

Задача А. Магнит VK

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

Школьник Василий, как и многие его ровесники, любит проводить время в социальной сети ВКонтакте. Долгое время он пытался раздобыть магнетик VK себе на холодильник. Но нигде такого не находил. И вот однажды в гостях у двоюродного брата он увидел на холодильнике слово, составленное из букв английского алфавита. Каждая буква была магнитом и все они были синего цвета, поэтому Василию пришла в голову идея составить из этих букв надпись VK для своего холодильника. Тем более что брату эти буквы были не нужны.

Помогите Василию разобраться, может ли он найти в слове две заглавные буквы V и K, чтобы забрать их себе на холодильник. Если таких букв нет, то проверьте найдутся ли две строчные буквы v и k в слове (в крайнем случае Василий готов составить надпись vk из двух строчных букв).

Формат входных данных

На вход подается единственная строка, состоящая из строчных и заглавных букв английского алфавита. Длина строки может быть от 1 до 100 символов.

Формат выходных данных

Если в заданной строке можно найти две заглавные буквы V и K, то в единственной строке выведите слово «VK» (без кавычек).

Если в заданной строке не удалось найти две заглавные буквы V и K, но можно найти две строчные буквы v и k, то в единственной строке выведите слово «vk» (без кавычек).

В противном случае в единственной строке выведите грустный смайлик «:(» (без кавычек).

Примеры

стандартный ввод	стандартный вывод
viking	vk
KILOVOLT	VK
Kvass	:(

Задача В. Робин Гуд и массив

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

Однажды Робин Гуд нашёл массив из n целых чисел. Он непременно решил *уравнять* этот массив, то есть сделать все элементы равными. Для этого он может выполнить над массивом несколько (возможно ноль) операций.

За одну операцию Робин Гуд может выбрать **три** подряд идущих элемента массива и заменить их на медиану этих трёх элементов.

Медианой набора чисел $a_1, a_2, \dots, a_{2m-1}$ называется число c_m , где $c_1, c_2, \dots, c_{2m-1}$ — набор чисел $a_1, a_2, \dots, a_{2m-1}$, отсортированный по **неубыванию**. Например, медиана набора чисел 5, 2, 3 равна 3, а медиана набора чисел 4, 8, 4 равна 4.

Можно показать, что такими операциями **всегда** можно уравнять массив длины хотя бы 3. Например, массив $[2, 4, 5, 5, 9]$ можно уравнять так: $[2, 4, 5, 5, 9] \rightarrow [2, 5, 5, 5, 9] \rightarrow [5, 5, 5, 5, 9] \rightarrow [5, 5, 5, 5, 5]$.

Робин Гуд хочет узнать **максимальное** число, к которому можно приравнять **все** элементы массива с помощью нескольких применений вышеописанной операции.

Формат входных данных

В первой строке дано целое число n ($3 \leq n \leq 2 \cdot 10^5$) — количество элементов в массиве.

Во второй строке даны n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — элементы массива.

Формат выходных данных

Выведите единственное целое число — ответ на задачу.

Примеры

стандартный ввод	стандартный вывод
5 1 2 4 4 2	4
6 4 7 2 1 9 18	9
3 2 1000000000 2	2

Задача С. Армагеддон

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

Магнус Карлсен и Хикару Накамура вышли в финал чемпионата мира по быстрым шахматам. После серии ничейных партий в тай-брейке им предстоит сыграть партию-армагеддон. Правила этой партии аналогичны обычным, за исключением одного: при ничейном исходе партии победителем объявляется шахматист, игравший черными.

У Хикару и Магнуса есть стратегия, как быстро получить ничью за черных, но для этого им нужно, чтобы изначально на часах у них было не меньше k минут и s секунд. Если времени на часах будет меньше, то шахматист не успеет сделать ничью и проиграет.

Количество времени на часах у черных и кто будет играть за черных определяется следующим образом:

- Магнус и Хикару загадывают определенное количество времени.
- Играть за черных будет тот, кто предложил наименьшую ставку, при этом на часах у него будет время равное наибольшей ставки.
- При равенстве ставок, шахматист, который играет за черных, выбирается случайным образом.

Найдите оптимальную ставку для Хикару, чтобы с максимальной вероятностью выиграть.

Формат входных данных

В первой строке задаются значения k_1 и s_1 ($1 \leq k_1 \leq 10$, $0 \leq s_1 < 60$) — время, которое нужно Хикару, чтобы сделать ничью за черных.

Во второй строке задаются значения k_2 и s_2 ($1 \leq k_2 \leq 10$, $0 \leq s_2 < 60$) — время, которое нужно Магнусу, чтобы сделать ничью за черных.

Формат выходных данных

Если у Магнуса есть стратегия, которая позволяет ему гарантированно выиграть, выведите -1. Иначе выведите ставку Хикару (количество минут и секунд), которая максимизирует его шансы стать чемпионом. Если таких ставок несколько, выведите минимальную

Пример

стандартный ввод	стандартный вывод
7 59 8 0	7 59

Задача D. Охота на монстра

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

Вы счастливый мэр процветающей столицы Берляндии, но вот незадача — в пригороде поселился ужасный монстр, который атакует торговые караваны, курсирующие между столицей и соседними городами. Чтобы избавиться от монстра, вы решили нанять команду героев в городской гильдии авантюристов.

