Задача A. Конспирация

Имя входного файла:	`conspiracy.in`
Имя выходного файла:	`conspiracy.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Петя и Вася играют в шпионов. Главная цель этой игры — обмениваться секретными сообщениями, сохраняя конспирацию. Для этого Петя и Вася используют особый шифр.

Метод шифровки и дешифровки очень прост: каждой букве α русского алфавита сопоставляется некоторая последовательность букв латинского алфавита p_α. Затем в сообщении все вхождения каждой буквы α заменяются на соответствующее ей p_α. А чтобы совсем запутать вероятного противника, Петя и Вася договорились, что будут отправлять n слов, значимым из которых будет только лексикографически минимальное слово. Напомним, что строка a лексикографически меньше строки b, если существует такое i, что для любого $j < i$ выполняется равенство $a_j = b_j$ и при этом $a_i < b_i$, или строка a является префиксом строки b.

Недавно Петя предложил Васе использовать следующую таблицу преобразования букв:

а	a	з	z	п	p	ч	tch
б	b	и	i	р	r	ш	sch
в	v	й	j	с	s	щ	scsh
г	g	к	k	т	t	ы	yy
д	d	л	l	у	u	э	ye
е	e	м	m	ф	f	ю	yu
ё	yo	н	n	х	kh	я	ya
ж	zh	о	o	ц	tc		

Однако теперь Вася не успевает дешифровывать сообщения Пети вовремя и просит вас помочь ему — определить номер слова, которое до шифрования, записанное русскими буквами, было лексикографически минимальным.

Формат входного файла

Первая строка входного файла содержит одно целое число n ($1 \le n \le 1000$) — количество слов в шифровке. Следующие n строк содержат по одному слову, состоящему из маленьких латинских букв — текст шифровки, которую получил Вася. Длина слова не превышает 50.

Формат выходного файла

В выходной файл выведите номер слова шифровки, которое до шифрования было лексикографически минимальным в русском алфавите. Шифровки нумеруются с единицы в порядке появления во входном файле.

Примеры

conspiracy.in	conspiracy.out
6 yolka jeti yabloko segodnya schtab kaktus	1

Примечание

Пояснение к примеру: исходное сообщение состояло из слов «ёлка», «йети», «яблоко», «сегодня», «штаб», «кактус».

Задача B. Бункеры

Имя входного файла:	`bunkers.in`
Имя выходного файла:	`bunkers.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Петя и Вася с упоением играют в шпионов. Сегодня они планируют, где будут расположены их секретные бункеры и штаб-квартира.

Пока Петя и Вася решили, что им понадобится ровно n бункеров, которые для секретности будут пронумерованы числами от 1 до n. Некоторые из них будут соединены двусторонними тоннелями, причем для надежности и секретности по тоннелям можно будет попасть из любого бункера в любой единственным образом. Петя и Вася даже решили, какие из бункеров будут соединены тоннелями, но выбрать, какой из них будет штаб-квартирой, они не могут. Мальчики хотят выбрать ее и разделить оставшиеся бункеры между собой таким образом, чтобы им досталось поровну бункеров и к штаб-квартире вело бы ровно два тоннеля: один от бункера, принадлежащего Васе, другой — от бункера, принадлежащего Пете.

Уставший Петя пошел к себе домой, а утром Вася показал ему план, на котором бункеры были обозначены точками, а тоннели отрезками. Кроме того, Вася выбрал штаб-квартиру таким образом, что нарисованный им план был симметричен относительно некоторой прямой, проходящей через точку, которая соответствовала штаб-квартире. При этом бункеры, принадлежащие Пете находились с одной стороны от этой прямов, а бункеры, принадлежащие Васе, с другой стороны.

Хотя Петя почти сразу показал Васе, что тот ошибся и не нарисовал половину бункеров, ему стало интересно, можно ли выбрать штаб-квартиру и нарисовать такой симметричный план.

Формат входного файла

В первой строке входного файла находится одно целое число n ($1 \le n \le 10^5$) — количество бункеров. В следующих $n-1$ строках находится по два целых числа u_i и v_i ($1 \le u_i, v_i \le n, u_i \ne v_i$) — номера бункеров, которые соединяет i-й тоннель. Гарантируется, что между любыми двумя бункерами существует единственный путь.

Формат выходного файла

В выходной файл выведите «`YES`», если можно выбрать штаб-квартиру и нарисовать такой план, или «`NO`» если это невозможно.

Примеры

bunkers.in	bunkers.out
2 1 2	NO
3 1 2 2 3	YES

Задача C. Игра

Имя входного файла:	`game.in`
Имя выходного файла:	`game.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Дедушка Марат очень любит свою внучку Машу. Тем более, что из всей его семьи она самая младшая. Поэтому он часто играет с ней в разные игры.

