

## Problem A. Where is the Boundary

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 seconds  
Memory limit: 256 mebibytes

Островное государство JAGония на некоторой планете вытянуто с запада на восток. В JAGонии присутствуют два вида культуры — западная и восточная. Регионы на востоке имеют в основном признаки восточной культуры, регионы на западе — западной; впрочем, провести чёткую границу довольно сложно.

Вам поручено оценить эту границу по имеющимся данным.

Более точно,

1. JAGония разделена на  $n$  префектур, расположенных с запада на восток. Можно считать, что префектуры соответствуют точкам  $1, 2, \dots, n$ , причём точка 1 — самая западная.
2. Собранные данные содержат по  $m$  признаков, каждый из которых определён для каждой префектуры и может принимать значения ‘E’ (восточная) или ‘W’ (западная) в зависимости от того, к какой культуре можно отнести население соответствующей префектуры по данному признаку.
3. Вы должны провести границу, которая минимизирует суммарную ошибку. То есть Вы должны минимизировать количество ‘W’ на территории восточнее границы и количество ‘E’ — на территории, расположенной западнее.
4. Границу можно проводить только по границе между префектурами.

В случае, если все префектуры относятся к восточной культуре, считать, что граница проходит между префектурами 0 и 1, если все относятся к западной — между  $n$  и  $n + 1$ . Если оптимальных границ несколько, выберите наиболее западную (то есть минимальную по значению выводимых чисел) из них.

### Input

Первая строка входа содержит два целых числа  $n$  ( $1 \leq n \leq 10^4$ ) и  $m$  ( $1 \leq m \leq 100$ ) — количество префектур и количество признаков. Каждая из последующих  $m$  строк содержит по  $n$  символов ‘E’ или ‘W’;  $j$ -й символ в  $i$ -й строке обозначает, что население в  $j$ -й префектуре относится по  $i$ -му признаку к восточной или западной культуре соответственно.

### Output

Выведите два целых числа — западную и восточную префектуры, между которыми проходит граница (в случае однородности выведите “виртуальную” нулевую или  $n + 1$ -ю префектуру первой или второй соответственно).

### Examples

standard input	standard output
2 1 WE	1 2
3 2 WWE WEE	1 2
3 1 WWW	3 4
3 1 WEW	1 2

## Note

Во втором примере ответы “1 2” и “2 3” дают по одной ошибке; ответ “1 2” выбирается, так как требуется провести границу как можно западнее.

В третьем примере все префектуры являются западными. Соответственно, граница проходит между третьей и “виртуальной” четвёртой префектурой.

Четвёртый пример показывает, что с точки зрения “здорового смысла” граница может оказаться весьма условной.

## Problem B. Vector Field

Input file: *standard input*  
Output file: *standard output*  
Time limit: 3 seconds  
Memory limit: 256 mebibytes

В 20015-м году группа, занимающаяся разработкой ускорителей элементарных частиц, открыла элемент-ускоритель “Force Point” точечного типа для изучений столкновений протонов на двумерной  $xy$ -плоскости. Если протон проходит через точку, в которой расположен атом элемента-ускорителя, то скорость протона увеличивается вдвое, направление движения изменяется (в зависимости от типа изотопа, расположенного в данной точке, после ускорения протон вне зависимости от того, как он двигался до этого, движется в одном из четырёх направлений, параллельных осям координат). При этом сам ускоритель полностью разрушается (то есть ускорение возможно только один раз).

На плоскости расположено  $n$  атомов элемента-ускорителя. Для каждого атома известно направление движения протона после столкновения с ним. Учёные собираются поместить один протон, движущийся в некотором направлении со скоростью 1, в некоторую точку плоскости так, чтобы на выходе скорость протона была максимальна (начальную точку и направление учёные могут выбирать).