Гильдия предлагает n воинов и у i -го воина h_i единиц здоровья и d_i единиц атаки. У монстра, которого надо убить, a единиц здоровья.

Пока у монстра осталось положительное число единиц здоровья и остался хотя бы один герой с положительным количеством единиц здоровья, происходит следующее:

1. Сначала каждый живой герой наносит монстру урон, равный значению его атаки. Если здоровье монстра упало до нуля, битва завершается;
2. Затем, если монстр еще жив, он наносит каждому герою по единице урона. Каждый герой, чей уровень здоровья упал до нуля, умирает.

Так как казна ограничена, вы хотите нанять наименьшее число героев, чтобы убить монстра. Но вы также не хотите, чтобы умер **хотя бы один** из героев команды, поэтому вам надо нанять наименьшее число героев, чтобы они могли убить монстра без потерь.

Формат входных данных

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

Далее следует описание наборов.

В первой строке каждого набора содержится целое число a ($1 \leq a \leq 10^9$) — здоровье монстра.

Во второй строке каждого набора содержится целое число n ($1 \leq n \leq 2 \cdot 10^5$) — общее число воинов, которых предлагает гильдия для найма.

В третьей строке каждого набора содержатся n целых чисел h_1, h_2, \dots, h_n ($1 \leq h_i \leq 10^9$) — единицы здоровья каждого героя.

В четвертой строке каждого набора содержатся n целых чисел d_1, d_2, \dots, d_n ($1 \leq d_i \leq 10^9$) — единицы атаки каждого героя.

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

Формат выходных данных

Для каждого набора входных данных в отдельной строке выведите единственное целое число:

- минимальное количество героев, которое нужно нанять, чтобы убить монстра без потерь,
- или -1 , если это невозможно.

Пример

стандартный ввод	стандартный вывод
2	3
12	1
5	
1 2 3 3 4	
2 4 1 1 1	
10	
1	
1	
14	

Замечание

В первом наборе входных данных достаточно трёх героев, чтобы убить монстра без потерь. Например, это могут быть герои с номерами 2, 3, 4. Их суммарная атака равна 6 и они убивают монстра за две атаки.

Во втором наборе входных данных первого теста есть единственный герой и его силы достаточно, чтобы сразу убить монстра.

Задача Е. Наибольшее общее простое

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

Наверное, вам хорошо известны понятия наибольшего общего делителя и наименьшего общего кратного. В этой задаче мы не будем просить вас их искать.

Вместо этого рассмотрим **наибольшее общее простое** двух натуральных чисел $\text{гср}(a, b)$ — это такое наибольшее простое число, на которое одновременно делится и a , и b .

Заметим, что не для любой пары чисел определено их наибольшее общее простое. Например, $\text{гср}(10, 15) = 5$, а $\text{гср}(2, 3)$ не определено.

Вам предстоит ответить на t запросов, содержащих пару чисел l и r . Для каждого запроса вам нужно:

- найти **наибольшее простое** число p такое, что существуют два числа a и b ($a \neq b$) на отрезке $[l; r]$ такие, что $\text{гср}(a, b) = p$,
- или сообщить, что такого простого числа p не существует.

Например, пусть заданы $l = 3$, $r = 10$. Тогда $p = 5$, так как $\text{гср}(5, 10) = 5$.

Формат входных данных

В первой строке входных данных содержится одно целое число t ($1 \leq t \leq 1000$).

Далее следует t строк, в каждой из которых содержатся два числа l и r ($1 \leq l \leq r \leq 10^9$) — отрезок запроса.

Формат выходных данных

Для каждого запроса в отдельной строке выведите ответ:

- наибольшее простое число p , для которого на отрезке $[l, r]$ существуют два разных числа a и b таких, что $\text{гср}(a, b) = p$,
- или -1 , если такого числа не существует.

Пример

стандартный ввод	стандартный вывод
4	47
1 100	50231
5000 100500	166666649
800000000 1000000000	-1
79 81	

Замечание

В примере $p = 47$, так как на отрезке $[1, 100]$ существуют числа $a = 47$, $b = 94$ такие, что $\text{гср}(a, b) = \text{гср}(47, 94) = 47$.

Задача F. Расстояние между точками

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1.5 секунд
Ограничение по памяти: 256 мегабайт

На плоскости есть два множества точек с целочисленными координатами, размера n и m соответственно. Требуется взять одну точку из первого множества и одну точку из второго, так, чтобы евклидово расстояние между ними было минимально.

Формат входных данных

В первой строке находятся два целых числа n и m ($1 \leq n, m \leq 10^5$) — размеры множеств.

В следующих n строках находятся координаты x_i, y_i ($-10^8 \leq x_i, y_i \leq 10^8, 1 \leq i \leq n$) точек первого множества, по одной точке в каждой строке.

В следующих m строках находятся координаты x_j, y_j ($-10^8 \leq x_j, y_j \leq 10^8, 1 \leq j \leq m$) точек второго множества, по одной точке в каждой строке.

Формат выходных данных

Требуется вывести одно целое число — минимальное расстояние между точками, **возведённое в квадрат**.

Пример

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

Замечание

В первом тесте ответ 4 достигается, если выбрать точки $(0, 0)$ и $(2, 0)$.

Задача A. Вердикты системы All Cups

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

В этой задаче речь пойдет про одного из участников первого отбора олимпиады «Когнитивные технологии» 2023-2024 года. Для анонимности будем звать этого участника именем Ратибор.