Сегодня, например, они играли в развивающую мышление игру, которая называется «Угадай слово». Правила игры достаточно просты:

- Сначала Дедушка загадывает слово. Словом считается любая последовательность из строчных латинских символов. Единственное ограничение — это слово должно быть палиндромом. Напомним, строка S длины n называется палиндромом, если $\forall i \in [1, n] : S[i] = S[n - i + 1]$.
- За один ход Маша может выбрать произвольное слово P и дописать P к S слева или справа.
- Если полученная строка не является палиндромом, то дедушка говорит об этом Маше и игра заканчивается
- Если полученная строка все еще палиндром, то игра продолжается
- Когда игра закончилась, Маша должна назвать загаданное слово

Дедушка записал все ходы своей внучки за всю игру. Теперь же он хочет вспомнить, чем закончилась игра. Если Маша отгадала слово, то он хочет определить, какое оно было. Иначе определить, не угадала она из-за того, что у нее оставалось несколько вариантов, или из-за того, что он ошибся в процессе игры.

Формат входного файла

Первая строка входного файла содержит два целых числа n и m ($1 \le n, m \le 3000$) — длину загаданного слова и количество ходов в игре, соответственно.

Следующие m строк содержат по одному слову s_i, состоящему из строчных латинских букв — слово, выбранное Машей на i-м ходу. Сумма длин всех s_i не превышает 3000.

Последняя строка содержит слово w из m символов «L» или «R». $w_i = L$ если слово s_i было записано слева от текущей строки на i-м ходу, $w_i = R$ — если справа.

Формат выходного файла

В выходной файл выведите слово «`Unique`», если Маша может однозначно установить, какое слово загадал дедушка, «`Ambiguous`» — если таких слов могло быть несколько, либо «`Impossible`» — если таких слов не существует и дедушка ошибся.

Примеры

game.in	game.out
3 2 a a LR	Impossible
7 1 abc L	Ambiguous

Задача D. k-сортировка

Имя входного файла:	`ksort.in`
Имя выходного файла:	`ksort.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В этом году Гриша поступил в Университет ИТ. В Университете ИТ очень много новых предметов, интересных и не очень. Особенно Грише нравится предмет «Алгоритмы и структуры данных». На последней лекции были рассказаны алгоритмы сортировки. Гриша — очень амбициозный молодой человек и хочет изобрести свой алгоритм, который впоследствии будет назван именем его любимого дедушки. Вдохновившись чтением многотомника Кнута, Гриша решил модернизировать какой-нибудь уже существующий алгоритм сортировки натуральных чисел, наложив следующее ограничение. Любые два элемента можно менять местами, только если они сравнимы по модулю некоторого натурального числа k, то есть дают одинаковые остатки при делении на k. Но все инновационные методы требуют проверки, поэтому Гриша обратился за помощью к Вам!

Проверьте, сможет ли новая версия алгоритма отсортировать заданный массив натуральных чисел.

Формат входного файла

Первая строка входного файла содержит два числа n ($1 \le n \le 1000$) и k ($1 \le k \le 10^9$) — количество элементов в массиве и число, по модулю которого сравниваются элементы массива.

Вторая строка входного файла содержит n целых чисел a_i — элементы массива ($1 \le a_i \le 10^9$).

Формат выходного файла

В выходной файл выведите «YES», если алгоритм сможет отсортировать заданный массив и «NO» — в обратном случае.

Примеры

ksort.in	ksort.out
5 2 5 4 3 2 1	YES
3 2 2 3 1	NO

Задача E. Зарплата

Имя входного файла:	`salary.in`
Имя выходного файла:	`salary.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Дима совсем недавно устроился на работу. Так как он очень ценный работник, его зарплата будет расти не по дням, а по часам. Если точнее, то каждый месяц количество рублей, составляющих его зарплату, будет равно возведенному в третью степень номеру месяца. Так, в первый месяц он получит 1 рубль, во второй — 8, в третий — 27 и так далее.

Как всем известно, законы о налогообложении — вещь сложная. Недавно в стране, в которой работает Дима, был принят новый закон. Суть его заключается в том, что процент налога, ежемесячно выплачиваемый сотрудником вычисляется по следующей формуле:

- Записывается зарплата сотрудника за первый месяц, к ней справа приписывается его зарплата за второй месяц и так далее, пока суммарное количество цифр в записанном числе не будет больше, чем k
- k-ая цифра записанного числа и является искомым процентом

Для каждой организации было введено свое k. Дима знает это число для организации, в которой он работает. И теперь он хочет узнать, сколько процентов заработка он будет ежемесячно отдавать в качестве налога.

