

Задача А. Рассадка

Имя входного файла: **input.txt** или стандартный поток ввода
Имя выходного файла: **output.txt** или стандартный поток вывода
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Олимпиада всегда приносит много хлопот для жюри и оргкомитета, и очный тур Открытой олимпиады не является исключением из этого правила. Пока пишутся условия, готовятся тесты, создаются разборы и настраиваются компьютеры, очень легко забыть правильно рассадить участников олимпиады по аудиториям.

В этот раз, в целях рационального использования рабочих мест, оргкомитет олимпиады поставил перед собой задачу — создать *компактную* рассадку школьников по M имеющимся аудиториям. Рассадка называется *компактной*, если для любых двух различных аудиторий количества детей в них отличаются не более чем на один.

В жюри олимпиады имеется человек с экстрасенсорными способностями, который может очень точно предсказывать результаты будущих соревнований по программированию. Хотя результаты заочного тура ещё не известны, он без труда определил, что на очный тур пройдут ученики из K школ. Кроме того, было предсказано, что из i -й школы ($1 \leq i \leq K$) пройдёт в точности n_i учеников. Хотя отношение большей части оргкомитета к подобным прогнозам зачастую скептическое, они имеют странное свойство сбываться, поэтому вам предлагается на всякий случай вычислить максимально возможное количество участников, которые попадут в одну аудиторию при компактной рассадке, если предсказание окажется верным.

Формат входных данных

В первой строке ввода находятся два целых числа M и K — количество аудиторий и количество школ, участвующих в очном туре олимпиады согласно предсказанию ($1 \leq M \leq 10^9$, $1 \leq K \leq 100\,000$).

В следующей строке находятся K чисел n_i , разделённых пробелами, i -е из которых обозначает, сколько участников предположительно приедет из i -й школы ($1 \leq n_i \leq 10^9$).

Формат выходных данных

Выведите одно число — максимальное количество людей в аудитории при компактной рассадке.

Примеры

ВВОД	ВЫВОД
2 2 5 9	7
1 2 20 10	30

Замечание

В первом тесте существует только одна компактная рассадка — в каждую из аудиторий рассадить по 7 человек.

Система оценки

Тесты к этой задаче состоят из пяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп..

Группа	Тесты	Баллы	Дополнительные ограничения			Комментарий
			K	M	n_i	
0	1–2	0	–	–	–	Тесты из условия.
1	3–13	25	$K \leq 2$	$M \leq 2$	$n_i \leq 100$	
2	14–20	25	$K \leq 2$	$M \leq 3$	–	
3	21–27	25	$K \leq 4$	–	–	
4	28–34	25	–	–	–	Дополнительных ограничений нет.

Задача В. Расписание

Имя входного файла:	input.txt или стандартный поток ввода
Имя выходного файла:	output.txt или стандартный поток вывода
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Как же Илье надоело учиться! Сначала школа, потом университет... Вот, наконец, наступил тот долгожданный день, когда Илье утром не надо ехать на учебу. Но, к несчастью для Ильи, оказалось, что после окончания университета начинается самое трудное — надо устраиваться на работу.

Во всемирно известной фирме «Goondex», в которую устроился Илья, принято очень много работать, в частности, для сотрудников установлена шестидневная рабочая неделя. Но, в качестве бонуса, «Goondex» каждый год предлагает своим сотрудникам выбрать любой день недели в качестве выходного. В свою очередь, оставшиеся шесть дней недели будут рабочими.

Илья сообразил, что с учётом государственных праздников (которые всегда являются выходными) с помощью правильного выбора выходного дня недели можно варьировать количество рабочих дней в году. Теперь он хочет знать, какой день недели ему следует выбрать в качестве выходного, чтобы отдыхать как можно больше дней в году, или, наоборот, демонстрировать чудеса трудолюбия, работая по максимуму.

Формат входных данных

В первой строке входных данных находится одно целое число N ($0 \leq N \leq 366$) — количество государственных праздников.

Во второй строке содержится одно целое число $year$ ($1800 \leq year \leq 2100$) — год, в который необходимо помочь Илье.

В каждой из последующих N строк расположено по паре чисел $day\ month$ (day — целое число, $month$ — слово, между day и $month$ ровно один пробел), обозначающих, что день day месяца $month$ является государственным праздником.

В последней строке расположено слово day_of_week — день недели первого января в год $year$.

Гарантируется, что все даты указаны корректно (в том числе указанный день недели первого января действительно является днём недели первого января соответствующего года $year$) и все дни государственных праздников различны.

Формат выходных данных

Выведите через пробел два дня недели — лучший и худший варианты дней недели для выходного (то есть дни недели, для которых достигается соответственно максимальное и минимальное количество выходных дней в году). Если возможных вариантов ответа несколько, выведите любой из них.

Примеры

ВВОД	ВЫВОД
2 2015 1 January 8 January Thursday	Monday Thursday
3 2013 1 January 8 January 15 January Tuesday	Monday Tuesday
3 2013 6 February 13 February 20 February Tuesday	Tuesday Wednesday

Замечание

Рассмотрим подробно **третий пример**.

2013 год начинается и заканчивается во вторник (**Tuesday**), при этом на вторник приходится 53 дня года, а на все остальные дни недели — по 52 дня. Все три государственных праздника выпадают на среду (**Wednesday**). Если Илья выберет в качестве выходного дня вторник, то в году у него будет $53 + 3 = 56$ выходных дней (53 вторника и 3 государственных праздника). Если Илья выберет в качестве выходного дня среду, то у него будет только 52 выходных дня. Если же Илья выберет в качестве выходного дня любой другой день недели, то у него будет $52 + 3 = 55$ выходных дней.

Таким образом, лучший вариант для выходного дня — вторник, худший — среда, и **единственным** правильным ответом в данном примере является **Tuesday Wednesday**.

Соответствие названий месяцев и дней недели в английском и русском языках:

Месяцы				Дни недели	
January	Январь	July	Июль	Monday	Понедельник
February	Февраль	August	Август	Tuesday	Вторник
March	Март	September	Сентябрь	Wednesday	Среда
April	Апрель	October	Октябрь	Thursday	Четверг
May	Май	November	Ноябрь	Friday	Пятница
June	Июнь	December	Декабрь	Saturday	Суббота
				Sunday	Воскресенье

Соответствие названий месяцев и количества дней в них:

Январь	31	Июль	31
Февраль	28 (29)	Август	31
Март	31	Сентябрь	30
Апрель	30	Октябрь	31
Май	31	Ноябрь	30
Июнь	30	Декабрь	31

В феврале 29 дней только в високосные года. Год является високосным, если он кратен 400, либо кратен 4 и не кратен 100. Например, 1996 и 2000 являются високосными, а 1999 и 1900 — нет.