Так вот, Ратибор узнал, что в системе All Cups, на которой проходит олимпиада, есть шесть основных вердиктов, которые можно получить за посылку по задаче: OK, WA, PE, RE, ML и TL. Есть еще несколько вспомогательных вердиктов, но они не заинтересовали Ратибора.

Некоторые из вердиктов Ратибор уже получил на первом отборе, и ему захотелось собрать коллекцию из всех шести основных вердиктов. Поэтому он принял решение о том, что на втором отборе получит все недостающие вердикты. Конечно, в первую очередь Ратибор постарается занять первое место на втором отборе, но собрать коллекцию тоже очень важно для него.

Помогите Ратибору определить, каких вердиктов ему не хватает для коллекции.

Формат входных данных

В первой строке дано целое число n ($1 \leq n \leq 5$) — количество различных вердиктов, которые Ратибор получил в системе All Cups на первом отборе.

В следующих n строках даны вердикты, которые получил Ратибор, по одному в строке.

Формат выходных данных

Выведите все вердикты, которых не хватает Ратибору для полной коллекции, по одному в строке. Вердикты можно выводить в любом порядке.

Пример

стандартный ввод	стандартный вывод
З	RE
OK	ML
WA	TL
PE	

Задача В. N-факториал

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

N -факториалом называется число вида $1 \cdot 2^2 \cdot 3^3 \cdot \dots \cdot N^N$. Другими словами, это произведение всех чисел от 1 до N , где каждый множитель возведен в степень, равную самому себе. Будем считать, что N -факториал определен для $N \geq 0$, а 0-факториал равен 1.

Вам дано целое положительное число x . Определите, является ли оно суммой двух N -факториалов?

Например, если $x = 112$, то оно является суммой 3-факториала и 2-факториала, так как $112 = 108 + 4$, и при этом $108 = 1^1 \cdot 2^2 \cdot 3^3$, а $4 = 1^1 \cdot 2^2$

Формат входных данных

Первая строка входных данных содержит единственное целое число x ($1 \leq x \leq 10^9$) — число, для которого необходимо сказать, является ли оно суммой двух N -факториалов.

Формат выходных данных

Выведите:

- «YES», если число x является суммой двух N -факториалов;
- «NO» в противном случае.

Вы можете выводить ответ в любом регистре (например, строки «yEs», «yes», «Yes» и «YES» будут распознаны как положительный ответ).

Примеры

стандартный ввод	стандартный вывод
112	YES
25	NO
8	YES
86400108	YES

Замечание

Первый тест разобран в условии задачи.

В третьем тесте число является суммой двух 2-факториалов.

Задача С. Граница

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

В стране Берляндии есть n городов и m двусторонних дорог между ними. Все города пронумерованы от 1 до n , и из любого города страны возможно добраться до любого другого. Между любой парой городов есть не более одной дороги, при этом каждая дорога имеет целую положительную длину.

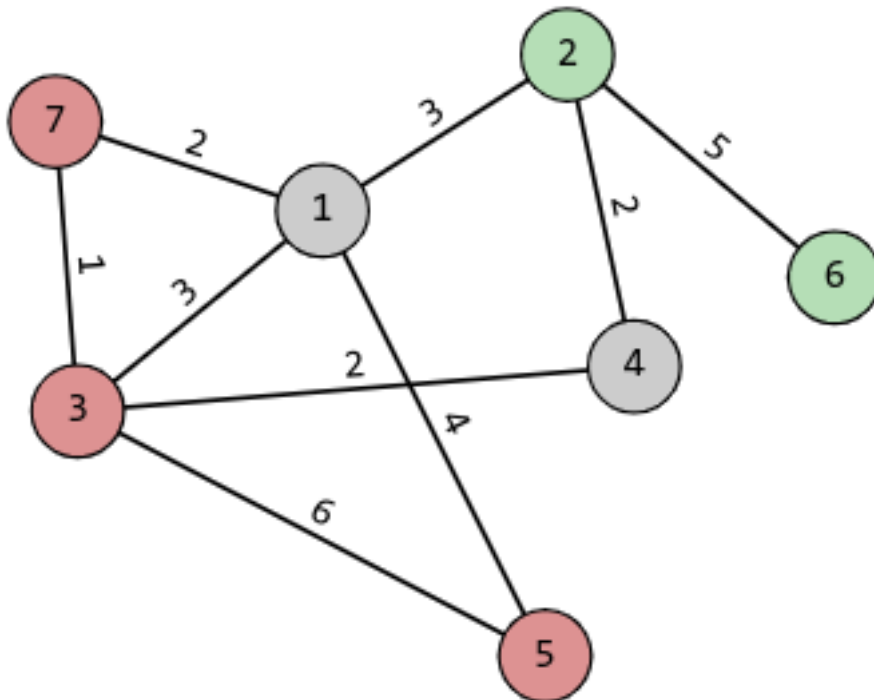
Внутри страны находятся 2 княжества, которые хотят разделить территорию между собой. Столица первого княжества находится в городе с номером a , а столица второго — в городе с номером b .

Княжества не очень ладят между собой, поэтому также хотят, чтобы между ними обязательно была граница, состоящая из непустого множества городов, не принадлежащих ни одному из княжеств.

