

Problem A. Анализ судейских записок

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 512 mebibytes

В профессиональном боксе счёт определяется судейскими записками. В каждом из 12 раундов судья ставит 10 баллов победителю раунда и 9 баллов проигравшему в этом раунде. Счёт боксёра после N раундов определяется как сумма набранных им баллов.

Если раунд заканчивается нокаутом, то бой останавливается и победитель объявляется досрочно.

Вам даны два целых числа — счёт на записках судьи по итогам боя для боксёра в синем углу и для боксёра в красном углу.

Вычислите, сколько раундов продлился бой.

Input

Первая строка входных данных содержит одно целое число — количество очков у боксёра в синем углу на момент остановки боя ($0 \leq b \leq 120$). Вторая строка содержит одно целое число — количество очков у боксёра в красном углу на момент остановки боя ($0 \leq r \leq 120$). Гарантируется, что судья ставил за каждый раунд либо 9:10, либо 10:9.

Output

Выведите одно целое число — количество раундов в бою.

Examples

standard input	standard output
10 9	1
120 108	12
19 19	2

Problem B. БиДжет

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 512 mebibytes

Компания Honey Paskard разработала линейку струйных принтеров «БиДжет», картриджи для которых являются модульными и представляет собой своеобразные соты, собранные из ячеек, в сечении являющихся правильными шестиугольниками. Перезаправка картриджа может производиться через любую из сот, при этом чернила свободно просачиваются через внутренние перегородки между отдельными ячейками. Одна ячейка вмещает один миллилитр чернил.

К сожалению, если картридж хранился слишком долго, то в каком-то количестве ячеек чернила засыхают; такая ячейка не может быть заполнена чернилами и не участвует в процессе просачивания. Так что возможно, что заполнять картридж чернилами придётся в несколько действий через разные ячейки.

У Байтазара есть N миллилитров чернил и старый картридж, в некоторых ячейках которого чернила засохли. Байтазар хочет узнать, за какое минимальное количество действий он зальёт чернила в картридж? Гарантируется, что конфигурация картриджа и засохших ячеек подобрана таким образом, что залить N миллилитров возможно.

Input

Первая строка входных данных содержит три целых числа N — количество чернил, а также h и v — размерность шестиугольной сетки картриджа ($1 \leq N \leq 10^6$, $1 \leq h, v \leq 10^3$).

Каждая из последующих h строк содержит по v разделённых пробелом символов. Ячейке с засохшими чернилами соответствует символ '#', пустой ячейке — символ '.'. При этом каждая чётная строка начинается с пробела.

Output

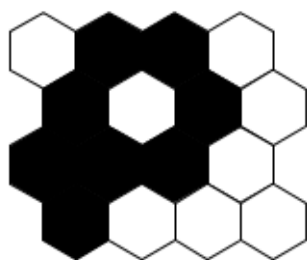
Выведите одно целое число — минимальное количество действий, которое потребуется Байтазару, чтобы залить N грамм чернил в принтер.

Example

standard input	standard output
<pre>8 4 4 . # # . # . # . # # # . # . . .</pre>	3
<pre>6 3 6 . . # # # # . # # # # . # #</pre>	2

Note

Конфигурации из примеров к задаче:



Problem C. Выбор нумерации

Input file: стандартный ввод
Output file: стандартный вывод
Time limit: 1 секунда
Memory limit: 512 мегабайт

Это задача с двойным запуском

Шахматы Фишера — вариант шахмат, предложенный Робертом Фишером, 11-м чемпионом мира по шахматам. Правила игры в основном такие же, как в классических шахматах, но начальная расстановка фигур определяется **случайным образом** со следующими ограничениями:

- Белые и чёрные фигуры располагаются соответственно на первом и восьмом рядах, расстановка фигур белых идентична расстановке фигур чёрных (то есть соответствующие фигуры стоят на одинаковых вертикалях).
- Белые и чёрные пешки располагаются соответственно на втором и седьмом рядах.
- У каждой из сторон слоны стоят на полях разного цвета (то есть один на белом, один на чёрном).
- Ладьи каждой из сторон располагаются по разные стороны от короля. Это необходимо для того, чтобы обеспечить возможность рокировки.
- Поля на рядах с третьего по шестой включительно пустые.

В данной задаче будут использованы стандартные обозначения:

Белые фигуры обозначаются заглавными буквами. Среди них — один король (K), один ферзь (Q), два коня (N), две ладьи (R) и два слона (B). Располагаются на самом нижнем (первом) ряду. Восемь белых пешек (P) располагаются на втором снизу (втором) ряду.