Система оценки

Тесты к этой задаче состоят из трёх групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и **всех тестов предыдущих** групп. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Тесты	Баллы	Дополнительные ограничения		Комментарий
			N	$year$	
0	1–3	0	–	–	Тесты из условия.
1	4–24	20	$N \leq 3$	$2000 \leq year \leq 2015$	
2	25–54	80	–	–	Offline-проверка.

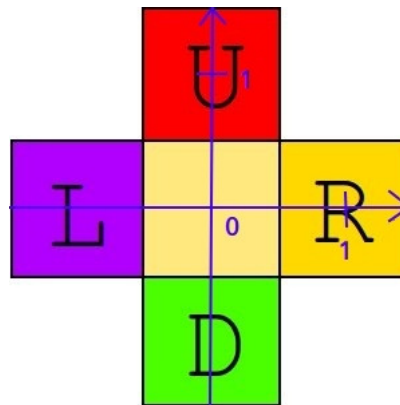
Задача С. DDR

Имя входного файла: **input.txt** или стандартный поток ввода
Имя выходного файла: **output.txt** или стандартный поток вывода
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вася играет в «Dance Dance Revolution». Задача игрока состоит в том, чтобы нажимать ногами расположенные на игровом полу стрелочки в порядке их появления на экране. Игровой процесс сопровождается энергичной музыкой, и невнимательному зрителю может показаться, что Вася просто танцует. На самом же деле он очень сосредоточен на том, чтобы минимизировать суммарное перемещение своих ног, не допустив при этом ни одной ошибки.

Уточним описание игры:

1. Игровое поле состоит из пяти одинаковых по размеру квадратных платформ: центральная без стрелки, а также четыре платформы с различными стрелками, находящиеся по разные стороны от центральной. Будем называть их левой, правой, верхней и нижней стрелками, и обозначим заглавными буквами латинского алфавита 'L', 'R', 'U' и 'D' соответственно. Для лучшего понимания рекомендуется посмотреть на картинку.



2. Песня состоит из N позиций. Для каждой позиции известно, какие стрелки должны быть нажаты. Поскольку Вася всего лишь человек, то это либо одна, либо две одновременно нажатых стрелки.
3. Вася уже достаточно опытный игрок, чтобы стоять на игровом поле как угодно: любая нога Васи может находиться на любом из пяти элементов поля вне зависимости от положения другой ноги (в том числе он может поставить обе ноги на одну и ту же клетку).
4. Вася любит наблюдать за осьминогами, поэтому для него не проблема переместить ноги из любого исходного положения в любое другое.

Цель Васи — нажать на все стрелки в требуемой последовательности. При этом он имеет право стоять на стрелке даже тогда, когда она не должна быть нажата: например, если в данный момент требуется нажать только на стрелку 'L', то он может поставить одну ногу на 'L', а другую — куда угодно. Также Вася в любой момент может поставить одну или обе ноги на центральный элемент без стрелки. Изначально Вася стоит обеими ногами на центральном элементе, а в конце песни его ноги могут находиться где угодно.

В рамках данной задачи будем считать, что ноги Васи являются точками. Для нажатия на платформу Васе нужно поставить ногу в центр соответствующей платформы. Обратите внимание, что это относится и к случаю, когда Вася пользуется своим правом поставить какую-либо ногу на центральную платформу. Таким образом, возможные координаты точек, в которых могут находиться ноги Васи, это $(0, 0)$, $(1, 0)$, $(0, 1)$, $(-1, 0)$ и $(0, -1)$ для центральной, правой, верхней, левой и нижней платформ соответственно.

Помогите Васе найти минимальное расстояние, которое в сумме придётся пройти его ногам во время песни. Здесь под расстоянием подразумевается евклидово расстояние для точек на плоскости. Например, расстояние между центром центральной платформы и центром любой другой равно 1, а расстояние между центрами верхней и правой платформ равно $\sqrt{2}$.

Формат входных данных

В первой строке содержится одно целое число N ($1 \leq N \leq 100\,000$) — количество позиций в песне.

Следующие N строк содержат описания позиций в следующем формате: сначала следует число k_i — количество стрелок, которые должны быть нажаты в данный момент времени ($k_i \in \{1, 2\}$). Затем следуют k_i различных символов (разделённых пробелом, если $k_i = 2$) из множества $\{‘L’, ‘R’, ‘U’, ‘D’\}$, обозначающих стрелки, участвующие в i -й позиции.

Формат выходных данных

Выведите одно вещественное число — минимальное суммарное перемещение ног, которого может достичь Вася, правильно нажав на все стрелки в требуемой последовательности. Ваш ответ будет считаться правильным, если его абсолютная или относительная ошибка не будет превосходить 10^{-9} .

Более формально, пусть ваш ответ равен A , а ответ жюри — B . Проверяющая программа сочтёт ваш ответ правильным, если $\frac{|A-B|}{\max(1, B)} \leq 10^{-9}$.

Примеры

ВВОД	ВЫВОД
3 1 U 1 D 2 L R	4.8284271247
2 1 U 1 U	1.0000000000

Замечание

В первом тесте Вася может первым делом переместить левую ногу на нижнюю платформу, а правую ногу на верхнюю, и пройти тем самым две первых стрелки, а затем переместить обе ноги по диагонали на левую и правую платформу соответственно. Пройденное ногами расстояние будет равно $1 + 1 + \sqrt{2} + \sqrt{2} \approx 4.8284271247$.

Во втором тесте Васе достаточно поставить одну из ног на верхнюю платформу, пройдя расстояние 1.

Система оценки

Тесты к этой задаче состоят из четырёх групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы.

Группа	Тесты	Баллы	Дополнительные ограничения		Комментарий
			N	k_i	
0	1–2	0	—	—	Тесты из условия.
1	3–10	30	$N \leq 20$	$k_i = 2$	Стрелка вниз не встречается.
2	11–19	30	$N \leq 20$	—	
3	20–29	40	—	—	Дополнительных ограничений нет.