Требуется вычислить, какой скорости протона могут достичь учёные при оптимальном выборе

### Input

Первая строка входа содержит одно целое число  $n$  ( $1 \leq n \leq 3000$ ) — количество атомов-ускорителей на плоскости. Каждая из последующих  $n$  строк содержит по два целых числа  $x_i$  и  $y_i$  ( $|x_i|, |y_i| \leq 10^9$ ) и один символ  $d_i$  ( $d_i$  — один из символов ‘>’, ‘v’, ‘<’ или ‘^’).  $x_i$  и  $y_i$  задают координату  $i$ -го атома-ускорителя, а  $d_i$  — направление, в котором протон будет двигаться после прохождения  $i$ -го атома-ускорителя: ‘>’ обозначает, что протон будет двигаться параллельно оси  $x$  с вектором  $(1, 0)$ , ‘v’ — параллельно оси  $y$  с вектором  $(0, 1)$ , ‘<’ — параллельно оси  $x$  с вектором  $(-1, 0)$  и ‘^’ — параллельно оси  $y$  с вектором  $(0, -1)$ . Координаты всех атомов-ускорителей попарно различны.

### Output

Выведите одну строку, содержащую целое число  $\log_2 v_{max}$ , где  $v_{max}$  — максимальная скорость протона при оптимальном выборе.

## Example

standard input	standard output
9 0 0 v 1 0 > 2 0 < 0 1 > 1 1 v 2 1 v 0 2 ^ 1 2 ^ 2 2 <	9
9 0 0 ^ 1 0 ^ 2 0 ^ 0 1 < 1 1 ^ 2 1 > 0 2 v 1 2 v 2 2 v	2

## Note

В первом примере ввод можно схематически обозначить следующей диаграммой:

```
v><  
>vv  
^^<
```

Если поместить протон в (1,1), то он пройдёт через все атомы-ускорители.

## Problem D. Identity Function

Input file: *standard input*  
Output file: *standard output*  
Time limit: 4 seconds  
Memory limit: 256 mebibytes

Вам задано целое число  $N$ , большее единицы.

Рассмотрим следующие функции:

- $f(a) = a^N \pmod N$
- $F_1(a) = f(a)$
- $F_{k+1}(a) = F_k(f(a)) (k = 1, 2, 3, \dots)$

Здесь  $\pmod$  — это операция взятия по целочисленному модулю. Для неотрицательного целого числа  $x$  и целого положительного  $y$   $x \pmod y$  — остаток от деления  $x$  на  $y$ .

Выведите наименьшее целое положительное  $k$  такое, что  $F_k(a) = a$  для всех целых положительных  $a$ , меньших  $N$ . Если таких  $k$  не существует, выведите  $-1$ .

### Input

Единственная строка входа содержит целое число  $N$  ( $2 \leq N \leq 10^9$ ).

### Output

Выведите наименьшее целое положительное  $k$  такое, что  $F_k(a) = a$  для всех целых положительных  $a$ , меньших  $N$ , или выведите  $-1$ , если таких  $k$  не существует.

### Examples

standard input	standard output
3	1
4	-1
15	2

## Problem F. Marching Course

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 seconds  
Memory limit: 256 mebibytes

Духовой оркестр Kitafuji Hight School будет участвовать в фестивале марширующих оркестров. Время выступления оркестра, согласно регламенту фестиваля, не должно превышать  $P$  минут. При этом оркестр может выбрать свой маршрут по территории, на которой проводится фестиваль. Территория представляет собой  $N$  площадок, пронумерованных последовательными целыми числами от 1 до  $N$ , и  $M$  дорог с двусторонним движением, соединяющих некоторые площадки. Участникам известна длина каждой дороги и количество зрителей, которые будут стоять вдоль неё во время фестиваля. Каждый оркестр должен стартовать с первой площадки и вернуться на первую площадку через  $P$  минут, двигаясь со скоростью 1 единицу длины в минуту. При этом оркестр может развернуться в любое время и в любом месте, включая точку посередине дороги.