Определите, можно ли разделить территорию так, чтобы **одновременно** выполнялись 4 следующих условия:

- если расстояние от некоторого города x до a и b одинаково, то он находится на границе и не принадлежит никакому из княжеств;
- если расстояние от некоторого города x до a меньше, чем до b , он принадлежит 1-му княжеству;
- если первые 2 условия не верны, город x принадлежит 2-му княжеству;
- нельзя пройти из одного княжества в другое, не пересекая граничный город;

Например, пусть $n = 7$, $m = 9$, столица первого княжества находится в городе 2, а второго — в городе 3. Карта Берляндии выглядит как на рисунке ниже:



Все города, которые будут принадлежать первому княжеству, покрашены в красный цвет, второму — в зеленый. Города, которые будут находиться на границе, покрашены в серый.

Формат входных данных

Первая строка входных данных содержит два целых числа n, m ($3 \leq n \leq 500$, $n - 1 \leq m \leq \frac{n \cdot (n-1)}{2}$) — количество городов в Берляндии и количество дорог между ними.

В следующих m строках находится по три целых положительных числа u, v, c ($1 \leq u, v \leq n$, $u \neq v$, $1 \leq c \leq 10^5$) — пара городов, соединенных дорогой и длина этой дороги.

В последней строке входных данных находятся два целых числа a, b ($1 \leq a, b \leq n$, $a \neq b$) — города, в которых находятся столицы первого и второго княжества.

Формат выходных данных

Если разделить города между княжествами так, чтобы выполнялись все условия, невозможно, выведите число -1 ;

Иначе на одной строке выведите ровно n чисел, где i -е число ($1 \leq i \leq n$) будет равно:

- 1, если город с номером i будет принадлежать 1-му княжеству;
- 2, если город с номером i будет принадлежать 2-му княжеству;
- 0, если город с номером i будет находиться на границе.

Примеры

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

Замечание

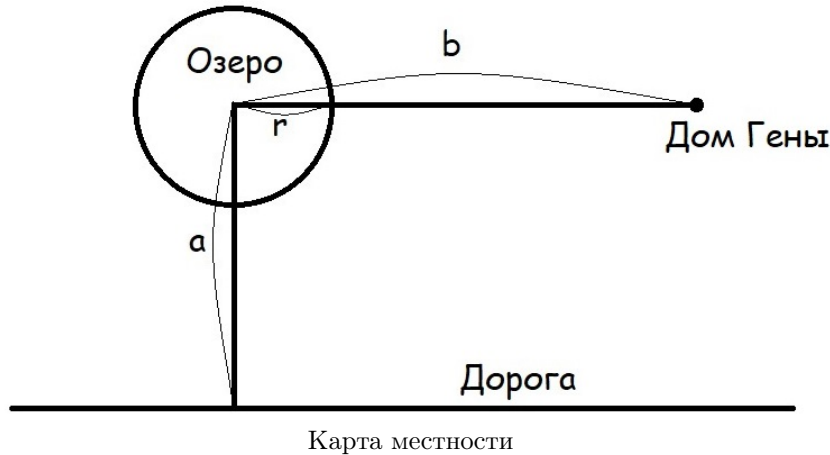
Первый тест разобран в условии задачи.

Во втором тесте нельзя построить границу между княжествами, так как нет непустого множества городов, имеющих одинаковые расстояния до a и b .

Задача D. Поход и озеро

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

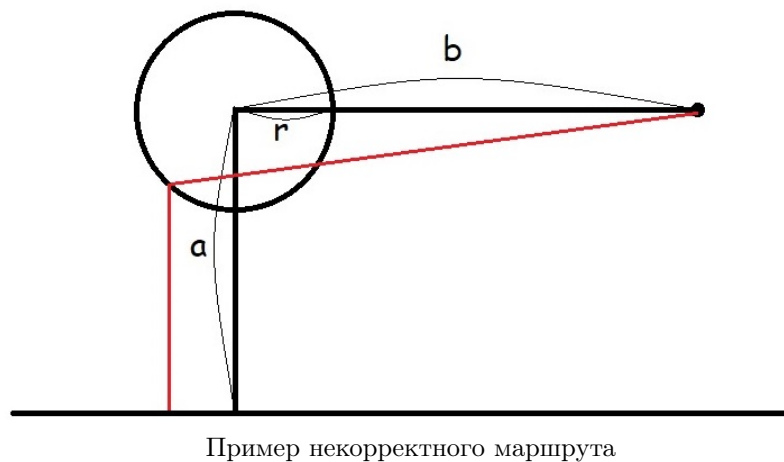
Мальчик Гена решил на выходных устроить поход. Дом Гены расположен в поле. Рядом с домом находится круглое озеро. Радиус окружности озера равен r . Расстояние от центра озера до дома Гены равно b , где $b > r$. Через поле проходит дорога на расстоянии a от центра озера, где $a > r$. Если через центр озера и дом Гены провести прямую, то окажется, что она параллельна дороге. Схема расположения объектов изображена на рисунке 1.