Задача D. Мультиплеер в змейке

Имя входного файла:	input.txt или стандартный поток ввода
Имя выходного файла:	output.txt или стандартный поток вывода
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Компания Konia, специализирующаяся на выпуске бюджетных мобильных устройств, в последнее время теряет свои позиции на рынке. Ради спасения компании совет директоров пошел на отчаянный шаг: было решено внести изменения в самую известную игру, которая доступна на всех устройствах от Konia — «Змейку».

Правила этой игры очень просты. На клетчатом поле, состоящем из N строк и M столбцов, находится змейка, изначально имеющая длину 1 (т.е. представляющая собой квадратик 1×1). На каждом ходу змейка перемещается на одну клетку в одном из четырёх направлений: вверх, вниз, влево или вправо. Если при этом змейка пересекает какую-либо границу экрана, то она появляется с противоположной стороны. Например, сделав шаг влево из первой клетки четвертой строки (будем говорить, что такая клетка имеет координаты $(4, 1)$) на поле размером 10 строк на 8 столбцов, змейка окажется в клетке с координатами $(4, 8)$. Иногда на поле появляется еда, которая может увеличить длину змейки, но это не имеет отношения к данной задаче.

По предложению совета директоров в Змейку было решено добавить режим парной игры. Изначально змейки двух игроков появляются в разных клетках и имеют длину 1, то есть состоят только из клеток, в которых они появились. Каждый ход состоит из того, что сначала перемещается змейка первого игрока, а затем змейка второго игрока. Правила перемещения аналогичны описанным выше для одиночной игры, но, в отличие от одиночного режима, змейки имеют возможность столкнуться друг с другом, после чего игра заканчивается. Маркетинговые исследования показали, что пользователи предпочитают как можно более длинные игры, поэтому перед разработчиками стоит задача исследовать, как зависит длительность игры от начальных координат змеек (a_1, b_1) и (a_2, b_2) . Так как в режиме парной игры пока реализована только возможность передвигаться, еда на поле не появляется, и фактически игра сводится к управлению двумя одноклеточными змейками.

Вам, как ведущему инженеру компании, поручено контролировать деятельность коллег. Для этого необходимо написать программу, которая по заданным размерам поля N и M и координатам двух различных клеток (a_1, b_1) и (a_2, b_2) вычисляет, на каком по счёту ходу впервые *может возникнуть* ситуация, что змейки столкнутся, то есть одна змейка совершит ход в клетку, в которой находится другая.

Формат входных данных

В первой строке ввода содержатся целые числа N и M ($1 \leq N, M \leq 1\,000\,000$), задающие соответственно количество строк и количество столбцов на игровом поле. Обратите внимание, что строки нумеруются от 1 до N снизу вверх, а столбцы — от 1 до M слева направо.

Во второй строке записаны два целых числа a_1 и b_1 ($1 \leq a_1 \leq N$, $1 \leq b_1 \leq M$) — начальные координаты змейки первого игрока: сначала номер строки, затем номер столбца.

В третьей строке аналогично задаются a_2 и b_2 — начальные координаты змейки второго игрока.

Гарантируется, что в начале игры змейки игроков находятся в разных клетках.

Формат выходных данных

Выведите номер первого хода, на котором змейки могут столкнуться.

Примеры

ВВОД	ВЫВОД
5 8 5 5 5 7	1
10 10 1 1 10 10	1
10 4 3 1 10 3	3

Замечание

В первом тесте существует единственная клетка в которой змейки могут встретиться через один ход — клетка с координатами (5, 6).

Система оценки

Тесты к этой задаче состоят из четырёх групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп.

Группа	Тесты	Баллы	Дополнительные ограничения	Комментарий
			N, M	
0	1–3	0	–	Тесты из условия.
1	4–16	30	$N, M \leq 4$	
2	17–49	30	$N, M \leq 100$	
3	50–69	40	–	Дополнительных ограничений нет.

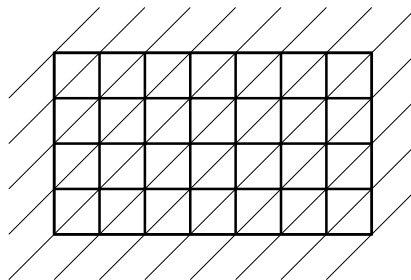
Задача Е. Плащ для Абсолема

Имя входного файла:	input.txt или стандартный поток ввода
Имя выходного файла:	output.txt или стандартный поток вывода
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Абсолему ещё далеко до превращения в бабочку, и он вкушает все прелести молодой жизни. К сожалению, во время мероприятия, связанного с неудачным выступлением на соревнованиях по программированию, Абсолем несколько потерял связь с реальностью и где-то забыл свой любимый плащ. Алиса предложила сшить опечаленному Абсолему такой же плащ, и тот на радостях поведал Алисе тайну волшебного узора.

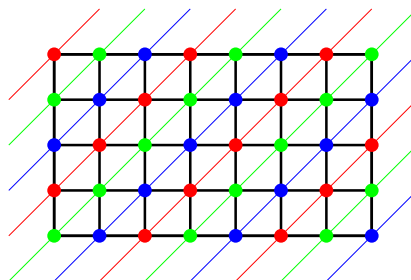
Плащ представляет из себя прямоугольник $N \times M$ с внутренней сеткой, узлы которой покрашены в K цветов особым образом:

- Плащ размещается так, чтобы два его противоположных угла лежали в точках с координатами $(0, 0)$ и (N, M) .
- Рассматриваются все прямые с угловым коэффициентом 1 (лежащие под углом 45° к оси абсцисс и направленные вверх и вправо), которые проходят хотя бы через один узел плаща. Эти прямые упорядочиваются слева направо (по тому, какая из них раньше (левее) пересекает ось абсцисс); самая левая прямая имеет номер ноль, а самая правая — $N + M$.



Плащ размером 7×4 .

- Точки, лежащие на i -й прямой, красятся в цвет номер $(i \bmod K) + 1$, то есть один плюс остаток от целочисленного деления i на K .



Плащ размером 7×4 , покрашенный в три цвета (цвет 1 — красный, цвет 2 — зелёный, цвет 3 — синий).

Шляпник, подслушав тайну узора плаща Абсолема, задумался, узлов какого цвета на плаще больше всего, а какого меньше всего. Помогите ему выяснить это!

Формат входных данных

В единственной строке содержатся три целых числа N , M и K — размеры плаща и количество цветов соответственно ($1 \leq N, M \leq 10^9$, $1 \leq K \leq N + M + 1$).