Формат входного файла

Первая строка входного файла содержит единственное целое число k ($1 \le k \le 10^{18}$).

Формат выходного файла

Выведите в выходной файл одно целое число x ($0 \le x \le 9$) — искомый процент.

Примеры

salary.in	salary.out
3	2
7	1

Примечание

При вычислении налога во втором примере получается следующая строка: 182764125. Седьмая цифра при этом равна 1.

Задача F. Часовые

Имя входного файла:	`sentinels.in`
Имя выходного файла:	`sentinels.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Петя и Вася продолжали играть в шпионов. Однако на этот раз вероятный противник в лице Васиной бабушки захватил Васю и увез на секретную дачу. В течение нескольких дней Вася ожидал спасения, когда вдруг понял, что Петя знает, где находится дача его бабушки. А это никак не укладывалось в сценарий игры в шпионов. Чтобы хоть немного спасти положение, Вася решил поставить в лесу вокруг бабушкиной дачи часовых — других своих друзей, которые должны мешать Пете его спасать.

Для точности Вася взял план местности, ввел на нем систему координат, направив ось Oy на север, а ось Ox — на восток, и отметил на нем часовых. Кроме того, каждому часовому Вася выдал мощный фонарик, который освещает угол величины 90 градусов перед часовым. Так как фонарики мощные, то они светят практически на любое расстояние. Затем Вася решил, что Пете может быть слишком сложно пробираться на дачу, и разрешил часовым светить фонариком только в одном из четырех направлений — строго на север, запад, юг или восток, так чтобы биссектриса угла, освещаемого фонариком часового, совпадала с этим направлением.

Теперь, чтобы оценить свои шансы быть спасенным, Вася хочет для каждого часового узнать скольких часовых, помимо него самого, освещает его фонарик.

Формат входного файла

В первой строке входного файла дано целое число n ($1 \le n \le 50000$) — число часовых в лесу. В следующих n строках содержатся данные о часовых. В i-й строке записаны два целых числа x_i и y_i ($|x_i|, |y_i| \le 10^9$) — координаты i-го часового на плане Пети и направление, куда смотрит часовой — N, E, S, W, соответствующие северу, востоку, югу и западу. Никакие два часовых не находятся в одной точке.

Формат выходного файла

В выходной файл выведите n строк. В i строке должно содержаться единственное целое число — скольких часовых освещает фонарик i-го часового, помимо него самого.

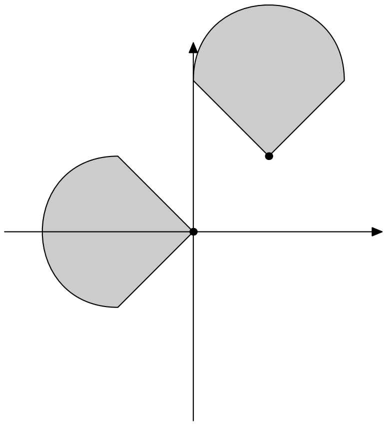
Примеры

sentinels.in	sentinels.out
2 0 0 W 1 1 N	0 0
4 0 0 E 1 0 N 1 1 W 0 1 S	2 2 2 2

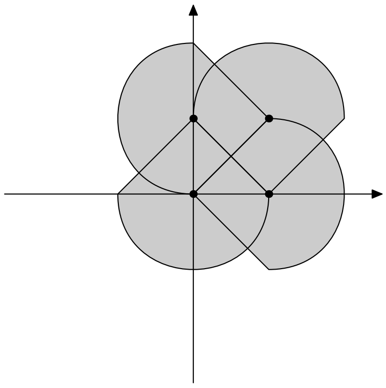
Примечание

Пояснения к примерам:

Первый пример:



Второй пример:



Задача G. Траволатор

Имя входного файла:	`travelator.in`
Имя выходного файла:	`travelator.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Боб — известный ученый, и поэтому он часто посещает различные конференции. Во время очередной поездки Боб опаздывал на самолет, и поэтому ему необходимо было попасть в терминал А как можно быстрее. Известно, что в аэропорту есть горизонтальный траволатор, благодаря которому Боб может передвигаться по аэропорту быстрее. Бобу известен план аэропорта, с обозначенными на нем входом, терминалом А и концами траволатора.

Если аэропорт представить в виде координатной плоскости, то вход в аэропорт находится в точке (0,0), терминал А в точке (x_1,y_1), начало и конец траволатора в точках (x_2,y_2) и (x_3,y_3). Также известно, что скорость Боба v, скорость траволатора w и по траволатору можно двигаться только из начала в конец. Помогите Бобу найти минимальное время, за которое он доберется от входа в аэропорт к терминалу А. При этом, так как у эскалатора есть перила, запрещено его пересекать, даже двигаясь вне траволатора, а можно только войти на него в точке начала траволатора и двигаться со скоростью w вплоть до конца траволатора.