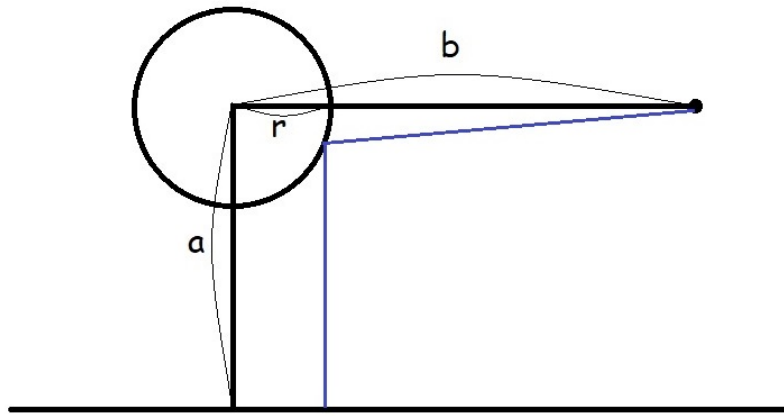
Гена хочет организовать поход следующим образом. Сначала родители высадят его в какой-то точке дороги. От этой точки Гена начнет свой путь. Сперва он пойдет перпендикулярно дороге, пока не дойдет до озера. Затем, от озера до дома, Гена хочет пойти по прямой, которая проходит только по полю и не пересекает озеро. Понятно, что не всякая точка на дороге может быть началом маршрута, но Гена выберет правильную точку для старта, ведь он опытный турист. При этом Гена выберет для старта такую точку, чтобы его маршрут имел наименьшую длину.

Найдите длину маршрута Гены.

На рисунке 2 красным цветом изображен некорректный маршрут. Этот маршрут проходит через озеро, а Гена хочет идти всегда по полю и только в одной точке маршрута побывать возле озера.



На рисунке 3 синим цветом изображен корректный маршрут — сначала он идет перпендикулярно дороге, пока не упрется в озеро, а затем от озера идет по прямой к дому, не проходя при этом через озеро. Это пример одного из возможных корректных маршрутов. Не утверждается, что это самый короткий корректный маршрут.



Пример корректного маршрута

Формат входных данных

В первой строке дано одно целое число r ($1 \leq r \leq 10$) — радиус озера.

Во второй строке дано одно целое число b ($r < b \leq 100$) — расстояние от центра озера до дома Гены.

В третьей строке дано одно целое число a ($r < a \leq 100$) — расстояние от центра озера до дороги.

Формат выходных данных

В единственной строке выведите одно вещественное число — длину маршрута Гены.

Ответ считается правильным, если его абсолютная или относительная погрешность не превышает 10^{-6} .

Пример

стандартный ввод	стандартный вывод
1	2.7380174596563809915899
2	
2	

Задача Е. Горилла залезает на дерево

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

Однажды горилла нашла дерево на n вершинах. Корнем этого дерева является вершина 1, а на рёбрах этого дерева записаны веса. Любая порядочная горилла захотела бы залезть на такое дерево и эта не стала исключением. Обратите внимание, что в этой задаче корень дерева находится наверху.

Горилла может начать свой путь в любой вершине с номером v ($2 \leq v \leq n$). Изначально её усталость равна 1. Когда горилла поднимается по ребру наверх (ближе к корню), её усталость **умножается** на вес этого ребра.

Горилла имеет выносливость k , поэтому она откажется идти по ребру, после которого её усталость станет **строго больше** k и **немедленно** закончит путешествие. Если такое ребро не встретилось, горилла закончит путешествие в корне.

Назовём *продолжительностью* путешествия количество рёбер, по которым прошла горилла. Обозначим *продолжительность* путешествия из вершины v как d_v .

Помогите горилле узнать числа d_2, d_3, \dots, d_n . Другими словами, найдите *продолжительности* путешествия, если горилла будет начинать свой путь из вершины с номером v ($2 \leq v \leq n$).

Формат входных данных

В первой строке дано целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных.

Далее следует описание наборов.

В первой строке каждого набора даны целые числа n, k ($2 \leq n \leq 2 \cdot 10^5, 1 \leq k \leq 10^9$) — количество вершин в дереве и выносливость гориллы.

В следующих $n - 1$ строках каждого набора даны целые числа v, u, w ($1 \leq v, u \leq n, 1 \leq w \leq k$) — концы соответствующего ребра в дереве и вес этого ребра.

Гарантируется, что заданный набор рёбер образует дерево. Также гарантируется, что сумма n по всем наборам входных данных не превосходит $2 \cdot 10^5$.