Формат выходных данных

В первой строке выведите два числа: номер цвета, в который будет покрашено *наибольшее* число узлов на плаще, и количество узлов, покрашенных в этот цвет.

Во второй строке выведите два числа: номер цвета, в который будет покрашено *наименьшее* число узлов на плаще, и количество узлов, покрашенных в этот цвет.

Если правильных ответов несколько, выведите любой из них.

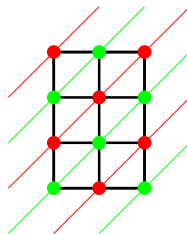
Примеры

ВВОД	ВЫВОД
7 4 3	2 14 3 13
2 3 2	1 6 2 6

Замечание

Плащ из первого примера изображён на рисунке из условия.

Во втором примере в каждый из двух цветов покрашено одинаковое число узлов плаща.



Плащ размером 2×3 , покрашенный в два цвета.

Система оценки

Тесты к этой задаче состоят из пяти групп. Баллы за каждую группу тестов ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп.

Группа	Тесты	Баллы	Дополнительные ограничения		Комментарий
			N, M	K	
0	1–2	0	–	–	Тесты из условия.
1	3–37	25	$N, M \leq 1000$	–	
2	38–55	25	$N, M \leq 1\,000\,000$	–	
3	56–68	25	–	$K \leq 1\,000\,000$	
4	69–91	25	–	–	Дополнительных ограничений нет.

Задача F. Анархия в Берляндии

Имя входного файла:	input.txt или стандартный поток ввода
Имя выходного файла:	output.txt или стандартный поток вывода
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

С незапамятных времен Берляндией правили добрые и справедливые цари, а власть передавалась по наследству. Но вот настали тяжёлые времена, по воле рока династия прервалась, и страна погрузилась в хаос. Некоторые города были захвачены враждующими друг с другом бандами головорезов, в других же городах какая-либо власть исчезла вовсе.

Всё началось с того, что в некоторых городах установилась власть преступных банд, причём под властью каждой из банд изначально оказалось ровно по одному городу. Каждая банда избрала в качестве столицы город своего первоначального зарождения. Начиная с того времени, каждый год *ровно одна* из банд решала расширить свои владения, захватывая некоторый город, до этого не бывший ни в чьём подчинении или находящийся под властью другой банды.

Для осуществления захвата вооружённый отряд данной банды выходил из своей столицы и двигался к выбранному городу, захватывая все города на своём пути. Также известно, что столицы всех банд были так серьёзно защищены, что через них невозможно было проехать (и уж тем более не могло идти речи об их захвате). Поэтому на пути от столицы банды-захватчика до выбранного города не могло быть столиц других банд.

Как уже неоднократно отмечалось различными историками, в описываемые времена в Берляндии было N городов, соединённых между собой $N - 1$ дорогой с двусторонним движением, при этом от каждого города можно было доехать до любого другого. Подобное устройство транспортной системы естественным образом налагало ограничения на возможные маршруты путешествия.

События эти происходили уже очень давно, а хроники тех лет были утрачены в вихре смутного времени. Теперь же к вам в руки попала карта Берляндии тех времён. Вы обнаружили, что на карте отмечены столицы каждой из банд, а также для каждого города указано, какая банда им владела на определённый момент времени (свободных городов к тому году уже не осталось).

Вычислите, при какой возможной последовательности захватов могло возникнуть описанное этой картой распределение контроля банд над городами, или же определите, что карта содержит ошибку, так как указанное на карте распределение власти банд над городами не могло быть получено с помощью последовательности описанных выше захватов. Более того, среди возможных ответов вам необходимо найти последовательность, состоящую из не более чем N захватов.

Формат входных данных

В первой строке входных данных через пробел указаны два целых числа N и M — количество городов на карте Берляндии и количество преступных банд ($1 \leq M \leq N \leq 300\,000$).

В следующих $N - 1$ строках описываются дороги Берляндии. Каждая строка содержит два различных целых числа a_i и b_i ($1 \leq a_i, b_i \leq N$), обозначающих, что между городами с номерами a_i и b_i существовала дорога с двусторонним движением.

В следующей строке входных данных через пробел указаны M различных целых чисел c_i ($1 \leq c_i \leq N$): согласно карте, в городе с номером c_i находилась столица банды i .

В последней строке входных данных находятся N разделённых пробелами целых чисел d_i , обозначающих, что город под номером i ($1 \leq i \leq N$) отмечен на карте как находящийся под властью банды с номером d_i ($1 \leq d_i \leq M$). Гарантируется, что $d_{c_i} = i$ для всех i .

Формат выходных данных

Если существует такая последовательность захватов, которая приводит к указанному на карте распределению контроля банд над городами, то в первой строке выведите слово «YES» (без кавычек). В следующей строке выведите единственное целое число S — длину найденной последовательности захватов (обратите внимание, должно выполняться неравенство $0 \leq S \leq N$). В следующих S строках выведите по два различных числа x_i и y_i — номера городов, являющихся столицей банды, захватывающей города в i -м году, и конечной целью захвата в i -м году соответственно.

Если же карта содержит ошибку и получить распределение контроля банд над городами, указанное на карте, невозможно, то в единственной строке вывода выведите слово «NO» (без кавычек).

Гарантируется, что если существует последовательность захватов, приводящая к указанному на карте распределению контроля банд над городами, то такое распределение контроля банд над городами всегда могло возникнуть не более чем за N лет.

Для удобства участников проверяющая программа не требует, чтобы на момент захвата города y_i он не принадлежал банде, владеющей городом x_i . Более того, разрешается выводить даже такие x_i и y_i , что все города на пути между ними принадлежат одной и той же банде. Обращаем ваше внимание, что это сделано исключительно для удобства участников. Все остальные требования к операциям захвата должны быть соблюдены.

Примеры

ВВОД	ВЫВОД
4 2 1 3 2 3 3 4 1 2 1 2 2 1	YES 2 1 4 2 3
5 3 1 3 2 3 3 4 3 5 1 2 3 1 2 3 1 3	NO

Замечание

Примером событий, приведших к картине, описанной в первом тесте, может служить следующая последовательность захватов:

- Первоначально в городах 1 и 2 образовалась по банде. Города 3 и 4 на этот момент никому не подвластны.
- В первый год первая банда отправилась захватывать город 4, по пути установив контроль над городом 3.
- Во второй год вторая банда отобрала контроль над городом 3 у первой банды.

Во втором примере не существует последовательности захватов, которая привела бы к описанной картине.