Чёрные фигуры обозначаются строчными буквами. Среди них — один король (k), один ферзь (q), два коня (n), две ладьи (r) и два слона (b). Располагаются на самом верхнем (восьмом) ряду. Восемь чёрных пешек (p) располагаются на втором сверху (седьмом) ряду.

Пустые поля обозначаются точками ..

Позиция задаётся 8 строками, каждая из которых имеет длину 8. Поле $h1$ соответствует первому символу в первой строке, поле $a1$ соответствует первому символу в последней строке.

Пример корректной начальной позиции шахмат Фишера, заданной в описанном выше формате:

```
bnrbnkrq
pppppppp
.....
.....
.....
.....
PPPPPPPP
BNRBNKRQ
```

Другое название шахмат Фишера — шахматы-960, по числу начальных позиций. Ваша задача — по заданному числу от 1 до 960 сгенерировать позицию из шахмат Фишера так, что разным числам соответствуют разные позиции, а затем по позиции в соответствии с вашей нумерацией восстановить номер. Ваше решение будет запущено на каждом тесте независимо два раза — один раз для кодирования, второй раз для декодирования.

Input

В первой строке входных данных содержится слово «**encode**», если от вашей программы требуется по числу вывести позицию, и «**decode**», если требуется по позиции вывести её номер. Вторая строка содержит одно целое число T — количество тестовых примеров ($1 \leq T \leq 32$).

Далее следуют тестовые примеры. Если первая строка входных данных содержит слово «**encode**», то каждый тестовый пример занимает одну строку и состоит из целого числа x_i ($1 \leq x_i \leq 960$). Если первая строка входных данных содержит слово «**decode**», то каждый тестовый пример занимает 8 строк и представляет собой корректную начальную позицию в шахматах Фишера, заданную в формате, описанном в условиях задачи.

Output

В случае **encode** (при первом запуске) для каждого числа выведите корректную начальную позицию в шахматах Фишера, соответствующую заданному числу в вашей нумерации. Если позиция не будет корректной, то решение будет признано неверным.

В случае **decode** (при втором запуске) для каждой позиции выведите номер этой позиции в вашей нумерации. Если номер не будет совпадать с соответствующим номером из первого запуска, решение будет признано неверным.

Examples

стандартный ввод	стандартный вывод
<code>encode 1 1</code>	<code>rnbqkbnr pppppppp PPPPPPPP RNBQKBNR</code>
<code>decode 1 rnbqkbnr pppppppp PPPPPPPP RNBQKBNR</code>	<code>1</code>

Problem D. Главное — симметрия

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 512 mebibytes

Байтика — владелец крупной девелоперской компании. Байтика хочет выбрать несколько участков для строительства новых офисных центров. При этом участки должны быть выбраны так, чтобы в приложении Vytex.Maps построенные здания образовывали невырожденный правильный многоугольник. По координатам доступных на рынке участков (обозначаемых на картах точками) определите, сколько существует различных способов выбора участков для постройки офисов.

Input

Первая строка входных данных содержит одно целое число n — количество участков ($1 \leq n \leq 500$). Каждая из последующих n строк содержит по два целых числа x_i и y_i ($-100 \leq x_i, y_i \leq 100$) — координаты очередного участка. Гарантируется, что никакие два участка не совпадают.

Output

Выведите одно целое число — количество различных способов выбрать участки так, чтобы они образовывали правильный многоугольник.

Example

standard input	standard output
5 1 0 0 0 0 1 1 1 3 3	1

Problem E. Движение без пробок

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 512 mebibytes

В рамках проекта «Город без пробок» разрабатывается приложение, позволяющее водителям выбирать оптимальную скорость движения для того, чтобы избежать пробок.

В процессе работы над этим приложением сбор данных о работе светофоров производился следующим образом.

Пусть на перекрёстке установлен светофор, работающий по следующим правилам: если в момент времени t сигнал светофора меняется с красного на зелёный, то в момент времени $t + g$ светофор переключится с зелёного на жёлтый (то есть зелёный свет будет гореть с момента времени t до момента времени $t + g$, исключая момент времени $t + g$), в момент времени $t + g + y$ переключится на красный (то есть жёлтый будет гореть с момента времени $t + g$ до момента времени $t + g + y$, исключая последний), а в момент времени $t + g + y + r$ переключится с красного на зелёный (то есть красный будет гореть с момента времени $t + g + y$ до момента времени $t + g + y + r$, исключая последний).