Формат выходных данных

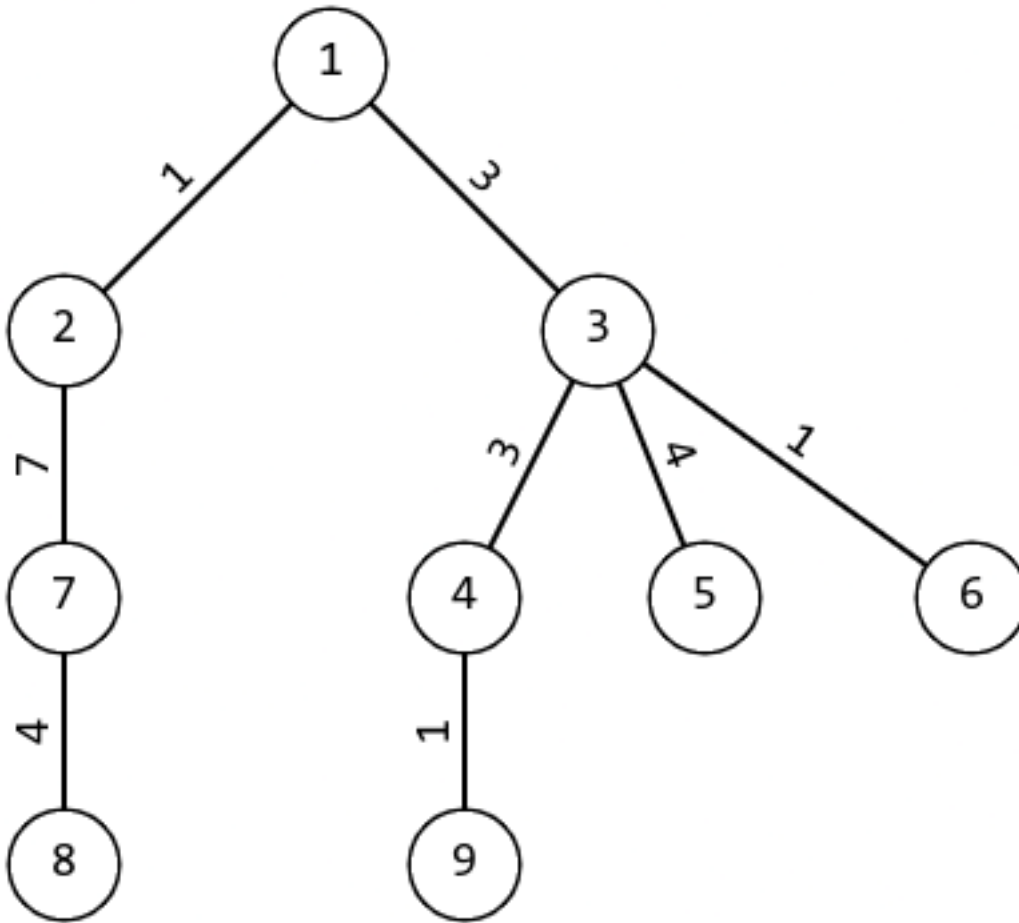
Для каждого набора входных данных выведите n целых чисел d_2, d_3, \dots, d_n .

Пример

стандартный ввод	стандартный вывод
3	1 1 2 1 2 2 1 3
9 10	1 2 3
1 2 1	1 1 1 1
3 1 3	
3 5 4	
4 3 3	
7 8 4	
6 3 1	
2 7 7	
4 9 1	
4 1	
1 2 1	
2 3 1	
3 4 1	
5 1000000000	
2 1 1000000000	
1 5 10	
3 1 1	
4 2 1000000000	

Замечание

Дерево в первом наборе входных данных первого теста выглядит так:



Задача F. Название команды

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

Во Всеберляндской олимпиаде по программированию могут участвовать команды в составе n человек. Для участия каждой команде нужно выбрать название.

Участники одной из команд составили название, включающее каждое из их имен, и записали его в строку t длины m . Однако, такое название получилось слишком длинным! Поэтому они хотят сократить его следующим образом:

- выбрать префикс строки t минимальной длины, который включал бы все их имена без пересечений.

Префиксом строки называется строка, полученная удалением нескольких (возможно, нуля) последних символов из исходной строки.

Некоторое множество подстрок входит в строку *без пересечений*, если никакой символ не принадлежит двум подстрокам одновременно. Например, подстроки «a» и «bc» входят в строку «ababc» без пересечений, а подстроки «aba» и «abc» — нет.

Если название команды возможно сократить — выведите длину минимального подходящего префикса.

Формат входных данных

Первая строка входных данных содержит два целых числа: n ($2 \leq n \leq 8$) — количество участников в команде, и m ($2 \leq m \leq 2 \cdot 10^5$) — длину названия, составленного командой.

Вторая строка входных данных содержит строку t , состоящую из строчных латинских букв — название, составленное командой.

Далее следуют n строк, каждая из которых содержит строку s_i , состоящую из строчных латинских букв ($1 \leq |s_i| < m$) — имя i -го участника команды ($1 \leq i \leq n$). Длина строки s_i обозначается как $|s_i|$.

Гарантируется, что сумма значений $|s_i|$ не превосходит значения m , и в строку t можно поместить имена всех участников без пересечений.

Формат выходных данных

Выведите:

- Число -1 , если название команды невозможно сократить;
- Иначе длину минимального префикса строки t , в который могут войти имена всех участников без пересечений.

Примеры

стандартный ввод	стандартный вывод
4 22 omihaakrisssashaanna kris miha sasha anna	20
2 16 coolnickgoodalex nick alex	-1
3 7 bababab a bab b	5

Задача А. VK Музыка

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

Студентка МИСИС Аня любит слушать музыку в приложении VK Музыка. В приложении ей очень нравится то, что можно посмотреть разные плейлисты, например своих друзей, и увидеть на сколько процентов каждый плейлист подходит ей по интересам.

В очередной раз Аня открыла приложение VK Музыка и увидела семь новых плейлистов. У каждого из них был указан процент совместимости с музыкальными вкусами Ани. Аня решила, что послушает только те плейлисты, которые подходят под её вкусы **не менее**, чем на 90 процентов.

Подскажите Ане, сколько плейлистов она послушает.

Формат входных данных

Входные данные состоят из семи строк. В каждой строке задано единственное целое число от 0 до 100 — совместимость в процентах очередного плейлиста со вкусами Ани.