Система оценки

Тесты к этой задаче состоят из четырёх групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и **всех тестов предыдущих групп**. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Тесты	Баллы	Дополнительные ограничения		Комментарий
			N	M	
0	1–2	0	–	–	Тесты из условия.
1	3–30	30	$N \leq 10$	–	
2	31–60	30	$N \leq 1000$	–	
3	61–90	40	–	–	Offline-проверка.

Задача G. Честное столовластие

Имя входного файла:	input.txt или стандартный поток ввода
Имя выходного файла:	output.txt или стандартный поток вывода
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

— Это называется столовластием.
— Столовластием?
— Ну да. Все вопросы решаются
заполнением бумажек, которые
рассматривает специальная порода
людей — чиновники.

С. Лукьяненко «Недотёпа»

После разработки закона о введении налога на постройку зданий выше двух этажей у N чиновников короля Маркеля Весёлого осталось K бюджетных золотых монет. Эти деньги они решили частично поделить между собой. Поскольку на службе у Короля состоят лишь честные чиновники, делить их они решили справедливо. Они знают, какой вклад w_i внес в законопроект i -й чиновник. Деньги должны быть распределены так, что если i -й чиновник получил c_i монет, а j -й — c_j монет, то числа c_i и c_j должны иметь такой же результат сравнения, как числа w_i и w_j . Это значит, что $c_i < c_j$ должно быть верно тогда и только тогда, когда $w_i < w_j$. В частности, из этого следует, что при равенстве вкладов чиновники должны получить одинаковое количество монет.

Среди чиновников у короля Маркеля Весёлого есть некоторое количество фаворитов. Он хочет, чтобы фаворит с минимальным вкладом в законопроект получил в итоге как можно больше денег. Помогите честным госслужащим разделить деньги по справедливости, соблюдая требование короля Маркеля. Заметьте, что в задачу не входит максимизация прибыли чиновников, не являющихся фаворитами Маркеля Весёлого.

Формат входных данных

В первой строке следуют целое положительное число N и два целых неотрицательных числа M и K , разделённых пробелами — количество чиновников, количество фаворитов Маркеля и количество монет для раздела соответственно ($M \leq N \leq 10^5$, $K \leq 10^{18}$).

Во второй строке находятся N целых положительных чисел w_i , разделённых пробелами ($1 \leq w_i \leq 10^9$). i -е из них обозначает вклад, который привнес i -й чиновник.

В третьей строке записаны M различных натуральных чисел f_i — номера чиновников, являющихся фаворитами короля ($1 \leq f_i \leq N$).

Формат выходных данных

Если справедливое распределение монет существует, первой строкой выведите слово «POSSIBLE» (без кавычек). Второй же строкой выведите N целых неотрицательных чисел через пробел. i -е из них должно равняться количеству монет, которые получит i -й чиновник.

Помните, что раздавать все монеты не обязательно, а Маркель Весёлый волнуется только о том, чтобы фаворит с минимальным вкладом (а значит и минимальным количеством монет) получил как можно больше, а в остальном чиновники вольны распределять монеты как угодно. Это значит, что и вы можете вывести любой правильный ответ.

Если же поделить деньги справедливо невозможно, выведите слово «IMPOSSIBLE» (без кавычек).

Примеры

ВВОД	ВЫВОД
5 2 15 15 2 33 12 8 3 4	POSSIBLE 4 0 5 3 1
4 2 3 1 2 3 2 1 2	IMPOSSIBLE

Замечание

Обратите внимание, что некоторая часть денег может остаться неподелённой. В первом примере входных данных излишек монет можно было бы также раздать второму и пятому чиновнику, но, так как они не являются фаворитами короля, делать это не обязательно.

Во втором примере входных данных корректный делёж потребует не менее четырёх монет, тогда как их осталось всего три.

Система оценки

Тесты к этой задаче состоят из четырёх групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы и **всех тестов предыдущих групп**.

Группа	Тесты	Баллы	Дополнительные ограничения		Комментарий
			N	K	
0	1–2	0	–	–	Тесты из условия.
1	3–14	25	$N \leq 10$	$K \leq 50$	
2	15–32	25	$N \leq 5000$	$K \leq 5000$	
3	33–56	50	–	–	

Задача Н. План бегства

Имя входного файла:	input.txt или стандартный поток ввода
Имя выходного файла:	output.txt или стандартный поток вывода
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Из офиса вашей энергетической корпорации были выкрадены чертежи инновационной электростанции. Будучи построенной, она решила бы большинство проблем современной энергетики, а вас сделала бы невероятно богатым. Не торопитесь отчаиваться: службе безопасности удалось выследить похитителя и выяснить, что сейчас чертежи хранятся в принадлежащем конкурентам подземном бункере. Лучший из ваших агентов уже отправился туда с целью выкрасть чертежи обратно, но миссия очень трудна!

Так как в системе конкурентов до сих пор присутствует уязвимость Heartbleed, ваш IT-отдел получил ограниченный доступ к системе безопасности бункера и обнаружил план помещений. Оказалось, что бункер состоит из N комнат и M соединяющих их коридоров. По каждому коридору можно ходить в обоих направлениях. Никакой коридор не соединяет комнату саму с собой, никакая пара комнат не соединена более чем одним коридором (действительно, кому может прийти в голову организовать коридоры иначе?).

В некоторых комнатах расположены выходы из бункера. Как только ваш агент возьмёт чертежи, в комнате, где он находится, сработает сигнализация. После этого все выходы закроются, и агент окажется в ловушке.

Закалённая участием в CTF, ваша IT-команда смогла изменить настройки системы безопасности таким образом, что при срабатывании в комнате i сигнализации закроются не все выходы из бункера, а только K ближайших к данной комнате. Расстояние между двумя комнатами определяется как минимальное количество коридоров, по которым надо пройти, чтобы добраться из одной комнаты в другую. Расположенные на одинаковом расстоянии от комнаты i выходы система будет закрывать в порядке возрастания номеров содержащих их комнат.

Когда сработает сигнализация и поднимется тревога, агенту необходимо как можно быстрее покинуть бункер. Поскольку расположение чертежей заранее неизвестно, то необходимо для каждой комнаты бункера вычислить, где будет расположен ближайший незапертый выход, если в этой комнате сработает сигнализация. Из незапертых выходов на одинаковом расстоянии от данной комнаты следует определить выход, расположенный в комнате с наименьшим номером.

