллайдер

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

*Это интерактивная задача.*

Майлз продолжает бороться с «Alchemax». Его задача в этот раз — взломать настройки коллайдера, чтобы остановить его повторный запуск.

Всего у коллайдера есть  $n$  настроек  $a_i$  с номерами от 1 до  $n$ , каждая из них является целым положительным числом. Компания предусмотрела защиту настроек от посторонних, поэтому узнать значение определенной настройки напрямую нельзя. Однако любой человек может запросить некоторое  $f(x)$ , равное значению настройки с номером, циклически сдвинутому от  $x$  на  $c$  вперед ( $0 \leq c \leq n-1$ ), то есть с номером  $x+c$ , если  $x+c \leq n$ , и  $x+c-n+1$  иначе.

Майлз выяснил, что на самом деле настройки идут по возрастанию, то есть что  $a_{i+1} > a_i$  для всех  $i$  от 1 до  $n-1$ . Теперь для остановки коллайдера Майлзу обязательно нужно найти известное только сотрудникам «Alchemax» число  $c$  — сдвиг в номерах, используемый в системе доступа к настройкам.

Узнайте значение  $c$ , если вам дано число  $n$  и возможность запрашивать значения  $f(x)$  не больше 42 раз.

### Формат входных данных

В первой строке задано одно целое число  $n$  — количество настроек ( $1 \leq n \leq 10^5$ ).

Также гарантируется, что все  $a_i$  — целые числа от 1 до  $10^9$  включительно.

### Протокол взаимодействия

Взаимодействие с интерактором проходит в виде запросов со стороны вашей программы и ответов со стороны интерактора.

Вы можете не более 42 раз запросить значение  $f(x)$  для  $1 \leq x \leq n$ . Чтобы запросить очередное значение, выведите на отдельной строке «?  $x$ », где вместо  $x$  укажите нужное число. Если вы не превысили лимит в 42 запроса вида «?», в следующей строке интерактор выведет единственное целое число  $f(x)$ , равное  $a_{x+c}$  при  $x+c \leq n$  или  $a_{x+c-n+1}$  иначе.

Для того, чтобы вывести ответ на задачу, напечатайте «!  $c$ », где вместо  $c$  должно быть значение сдвига ( $0 \leq c \leq n-1$ ). Вывод ответа не учитывается в количестве запросов. После вывода ответа ваша программа должна завершиться с нулевым кодом возврата.

Если в какой-то момент ваша программа превышает лимит в 42 запроса, интерактор выведет на следующей строке  $-1$  и завершится с вердиктом WA (Wrong Answer). Во избежание получения вердиктов RE, TL или IL, прочитав ответ  $-1$ , ваша программа должна завершиться с нулевым кодом возврата.

Не забывайте после каждого запроса сбрасывать буфер вывода, чтобы интерактор получил ваш запрос. Это можно сделать с помощью `std::cout.flush()` в C++, `System.out.flush()` в Java и `sys.stdout.flush()` в Python, а также аналогичными командами в других языках. Если ваша программа не сбрасывает буфер вывода, она получит вердикт TL или IL.

## Примеры

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