Формат выходных данных

В единственной строке выведите единственное целое число — количество плейлистов, которые слушает Аня.

Пример

стандартный ввод	стандартный вывод
90	3
91	
95	
47	
32	
20	
19	

Задача В. Строка из палиндромов

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

Вам дана строка s , состоящая из n строчных латинских букв. Определите, можно ли, используя ее символы, составить **ровно** m палиндромов так, чтобы каждый символ входил ровно в **один** палиндром?

Строка является палиндромом, если она читается одинаково как слева направо, так и справа налево. Например, строки «abacaba», «сссс», «z» и «dxd» являются палиндромами, а строки «abab» и «aaabaa» — нет.

Например, пусть $s = \text{«ababcab»}$, $n = 7$, $m = 3$. Тогда из ее букв можно составить 3 палиндрома:

- «аса»
- «bcb»
- «а»

Формат входных данных

Первая строка входных данных содержит единственное целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных в тесте.

Далее следуют описания наборов входных данных.

Первая строка описания каждого набора входных данных содержит два целых числа n и m ($1 \leq m \leq n \leq 2 \cdot 10^5$) — длина строки и количество палиндромов, которые необходимо составить из ее символов.

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

Гарантируется, что сумма длин всех строк в тесте не превосходит $2 \cdot 10^5$.

Формат выходных данных

Для каждого набора входных данных в отдельной строке выведите:

- «YES», если из символов строки возможно составить **ровно** m палиндромов так, чтобы каждый символ входил ровно в один палиндром;
- «NO» иначе.

Вы можете выводить ответ в любом регистре (например, строки «yEs», «yes», «Yes» и «YES» будут распознаны как положительный ответ).

Пример

стандартный ввод	стандартный вывод
3	YES
7 3	NO
ababcbab	YES
5 2	
cabad	
6 6	
vkvkvk	

Замечание

Первый набор входных данных разобран в условии задачи.

Во втором наборе входных данных из символов строки нельзя получить 2 палиндрома.

Задача С. Судьба

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

Эмия Кирицугу потерял своего верного слугу в тяжёлой схватке и сейчас пытается укрыться в семейном поместье Айнцберн. В нём n комнат, которые соединены m коридорами.

Великий царь Гильгамеш преследует Эмию и уже добрался до его поместья. За один час Гильгамеш **одновременно** уничтожает **все** комнаты из которых выходит **наибольшее** количество коридоров. Вместе в комнате уничтожаются **все** коридоры, которыми она была связана с какими-то другими комнатами.

Эмия Кирицугу очень устал, поэтому хочет перевести силы перед битвой с Гильгамешем. Другими словами, он хочет прятаться как можно дольше, поэтому заранее выберет одну из комнат, которую Гильгамеш уничтожит последней.

Помогите Кирицугу определить, какое максимальное количество часов он сможет прятаться от Гильгамеша.

Формат входных данных

В первой строке дано целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных.

В первой строке каждого набора даны целые числа n, m ($2 \leq n \leq 2 \cdot 10^5, 1 \leq m \leq 2 \cdot 10^5$) — количество комнат и коридоров соответственно.

В следующих m строках каждого набора даны целые числа v, u ($1 \leq v, u \leq n, v \neq u$) — комнаты, соединённые соответствующим коридором.

Между одной парой комнат могут быть несколько коридоров. Кроме того, **не** гарантируется, что существует путь по коридорам между каждой парой комнат.

Гарантируется, что сумма n по всем наборам входных данных не превосходит $2 \cdot 10^5$. То же самое гарантируется для m .

Формат выходных данных

Для каждого набора входных данных выведите единственное целое число — максимальное количество часов, которое сможет прятаться Эмия Кирицугу.

Пример

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

Задача D. Правильный шестиугольник

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

Женя решил подготовить на Новый Год украшения на ёлку. А самое лучшее украшение — это правильный шестиугольник! Поэтому Женя поручил своей младшей сестре Кате сделать несколько правильных шестиугольников.

Работу Женя организовал следующим образом. Сначала Катя на клетчатом листке бумаги рисует шесть точек в узлах сетки так, чтобы получился выпуклый шестиугольник. Затем Женя проверяет является ли этот шестиугольник правильным. Если шестиугольник правильный, то Катя его вырезает и вешает на ёлку, а иначе Катя выбрасывает этот листок.

Женя догадывается, что правильный шестиугольник в узлах сетки может не существовать, поэтому допускает некоторые отклонения. Женя будет считать, что выпуклый шестиугольник является правильным, если выполнены три условия:

- $\frac{\text{SideMax}}{\text{SideMin}} \leq \frac{111}{100}$, где SideMax и SideMin — максимальная и минимальная длина стороны многоугольника.
- $\frac{\text{DiagMax}}{\text{DiagMin}} \leq \frac{111}{100}$, где DiagMax и DiagMin — максимальная и минимальная длина диагонали, соединяющей противоположные вершины шестиугольника.
- Существует точка O в узле сетки такая, что $\frac{\text{DistMax}}{\text{DistMin}} \leq \frac{111}{100}$, где DistMax и DistMin — максимальная и минимальная длина отрезка соединяющего точку O с вершиной шестиугольника.

Формат входных данных

В первой строке дано единственное целое число t ($1 \leq t \leq 1000$) — количество шестиугольников, которые сделала Катя.