Формат входных данных

В первой строке ввода записаны через пробел два целых числа N и M , обозначающие количество комнат и коридоров в бункере ($1 \leq N \leq 100\,000, 0 \leq M \leq 100\,000$).

Следующие M строк содержат по два числа a_i и b_i — номера комнат, соединённых i -м коридором ($1 \leq a_i, b_i \leq N$).

Далее следует число A — количество комнат с выходами из бункера ($1 \leq A \leq N$).

В следующей строке содержатся A разделённых пробелами различных чисел e_i , обозначающих номера комнат, в которых расположены выходы ($1 \leq e_i \leq N$).

В последней строке ввода содержится целое число K ($0 \leq K \leq 10$) — количество выходов, закрывающихся при срабатывании сигнализации.

Формат выходных данных

Выведите N чисел, i -е из которых является номером ближайшей комнаты с открытым выходом после срабатывания сигнализации в комнате с номером i . Если для некоторого i подходящих комнат несколько, то выберите среди них комнату с наименьшим номером. Если же из комнаты i эвакуироваться не удастся, то выведите -1 .

Примеры

ВВОД	ВЫВОД
5 5 1 2 2 3 3 4 4 5 5 1 2 1 3 0	1 1 3 3 1
9 10 1 2 2 3 3 4 4 5 6 7 6 8 6 9 7 8 7 9 8 9 7 1 2 3 4 5 6 7 2	3 3 4 5 3 -1 -1 -1 -1

Замечание

В первом примере $K = 0$, а значит необходимо просто найти ближайший выход для каждой комнаты. Ответ определяется однозначно для всех комнат, кроме 2, — выходы в комнатах 1 и 3 находятся на расстоянии одного перехода по коридору от неё, среди них требуется выбрать комнату с меньшим номером, то есть 1.

Во втором примере из комнат с номерами от 6 до 9 выбраться невозможно, так как из них достижимы только два выхода, которые оба закроются в случае срабатывания сигнализации.

Система оценки

Тесты к этой задаче состоят из четырёх групп. Баллы за каждую группу тестов ставятся только при прохождении всех тестов группы и **всех тестов предыдущих групп**, в том числе **тестов из условия**. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Тесты	Баллы	Дополнительные ограничения	Комментарий
0	1–2	0	–	Тесты из условия.
1	3–26	30	$K = 0$	
2	27–44	30	$K \leq 1$	
3	45–64	40	–	Offline-проверка.

Задача I. Домашнее задание

Имя входного файла:	input.txt или стандартный поток ввода
Имя выходного файла:	output.txt или стандартный поток вывода
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт

Спустя тридцать сезонов Мэгги Симпсон наконец поступила в первый класс. В целях тренировки чистописания учительница задала детям выписать в строку N цифр от '0' до '9' в некотором порядке.

После нескольких часов страданий Мэгги всё-таки справилась с домашним заданием и убежала играть с друзьями, оставив свою работу лежащей на письменном столе. В это время в комнату зашла Лиза Симпсон. Игра на саксофоне уже успела её порядком утомить, и она находилась в поисках какого-нибудь нового увлекательного занятия, когда её взгляд остановился на исписанной тетрадке Мэгги.

Лиза быстро справилась с тем, чтобы посчитать сумму всех написанных Мэгги цифр, но тут в комнату ворвался Барт и предложил ей задачу поинтереснее — произвести M операций с данной строкой цифр. Фантазия Барта довольно скудна, поэтому операции бывают только двух типов. Пусть позиции в строке занумерованы числами от 1 до N слева направо. Тогда операции имеют следующий вид:

1. Барт называет два числа l_i и r_i , а также цифру c_i , после чего Лиза должна заменить все цифры на позициях с l_i -й по r_i -ю включительно на цифру c_i .
2. Барт называет два числа l_i и r_i , после чего Лиза должна посчитать сумму всех чисел, десятичные записи которых встречаются как подстроки отрезка исходной строки с l_i -й по r_i -ю позицию включительно. При этом каждое число должно быть просуммировано с **учётом кратности**, то есть столько раз, сколько его запись встречается на отрезке с l_i -й по r_i -ю. Также требуется учитывать только корректные десятичные записи чисел, то есть подстроки, начинающиеся на '0', **не должны** входить в сумму.

Барт сегодня добрый, поэтому на операцию второго типа он просит от Лизы только остаток от деления ответа на $10^9 + 7$.

Желая доказать своё превосходство над братом, Лиза немедленно согласилась. Поскольку Барт сам не знает правильных ответов, он никак не может проверить вычисления Лизы, поэтому ему крайне необходима ваша помощь.

Формат входных данных

В первой строке ввода записаны через пробел два целых числа N и M — количество цифр, выписанных Мэгги, и количество операций Барта ($1 \leq N, M \leq 100\,000$).

Вторая строка ввода содержит N цифр от 0 до 9, разделённых пробелами.

Следующие M строк описывают операции.

Запись вида 1 l_i r_i c_i обозначает операцию первого типа ($1 \leq l_i \leq r_i \leq N$, $0 \leq c_i \leq 9$).

Запись вида 2 l_i r_i обозначает операцию второго типа ($1 \leq l_i \leq r_i \leq N$).

Формат выходных данных

На каждую операцию второго типа выведите остаток от деления ответа на $10^9 + 7$.

Если среди операций Барта не присутствует ни одной операции второго типа, ничего выводить не требуется.

Примеры

ВВОД	ВЫВОД
4 3 1 2 1 3 2 1 3 1 3 3 0 2 2 4	158 228
9 3 1 1 1 1 1 1 1 1 2 1 9 1 1 9 9 2 1 9	137174205 234567838

Замечание

На первую операцию второго типа ответ представляет собой следующую сумму:
 $1 + 12 + 121 + 2 + 21 + 1 = 158$.

На вторую операцию второго типа ответ представляет собой следующую сумму:
 $2 + 20 + 203 + 0 + 3 = 228$. Обратите внимание: в ней присутствует меньше слагаемых, так как строка 03 не представляет собой корректную десятичную запись целого числа.

Система оценки

Тесты к этой задаче состоят из пяти групп. Баллы за каждую группу тестов ставятся только при прохождении всех тестов группы и **всех тестов предыдущих групп**. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Тесты	Баллы	Дополнительные ограничения	Комментарий
0	1–2	0	–	Тесты из условия.
1	3–11	20	$N, M \leq 50$	
2	12–20	20	$N, M \leq 400$	
3	21–29	20	$N, M \leq 2000$	
4	30–38	40	–	Offline-проверка.