Руководитель оркестра хочет выбрать маршрут таким образом, чтобы максимизировать *впечатление на зрителей*. Оно определяется следующим образом: если оркестр играет в течение  $t$  минут на дороге длиной  $d$  единиц длины, на которой ожидается  $v$  зрителей, то впечатление равно  $t \times v/d$ . Общее впечатление является суммой таких параметров по всем дорогам. При этом впечатление не зависит от того, проходил уже данный оркестр по данной дороге или нет.

По заданной схеме площадок фестиваля вычислите максимальное суммарное впечатление, которое может произвести оркестр.

### Input

Первая строка ввода содержит три целых числа  $N$ ,  $M$ , и  $P$ : количество площадок  $N$  ( $2 \leq N \leq 200$ ), количество дорог  $M$  ( $N - 1 \leq M \leq N(N - 1)/2$ ) и время выступления  $P$  ( $1 \leq P \leq 1000$ ). Последующие  $M$  задают информацию относительно дорог.  $i$ -я из этих строк содержит четыре целых числа  $s_i$ ,  $t_i$ ,  $d_i$  и  $v_i$ :  $i$ -я дорога соединяет площадки  $s_i$  и  $t_i$  ( $1 \leq s_i, t_i \leq N$ ,  $s_i \neq t_i$ ) имеет длину  $d_i$  ( $1 \leq d_i \leq 1000$ ) и вдоль неё предполагается  $v_i$  ( $1 \leq v_i \leq 1000$ ) зрителей.

Гарантируется, что между любыми двумя площадками можно пройти по одной или нескольким дорогам и что любые две площадки непосредственно соединены не более, чем одной дорогой.

### Output

Выведите максимальное суммарное впечатление от  $P$ -минутного выступления с абсолютной погрешностью не хуже  $10^{-4}$ .

## Examples

standard input	standard output
3 3 4 1 2 1 1 2 3 2 4 3 1 1 1	6
4 3 9 1 2 2 1 1 3 2 2 1 4 2 3	13.5
4 3 5 1 2 10 1 2 3 2 100 1 4 3 10	16.6666666667
3 3 10 1 2 3 1 1 3 4 5 2 3 2 10	22

## Problem G. Surface Area of Cubes

Input file: *standard input*  
Output file: *standard output*  
Time limit: 3 seconds  
Memory limit: 256 mebibytes

Таро играет в игру “Surface Area of Cubes”.

В этой игре дан параллелепипед  $A \times B \times C$ , составленный из  $A \times B \times C$  единичных кубиков. Центр каждого единичного кубика находится в целочисленной точке  $(x, y, z)$  ( $0 \leq x \leq A - 1$ ,  $0 \leq y \leq B - 1$ ,  $0 \leq z \leq C - 1$ ). После чего мастер убирает  $N$  различных единичных кубиков. Игрок после этого должен сообщить общую площадь поверхности получившегося объекта.

Операция удаления не меняет положения кубиков, которые не были удалены; удалены могут быть не только кубики, примыкающие к поверхности исходного параллелепипеда; получившийся объект может в результате представлять собой несколько несвязных частей; в площадь поверхности входит и площадь “внутренних поверхностей”, недоступных извне, если таковые имеются.

Напишите программу, которая помогает игроку подсчитать требуемую площадь.

### Input

Первая строка входа содержит четыре целых числа  $A$ ,  $B$ ,  $C$  и  $N$  ( $1 \leq A, B, C \leq 10^8$ ,  $0 \leq N \leq \min\{1,000, A \cdot B \cdot C - 1\}$ ).

Каждая из последующих  $N$  строк содержит целые неотрицательные числа  $x$ ,  $y$  и  $z$ , которые задают координаты очередного удаляемого кубика. Гарантируется, что кубик с такими координатами существует и ещё не был удалён.

### Output

Выведите площадь поверхности получившегося после удаления  $N$  кубиков объекта.

### Examples

standard input	standard output
2 2 2 1 0 0 0	24
1 1 5 2 0 0 1 0 0 3	18
3 3 3 1 1 1 1	60



## Problem H. Laser Cutter

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2.5 seconds  
Memory limit: 256 mebibytes

Ciel хочет сделать разрез в деревянной доске с помощью лазерного ножа.