В следующих строках заданы t шестиугольников. Входные данные для каждого шестиугольника занимают шесть строк. В шести строках, задающих очередной шестиугольник, даны через пробел по два целых числа X, Y ($0 \leq X, Y \leq 1000$) — координаты очередной вершины шестиугольника.

Вершины каждого шестиугольника даны в порядке обхода его вершин по часовой стрелке. Гарантируется, что каждый заданный шестиугольник является строго выпуклым.

Формат выходных данных

В единственной строке выведите единственную строку S длины t . Если i -й шестиугольник является правильным по критерию Жени, то i -й символ строки S должен быть равен '1', а иначе i -й символ строки S должен быть равен '0'.

Пример

стандартный ввод	стандартный вывод
2 0 3 2 6 6 6 8 3 6 0 2 0 0 1 1 2 2 2 3 1 2 0 1 0	10

Задача Е. Возводи в степень и суммируй!

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

Аналитик Жора работает в маленькой компании, где он должен выполнять определенные вычисления. Еще десять лет назад ему выдали массив чисел. После этого каждый день он должен был вычислять сумму чисел на отрезке, концы которого задавали ему утром.

Жора изучал программирование в МИСИС, поэтому без труда написал программу, которая автоматически вычисляет требуемую сумму. Благодаря этому у Жоры была возможность весь день пить чай с печеньем и читать классику мировой литературы.

Но в один прекрасный день пришел начальник Жоры и сказал, что появилась необходимость иногда менять имеющийся массив. Начальник пояснил как именно нужно менять массив и ушел.

Новость очень шокировала Жору, ведь его привычный уклад жизни был под угрозой. Он сразу понял, что нужно переписать свою программу таким образом, чтобы она выполняла еще и изменения массива. Но за десять лет чтения классической литературы Жора подзабыл основы программирования.

Помогите Жоре написать программу, которая будет вычислять запросы и выполнять изменения в массиве.

Будем считать, что массив Жоры состоит из n целых положительных чисел a_1, a_2, \dots, a_n . Запросы, которые необходимо выполнять, бывают трех типов:

- 1 k — запрос первого типа, в котором необходимо все элементы массива возвести в степень k . Если после выполнения этого запроса хотя бы одно число станет больше, чем 10^5 , то этот запрос нужно проигнорировать.
- 2 k — запрос второго типа, в котором необходимо из всех элементов массива извлечь корень степени k . Если после выполнения этого запроса хотя бы одно число перестанет быть целым, то этот запрос нужно проигнорировать.
- 3 $l r$ — запрос третьего типа, в котором необходимо вычислить сумму элементов с номерами от l до r включительно.

Формат входных данных

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

Во второй строке через пробел заданы n целых чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^5$) — элементы массива.

В третьей строке задано целое число q ($1 \leq q \leq 10^5$) — количество запросов.

В следующих q строках заданы запросы по одному в строке. Запросы бывают трех видов:

- 1 k ($2 \leq k \leq 10^5$) — запрос первого типа.
- 2 k ($2 \leq k \leq 10^5$) — запрос второго типа.
- 3 $l r$ ($1 \leq l \leq r \leq n$) — запрос третьего типа.

Формат выходных данных

На каждый запрос третьего типа выведите в отдельной строке ответ на этот запрос.

Примеры

стандартный ввод	стандартный вывод
5	15
1 2 3 4 5	55
14	979
3 1 5	979
1 2	15
3 1 5	1
1 2	2
3 1 5	3
2 3	4
3 1 5	5
2 4	
3 1 5	
3 1 1	
3 2 2	
3 3 3	
3 4 4	
3 5 5	
1	12
12	
3	
1 6	
2 3	
3 1 1	

Задача F. Метро в Берляндии

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

В Берляндии открылось метро. Жители рады новому виду транспорта, а особенно рад мэр города. Однако всегда есть что улучшить.

Метро представляет собой связный граф с n вершинами и $n - 1$ ребрами. Мэр хочет улучшить метро, делает он это с помощью приказов. Приказы бывают двух видов:

- $+ v$: Построить новую станцию и соединить ее со станцией v . Если номер последней построенной станции был x , то номер новой станции будет $x + 1$
- $- v$: Разрушить станцию v . Гарантируется, что после удаления этой станции, схема метро остается связной, то есть от любой станции можно добраться до любой другой.

Мэру города после каждого своего приказа интересно, за какое минимальное количество минут можно посетить все станции хотя бы один раз. Свой путь можно начинать с любой станции, а время перемещения между двумя соединенными станциями составляет ровно **одну** минуту.

Формат входных данных

Первая строка содержит число n ($2 \leq n \leq 10^5$) — количество станций метро в изначальной схеме.

Далее идут $n - 1$ строк. В i -й строке содержатся два числа u_i и v_i ($1 \leq u, v \leq 10^5$) — номера соединённых станций.

На следующей строке дано число q ($1 \leq q \leq 10^5$) - количество приказов мэра.

В следующих q строках даются описания приказов в следующем виде:

- $+ v$ — построить новую станцию;
- $- v$ — разрушить станцию v .

Гарантируется, что v это номер существующей на данный момент станции.

Формат выходных данных

После каждого приказа выведите минимальное количество минут, за которое можно посетить все станции хотя бы один раз.

Пример

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