Задача J. Обгон запрещён

Имя входного файла:	input.txt или стандартный поток ввода
Имя выходного файла:	output.txt или стандартный поток вывода
Ограничение по времени:	4 секунды
Ограничение по памяти:	256 мегабайт



Запрещается обгон всех транспортных средств, кроме тихоходных транспортных средств, гужевых повозок, мопедов и двухколесных мотоциклов без коляски.

Сегодня вечером супергерой Ватман планирует навестить свою любимую бабушку. Так как он вовремя не продлил лицензию на полёты, ему придётся ехать на ватмобиле сквозь Готэмские пробки. Необходимо закончить дела и добраться до бабушки как можно быстрее, ведь пирожки уже остывают!

Часть пути Ватману предстоит проехать по дороге, имеющей лишь одну полосу в каждом из двух возможных направлений движения. В рамках данной задачи дорогу можно считать бесконечной прямой. На обеих полосах движения в совокупности находятся N автомобилей, про каждый из которых известно его начальное положение p_i и его постоянная скорость v_i . Таким образом, координата i -го автомобиля в момент времени t будет равняться $x_i(t) = p_i + v_i \cdot t$. Однако, данная формула справедлива только до тех пор, пока этот автомобиль не догонит другой, движущийся в том же направлении (а значит, и по той же полосе), но медленнее. Так как обгон на данном участке дороги запрещён, то, догнав более медленный автомобиль, более быстрый вынужден снизить скорость и двигаться сразу за ним всё оставшееся время.

Ватман эгоистично полагает, что достаточно часто думает о судьбах простых людей, чтобы позволить себе пренебречь некоторыми правилами дорожного движения. Сегодня вечером он планирует обгонять другие автомобили по встречной полосе, но, как и любой нормальный супергерой, Ватман стремится избегать аварийных ситуаций. В связи с этим он совершает обгон, только если ближайший двигающийся навстречу автомобиль находится на безопасном расстоянии, обозначаемом в данной задаче как d .

Планы Ватмана постоянно меняются, он пока не знает, когда именно поедет в гости к бабушке, какой маршрут выберет, на каком из многочисленных ватмобилей поедет и какую дистанцию для обгона будет считать безопасной. Поэтому он попросил вас рассчитать минимальное время поездки по данной дороге для Q различных значений параметров задачи.

Длиной автомобилей следует пренебречь, то есть можно считать их точками на прямой. Также следует считать, что как только более быстрая машина догоняет более медленную, она движется за ней на расстоянии 0. Ватман может совершить обгон, если на встречной полосе в направлении движения ватмобиля отсутствуют встречные машины на расстоянии как минимум d . Машины, находящиеся на встречной полосе и имеющие такую же координату как и ватмобиль, помехой для обгона не являются. Начав обгон, Ватман совершает его мгновенно и тут же возвращается на исходную полосу.

Поскольку любые вычисления, связанные с вещественными числами, сопряжены с неизбежной погрешностью, жюри гарантирует вам, что Ватману никогда не придётся принимать решение об обгоне, основываясь на сравнении очень близких друг к другу чисел. Это значит следующее: изменение d на $0 \leq eps \leq 0.001$ в любую сторону не повлияет на правильный ответ к задаче.

Формат входных данных

В первой строке ввода содержатся два целых числа N и Q — количество автомобилей на дороге и запросов Ватмана соответственно ($0 \leq N \leq 10\,000$, $1 \leq Q \leq 5\,000$).

В каждой из следующих N строк содержатся по два целых числа p_i и v_i , описывающих начальное положение и начальную скорость i -го автомобиля соответственно ($|p_i| \leq 100\,000$, $|v_i| \leq 100\,000$, $v_i \neq 0$). Машины с положительным значением v_i двигаются по одной полосе, а с отрицательным — по другой. Гарантируется, что все p_i различны.

Далее следуют Q строк с описаниями запросов. Каждый из них задаётся пятью целыми числами t_i, a_i, b_i, v_i, d_i — соответственно, момент времени, в который Ватман планирует выехать на дорогу, начальная и конечная точки его маршрута, максимальная скорость ватмобиля и минимальная дистанция до встречного автомобиля, которую Ватман считает безопасной для обгона ($|a_i|, |b_i| \leq 100\,000$, $0 < |v_i| < 100\,000$, $0 \leq t_i, d_i \leq 100\,000$). Гарантируется, что $a_i \neq b_i$, и что в момент времени t_i никакая машина, кроме ватмобиля, не имеет координату a_i . Также гарантируется, что $v_i > 0$ тогда и только тогда, когда $b_i > a_i$.

Формат выходных данных

Для каждого запроса выведите в отдельной строке одно число — минимальное время, за которое Ватман может успеть добраться из начальной точки в конечную. Ваш ответ будет считаться правильным, если его абсолютная или относительная погрешность не будет превосходить 10^{-6} .

Более формально, пусть ваш ответ равен A , а ответ жюри — B . Проверяющая программа будет считать ваш ответ правильным, если $\frac{|A-B|}{\max(1, B)} \leq 10^{-6}$.

Примеры

ВВОД	ВЫВОД
1 1 10 10 0 0 100 20 10	5.0000000000
2 1 20 5 200 -10 4 10 100 10 40	10.0000000000
6 3 20 1 10 10 5 100 80 -1 90 -10 95 -100 0 0 100 10 1 0 0 100 10 100 0 100 0 -10 100	10.0000000000 35.0000000000 35.0000000000

Замечание

В первом примере никто не мешает Ватману совершить обгон, и он доедет за минимально возможное для своей максимальной скорости время.

Во втором примере Ватман догонит первую машину в точке с координатой 70, в этот момент вторая машина будет иметь координату 100, то есть обгонять будет уже слишком поздно. Ватману придётся две секунды ехать медленнее, пока вторая машина не проедет мимо.

Система оценки

Тесты к этой задаче состоят из пяти групп. Баллы за каждую группу тестов ставятся только при прохождении всех тестов группы и всех тестов **предыдущих** групп.

Группа	Тесты	Баллы	Дополнительные ограничения		Комментарий
			N	Q	
0	1–3	0	–	–	Тесты из условия.
1	4–31	30	$N \leq 100$	$Q \leq 100$	
2	32–51	30	–	$Q \leq 100$	
3	52–81	40	–	–	Дополнительных ограничений нет.