На доске уже намечены отрезки, вдоль которых надо делать разрез. Линии являются *направленными*, то есть структура материала позволяет корректно разрезать доску вдоль каждого отрезка только в заданном направлении. При этом *без учёта направления* разрез является связным, то есть любые две точки разреза соединены одним или несколькими отрезками.

Когда лазерный нож включён, он делает разрез вдоль линии, по которой движется. Лазер первоначально находится над точкой  $(x, y)$ . Возможны следующие две операции:

- Движение включённого лазерного ножа; тогда он делает разрез вдоль траектории движения.
- Движение выключенного лазерного ножа.

При этом делать разрез вдоль отрезка целиком не обязательно: можно включать и выключать лазер над любой точкой разреза.

Требуется найти минимальное расстояние, которое пройдёт лазерный нож для того, чтобы сделать необходимый разрез и вернуться в исходную точку.

### Input

Первая строка входа содержит целое число  $n$  ( $1 \leq n \leq 300$ ) — количество отрезков. Следующая строка содержит два целых числа  $x$  и  $y$  ( $-1000 \leq x, y \leq 1000$ ), задающие начальную (и конечную) точку пути лазера.  $i$ -я из последующих  $n$  строк содержит четыре целых числа  $sx_i$ ,  $sy_i$ ,  $tx_i$  и  $ty_i$  ( $-1000 \leq sx_i, sy_i, tx_i, ty_i \leq 1000$ ) — начальная и конечная точка  $i$ -го отрезка соответственно, причём направление разреза должно всегда идти от  $(sx_i, sy_i)$  к  $(tx_i, ty_i)$ .

Для всех  $i$  ( $1 \leq i \leq n$ ),  $(sx_i, sy_i) \neq (tx_i, ty_i)$ . Начальная точка  $(x, y)$  лежит как минимум на одном отрезке. Для всех различных пар  $i, j$  ( $1 \leq i, j \leq n$ )  $i$ -й и  $j$ -й отрезки имеют не более одной общей точки.

### Output

Выведите одно число — минимальное расстояние, которое пройдёт лазерный нож, с абсолютной или относительной погрешностью не хуже  $10^{-6}$ .

## Examples

standard input	standard output
3 0 1 0 0 0 1 0 1 0 2 0 2 0 3	6.0
2 0 1 0 0 0 2 -1 1 1 1	6.8284271247461900
5 0 0 0 0 1 0 1 1 -1 1 -1 1 -1 -1 -1 -1 1 -1 1 -1 1 1	10.00

## Problem J. Black Company

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 seconds  
Memory limit: 256 mebibytes

Подпольный синдикат собирается установить зарплаты  $N$  своим сотрудникам, пронумерованным числами от 1 до  $N$ . Зарплата каждого сотрудника должна быть целым положительным числом. Если вклад  $c_i$  сотрудника  $i$  выше, чем вклад  $c_j$  сотрудника  $j$ , то сотрудник  $i$  должен получать более высокую зарплату. Однако в случае, если сотрудники не знакомы между собой и не имеют общих знакомых, это правило может не выполняться.

Иначе говоря,

- Если сотрудники  $i$  и  $j$  знакомы друг с другом, то должно выполняться  $c_i < c_j \Leftrightarrow p_i < p_j$ , где  $p_i$  — зарплата  $i$ -го сотрудника.
- Если сотрудник  $i$  знаком с сотрудниками  $j$  и  $k$ , должно выполняться  $c_j < c_k \Leftrightarrow p_j < p_k$ .

Найдите минимальную суммарную зарплату всех сотрудников с учётом вышеупомянутых ограничений.

### Input

Первая строка ввода содержит целое число  $N$  ( $1 \leq N \leq 10^5$ ) — количество сотрудников. Вторая строка содержит  $N$  целых чисел  $c_i$  ( $1 \leq c_i \leq 10^5$ ) — вклад  $i$ -го сотрудника.