Формат входного файла

Первая строка входного файла содержит шесть целых чисел x_1, y_1, x_2, y_2, x_3, y_3 $(-10000 \le x_1, x_2, x_3 \le 10000, 0 < y_1, y_2, y_3 \le 10000)$ — координаты терминала А, начала и конца траволатора. Вторая строка содержит два целых числа v, w $(0 < v \le 50, 0 < w \le 50)$ — скорости Боба и траволатора.

Известно что терминал А и вход в аэропорт не лежат на траволаторе.

Формат выходного файла

Выведите минимальное время, за которое Боб может добраться от входа до терминала А. Ответ считается правильным, если он отличается от минимального не более чем на 10^{-5}.

Примеры

travelator.in	travelator.out
0 3 0 1 0 2 1 4	2.25
0 3 -2 1 2 1 1 4	5.064495

Примечание

В первом примере Бобу выгодно воспользоваться траволатором.

Задача H. Где я?

Имя входного файла:	`where.in`
Имя выходного файла:	`where.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Дедушка Марат живет в далеком-далеком городе Ч. Дедушка очень любит ходить в гости, иногда он уходит на несколько дней, обходя при этом очень-очень много своих друзей. Дедушка, уходя из очередного дома, всегда идет только к друзьям хозяев этого дома. К некоторым Дедушка Марат мог заходить по нескольку раз. Дедушка мог даже заходить к себе домой попить чаю с внуками. Однако Дедушка очень забывчив, поэтому он иногда попросту забывает вернуться домой. Его внуки очень волнуются за него, поэтому всегда находят его и возвращают его домой. За несколько лет внуки поняли, что прежде чем они успевают найти Дедушку Марата, он успевает обойти ровно k друзей (внуки тоже считаются друзьями).

Несколько дней назад Дедушка Марат снова ушел погостить, и внуков интересует, где же они могут его встретить? Помогите им узнать ответ на этот вопрос.

Формат входного файла

Первая строка входного файла содержит три числа n, m и k, где n — количество домов в городе Ч., а m — количество пар друзей ($1 \le n \le 1\,000$, $1 \le m \le 200\,000$, $1 \le k \le 10^9$).

Следующие m строк содержат описания пар друзей, по одному на каждой строке. Описание состоит из двух чисел — номера домов, хозяева которых дружат (если хозяева дома i дружат с хозяевами дома j, то и хозяева дома j дружат с хозяевами дома i

Дедушка Марат и внуки живут в доме с номером 1.

Формат выходного файла

В первой строке выходного файла должно быть число p — количество домов, в которых мог оказаться Дедушка Марат. Во второй строке должно быть p чисел — номера домов, в которых мог оказаться Дедушка Марат, в возрастающем порядке.

Примеры

where.in	where.out
3 3 3 1 2 1 3 2 3	3 1 2 3

Задача I. Камни

Имя входного файла:	`stones.in`
Имя выходного файла:	`stones.out`
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

У Васи есть несколько плоских камней. Каждый камень белый с одной стороны и чёрный с другой. Он разложил их в ряд чёрной стороной вниз. Вася догадливый и заметил, что картинка перед ним — на самом деле двоичное число. Картина переводится в двоичное число следующим образом. Камень, лежащий белой стороной вверх считается нулём, чёрной — единицей. Первый камень берётся с коэффициентом 1, второй — 2, третий — 4, и т.д.

Теперь он задумал число n и хочет получить его. Однако, единственное, что он может делать — это перевернуть какой-то отрезок камней ненулевой длины. При перевороте отрезка все камни на нём переворачиваются с чёрной стороны на белую, с белой — на чёрную. Однако, даже отрезки он может переворачивать не все. Он заметил, что переворачиваемый отрезок на самом деле тоже двоичное число. Он может переворачивать только те отрезки камней, соответствующее которым число не превышает k.

Также, Вася не хочет повторяться, и все переворачиваемые им отрезки должны быть различны. Его заинтересовало количество способов добиться своей цели, а именно, получить картину, соответствующую числу n. Так как такое количество может быть большим, выведите его остаток от деления на $10^9 + 7$.

Формат входного файла

В первой строке входного файла через пробел заданы числа n ($1 \le n \le 10^{18}$) и k ($1 \le k \le 10^{18}$).

Формат выходного файла

Выведите требуемое количество способов по модулю $10^9 + 7$.

Примеры

stones.in	stones.out
3 4	2