Установленный на автомобиле датчик распознавания сигнала светофора при пересечении перекрёстка записывает момент времени, в который автомобиль пересекает перекрёсток, и цвет сигнала светофора в этот момент.

По собранным данным требуется выяснить, с какой вероятностью в заданный момент времени T на светофоре будет сигнал заданного цвета c .

Input

Первая строка входных данных содержит три целых числа g , y и r — длительность зелёного, жёлтого и красного сигнала соответственно ($0 \leq g, y, r \leq 10^8$).

Вторая строка входных данных содержит одно целое число n — количество записей, собранных датчиком ($1 \leq n \leq 1000$).

Каждая из последующих n строк содержит одну запись. Запись состоит из двух полей: момента времени t ($0 \leq t \leq 10^9$) и одной буквы, задающей сигнал светофора ('g' для зелёного, 'y' для жёлтого и 'r' для красного).

Последняя строка содержит в аналогичном формате момент времени T и цвет сигнала светофора c , для которых делается запрос.

Output

Выведите одно число — вероятность того, что в момент времени T на светофоре будет сигнал c с абсолютной или относительной погрешностью 10^{-3} .

Example

standard input	standard output
4 4 4 3 2 g 18 y 34 r 5 g	0.25
4 4 4 4 2 g 6 y 10 r 14 g 4 r	0
6 6 6 6 5 g 6 g 9 y 12 y 15 r 19 r 7 g	1

Problem F. Если цель неизвестна...

Input file: *standard input*
Output file: *standard output*
Time limit: 1 second
Memory limit: 512 mebibytes

Это — интерактивная задача

Недавно был предложен новый вариант соревнований по стрельбе из лука — стрельба с неизвестным «яблочком».

Мишень состоит из N целей, расположенных по кругу. Соревнование состоит из Q раундов.

Перед началом первого раунда стрелок может соединить некоторые пары целей проводами, при этом никакие две цели не могут быть соединены напрямую более, чем одним проводом, и никакая цель не может быть соединена сама с собой. Соединение остаётся неизменным в течение всех раундов.

В начале каждого раунда одна из целей назначается основной, остальные являются вспомогательными. Стрелок не знает, какая именно цель выбрана. Задача стрелка — не более, чем за 10 выстрелов поразить основную цель. Если стрелок попадает в основную цель, на табло над стендом загорается зелёный цвет («**green**») и раунд заканчивается. Если стрелок попадает во вспомогательную цель, которая соединена с основной целью **напрямую** проводом, загорается жёлтый цвет («**yellow**»). Иначе загорается красный цвет («**red**»).

Ваша задача — написать программу, которая соединяет мишени так, что сможет поразить цель за не более, чем 10 выстрелов в каждом из Q раундов.

Interaction Protocol

Взаимодействие начинает программа жюри, выводя два целых числа N и Q — количество мишеней и раундов, соответственно ($1 \leq N, Q \leq 200$).

Далее ваша программа выводит информацию о соединяемых проводами целях. Сначала выведите строку с одним целым числом K — количество проводов. Затем выведите K строк, каждая из которых содержит по два целых числа от 1 до N — номера целей, которые соединяет очередной провод. Нельзя соединять две цели напрямую более, чем одним проводом, также нельзя соединять цель саму с собой.

После этого ваша программа делает выстрел, выводя целое число от 1 до N — номер очередной мишени. Программа жюри выдаёт в ответ **green**, если выстрел поразил основную цель (после чего вы уже делаете первый выстрел следующего раунда, или заканчиваете выполнение программы, если раунд был последним), **yellow**, если выстрел поразил цель, соединённую с основной напрямую, и **red** в противном случае. После любого из этих двух ответов ваша программа продолжает стрельбу, выбирая очередную цель. Если в каком-то раунде потрачено больше 10 попыток, решение признаётся неверным.

Note

Не забывайте после каждого хода выводить перевод строки и сбрасывать буфер вывода вызовом функции **flush** используемого Вами языка программирования. В противном случае попытка может получить превышение лимита реального времени (Wall Time Limit Exceed).