Третья строка содержит целое число  $M$  ( $0 \leq M \leq 2 \cdot 10^5$ ) — количество пар сотрудников, знакомых между собой. Каждая из последующих  $M$  строк содержит два целых числа  $a_i$  и  $b_i$  ( $a_i \neq b_i$ ,  $1 \leq a_i, b_i \leq N$ ) — номера сотрудников, знакомых между собой. Гарантируется, что никакая пара сотрудников не встречается в этих строках дважды.

### Output

Выведите одно целое число — минимальную суммарную зарплату всех сотрудников.

## Examples

standard input	standard output
3 1 3 3 2 1 2 1 3	5
3 1 2 3 2 1 2 1 3	6
4 1 1 2 2 2 1 2 3 4	4
5 1 2 5 5 1 6 1 2 4 1 2 3 5 2 4 3 4 5	10
6 4 3 2 1 5 3 7 4 2 1 5 2 6 6 5 4 1 1 6 6 3	13

## Problem K. Emoticon Counter

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 seconds  
Memory limit: 256 mebibytes

Три последовательных символа “:-)” в текстовом сообщении обозначают радостный смайлик, а три последовательных символа “:-(” — грустный.

Общее настроение текстового сообщения определим так: если в сообщении весёлых смайликов больше, чем грустных, то ответ будет “**happy**”, если грустных больше, чем весёлых, то ответ будет “**sad**”, иначе ответ — “**neutral**”.

Напишите программу, которая определяет общее настроение сообщения.

### Input

Одна строка, содержащая от 1 до 255 символов с кодами, не меньшими 32 и не превосходящими 127.

### Output

Выведите строку, соответствующую общему настроению сообщения.

### Examples

standard input	standard output
Solved the problem :-) got TL :-( then Accepted :-)	happy
Happy? :)	neutral
Emo:-(ticons are go:-(:(-ing out of co:-)ntrol	sad

## Problem L. Rogue Language

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

Задан шифр, строящийся следующим образом:

- Гласные ('a', 'e', 'i', 'o' and 'u') остаются на месте.
- Согласные заменяются тремя буквами: исходной согласной, ближайшей к ней гласной (для 'b' ответом будет 'a'), если расстояния от двух гласных равны, то гласной с наименьшим номером (для 'c' ответом также будет 'a'), и следующей за исходной согласной; для 'z' снова берётся 'z'.

Зашифруйте заданное слово.

### Input

Входной файл состоит из одного непустого слова, составленного из не более, чем 30 строчных латинских букв.

### Output

Выведите зашифрованное слово.

### Examples

standard input	standard output
opencup	opoqenopcadupoq
xyz	xuyyuzzuz

## Problem M. RPS

Input file: *standard input*  
Output file: *standard output*  
Time limit: 2 seconds  
Memory limit: 256 mebibytes

Алиса и Боб играют в игру “камень-ножницы-бумага”. В каждой партии они одновременно выбирают одну из трёх фигур: камень (rock), ножницы (scissors) или бумага (paper). При этом камень выигрывает у ножниц, ножницы — у бумаги, а бумага — у камня. Если выбраны одинаковые фигуры, фиксируется ничья.

Вычислите, сколько партий выиграла Алиса, и сколько — Боб.

### Input

Первая строка входа содержит целое число  $N$  ( $1 \leq N \leq 100$ ) — количество сыгранных партий.

Вторая строка задаёт последовательность ходов Алисы и содержит  $N$  слов.  $i$ -е слово в последовательности задаёт ход Алисы в  $i$ -й партии и может быть одним из вариантов “rock” (для камня), “paper” (для бумаги) и “scissors” (для ножниц).

Третья строка задаёт последовательность ходов Боба в том же самом формате.

### Output

Выведите два целых числа — количество партий, выигранных Алисой, и количество партий, выигранных Бобом, соответственно.

### Examples

standard input	standard output
4 paper scissors rock paper rock rock scissors paper	2 1