Задача К. Программируемая змейка

Имя входного файла:	input.txt или стандартный поток ввода
Имя выходного файла:	output.txt или стандартный поток вывода
Ограничение по времени:	3 секунды
Ограничение по памяти:	256 мегабайт

Компания Konia, специализирующаяся на выпуске бюджетных мобильных устройств, в последнее время теряет свои позиции на рынке. Ради спасения компании совет директоров пошёл на отчаянный шаг: было решено внести изменения в самую известную игру, которая доступна на всех устройствах от Konia — «Змейку». Да, снова.

Игра происходит на клетчатом поле, состоящем из H строк и W столбцов. Будем нумеровать строки сверху вниз от 1 до H , а столбцы слева направо от 1 до W . Таким образом, каждая клетка A поля однозначно задаётся парой чисел (A_r, A_c) , где A_r — номер строки, а A_c — номер столбца. Две клетки A и B называются соседними, если они соседние по стороне, то есть $|A_r - B_r| + |A_c - B_c| = 1$. Помимо этого, соседними являются клетки верхней и нижней строки, а также крайнего левого и крайнего правого столбца, то есть все пары клеток вида $(1, i) - (H, i)$, где i пробегает от 1 до W , и все пары вида $(i, 1) - (i, W)$, где i пробегает от 1 до H .

На поле находится змейка, представляющая собой последовательность клеток $A_1, A_2, \dots, A_L$, где L — целое положительное число, означающее её длину. Каждая клетка должна встречаться в данной последовательности не более одного раза, при этом любые две клетки, соседние в последовательности, должны являться соседними клетками поля. Несложно заметить, что максимально допустимая длина змейки равняется $H \cdot W$. Клетка A_1 называется головой змейки.

Игрок управляет только перемещением головы змейки. Для этого он может использовать четыре команды:

- ‘L’ — перемещает голову змейки влево, то есть уменьшает на единицу номер столбца, в котором она находится. Если перед использованием команды голова находится в крайнем левом столбце, то она переносится в крайний правый столбец, оставаясь при этом в той же строке.
- ‘R’ — перемещает голову змейки вправо, то есть увеличивает на единицу номер столбца. Аналогично предыдущей команде, из крайнего правого столбца голова переносится в крайний левый.
- ‘U’ — перемещает голову змейки вверх, то есть уменьшает на единицу номер строки, в которой она находится. Граница обрабатывается так же, как и в предыдущих командах, то есть применение в верхней строке переносит голову змейки в нижнюю строку, не меняя столбец.
- ‘D’ — перемещает голову змейки вниз, то есть увеличивает на единицу номер строки. Соответственно, из нижней строки голова змейки переносится в верхнюю.

Как только игрок использует команду, все клетки, образующие змейку, перемещаются одновременно, при этом голова сдвигается по описанным выше правилам, а все остальные клетки «подтягиваются» за ней, то есть для всех i от 2 до L клетка A_i заменяется на бывшее значение клетки A_{i-1} . Например, вторая клетка змейки (если она существует, то есть $L \geq 2$) перемещается туда, где находилась голова, третья клетка (если $L \geq 3$) перемещается туда, где находилась вторая, и так далее.

Игра заканчивается, как только змейка врежется сама в себя, то есть на очередном шаге перестанет выполняться условие уникальности всех клеток. Это может произойти из-за того, что голова переместится в клетку, занимаемую какой-то другой частью змейки. В частности, если змейка состоит хотя бы из трёх клеток, то команда перемещения головы в сторону клетки A_2 приведёт к столкновению, так как в новом положении змейки совпадут клетки A_1 и A_3 .

Также в классическом варианте игры присутствует возможность кушать яблоки и увеличивать длину своей змейки, но в рамках данной задачи эта особенность игры не рассматривается.

В новой модификации данной игры от компании Konia игроку больше не требуется регулярно нажимать на кнопки, он должен лишь изначально разместить свою змейку на игровом поле и написать программу действий длины K , то есть задать последовательность из K команд перемещения головы змейки. После запуска змейка будет последовательно выполнять команды, пока не врежется сама в себя. Если змейка успешно выполнит все K команд, то она начинает выполнять их заново в той же последовательности, и так до тех пор, пока не произойдет столкновение или игроку не надоест следить за процессом.

Вам, как ведущему инженеру компании Konia по программированию змеек, поручено решить следующую задачу: для данных **различных простых** чисел H и W , определяющих размеры поля, а также данной программы игры, определить, змейку какой максимальной длины можно разместить на игровом поле так, что она никогда не врежется сама в себя после запуска игры. Руководство так и не смогло объяснить вам, почему H и W являются простыми числами, но напомнило, что число является простым, если у него ровно два различных целых положительных делителя.

Формат входных данных

В первой строке ввода содержится три целых числа H , W и K — размеры поля и длина программы ($2 \leq H, W \leq 10^9 + 9$, $H \neq W$, $1 \leq K \leq \min\{100\,000, H \cdot W\}$).

Обратите внимание: гарантируется, что H и W — **различные простые** числа.

Во второй строке подряд записаны K символов, задающих последовательность команд. Никакие символы, кроме 'L', 'R', 'U' и 'D', в данной строке не встречаются.

Формат выходных данных

Выведите одно число — максимальную длину змейки, которая сможет неограниченно долго перемещаться в соответствии с данной последовательностью команд, ни разу не врезавшись в саму себя.

Примеры

ВВОД	ВЫВОД
5 7 4 LURD	4
11 2 2 RD	21
5 13 4 ULUD	2

Замечание

В первом тесте после выполнения всех четырёх команд голова змейки оказывается в своём стартовом положении, поэтому змейка длиннее четырёх клеток не проживёт больше четырёх ходов.

Во втором тесте правильно расположенная змейка может занять почти всё поле.

Система оценки

Тесты к этой задаче состоят из пяти групп. Баллы за каждую группу ставятся только при прохождении всех тестов группы. Группы не являются вложенными, то есть ваше решение будет тестироваться на всех группах, если оно пройдет тесты из условия. **Offline-проверка** означает, что результаты тестирования вашего решения на данной группе станут доступны только после окончания соревнования.

Группа	Тесты	Баллы	Дополнительные ограничения		Комментарий
			H, W	K	
0	1–3	0	–	–	Тесты из условия.
1	4–15	15	$H, W \leq 100$	–	
2	16–28	30	$H, W \leq 2000$	–	
3	29–40	30	–	$K \leq 600$	Offline-проверка.
4	41–51	25	–	–	Offline-проверка.