Example

standard input	standard output
3 2	2
	1 2
	1 3
	1
green	
	1
yellow	
	2
red	
	3
green	

Problem G. Ёрмунгандский язык

Input file: *standard input*
Output file: *standard output*
Time limit: 2 секунды
Memory limit: 512 мегабайт

Институт Мифических Языков (ИМЯ) работает над автоматическим переводчиком с ёрмунгандского языка. Как известно, в ёрмунгандском языке слова имеют следующую структуру. Для начала определяются два элемента — префикс A и суффикс B . Элементы записываются строчными латинскими буквами.

- Пустое слово принадлежит языку.
- Если слова X и Y принадлежат языку, то и их конкатенация принадлежит языку.
- Если слово X принадлежит языку, то и слово AXB принадлежит языку (то есть конкатенация префикса, слова X и суффикса).

Если заменить в полученной строке каждую из строк A и B специальным символом, то длина получившейся строки задаёт *сложность* соответствующей фразы в ёрмунгандском языке.

Лингвистам, изучающим ёрмунгандские языки, стало интересно: если взять некоторую строку, составленную из строчных латинских букв, какие значения префикса и суффикса можно выбрать, чтобы сложность соответствующей строке ёрмунгандской фразы была максимально возможной?

Input

В единственной строке входных данных содержится строка s , состоящая из строчных латинских букв. Длина строки не превышает 10^3 .

Output

Выведите через пробел в единственной строке префикс A и суффикс B , при которых данная строка является корректным словом ёрмунгандского языка и имеет при этом максимально возможную сложность. Если вариантов ответа несколько, выберите любой.

Examples

standard input	standard output
abacaba	a bacaba
abab	a b

Problem H. Железный Рикша

Input file: *standard input*
Output file: *standard output*
Time limit: 2 seconds
Memory limit: 512 mebibytes

Город Мегабайтск состоит из N перекрёстков и M улиц с двусторонним движением, соединяющих эти перекрёстки, причём каждые два перекрёстка соединены напрямую не более, чем одной улицей. Для каждой улицы задана её длина; по улицам города можно проехать между любыми двумя перекрёстками, кроме того, верно следующее утверждение: если рассматривать маршруты между двумя перекрёстками, не проходящие дважды через один и тот же перекрёсток, то для любой пары перекрёстков существует **не более двух** таких маршрутов.

Для оптимизации тарифов в приложении агрегатора такси «Железный Рикша» требуется разделить город на две тарифные зоны — «красную» и «синюю». Распределение производится по следующей схеме:

1. Изначально выбираются два различных перекрёстка, один перекрёсток распределяется в красную зону, а второй в синюю.
2. Пусть у нас есть какой-то перекрёсток, уже окрашенный в один из цветов. Если среди перекрёстков, соединённых с данным одной улицей, ещё есть неокрашенный, мы можем окрасить этот перекрёсток в тот же самый цвет.
3. Предыдущий шаг повторяется до тех пор, пока все перекрёстки не будут распределены.

Тарифные зоны требуется выбрать таким образом, чтобы расстояние между зонами было максимальным. Расстоянием между зонами называется длина наименьшей дороги, соединяющей перекрёстки, принадлежащие разным зонам.

Требуется найти значение максимального расстояния и количество способов разделения города на две зоны, позволяющих достичь этого максимума. Разбиения считаются различными, если хотя бы один город в одном разбиении принадлежит зоне одного цвета, а в другом — зоне другого цвета.

Input

Первая строка входного файла содержит два целых числа N и M ($2 \leq N \leq 100$, $N - 1 \leq M \leq N(N - 1)/2$) — количество перекрёстков и количество улицами в Мегабайтске соответственно.

Каждая из последующих M строк содержит по три целых числа a_i , b_i и l_i — номера перекрёстков, соединяемых очередной улицей, и длина улицы, соответственно ($1 \leq a_i, b_i \leq N$, $a_i \neq b_i$, $1 \leq l_i \leq 10^9$). Гарантируется, что никакие два перекрёстка не соединены непосредственно более, чем одной улицей, что между любыми двумя перекрёстками можно проехать по сети улиц и что между любыми двумя перекрёстками существует не более двух различных простых маршрутов.

Output

Выведите два целых числа — наибольшее возможное расстояние между зонами и количество способов разделения города на две зоны, позволяющее получить такое расстояние. Так как ответ может быть слишком большим, выведите его остаток от деления на 998 244 353.

Example

standard input	standard output
6 7 2 3 1 3 4 2 4 2 3 5 6 4 6 1 5 1 5 6 2 5 7	7 18