KTH Challenge

KTH Challenge 2019

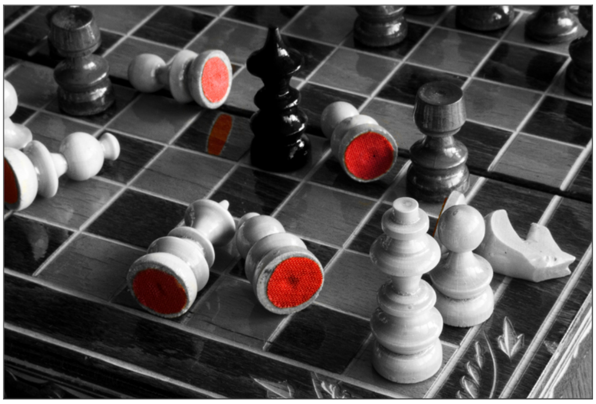
April 27, 2019

Problems

Do not open before the contest has started.

This page is intentionally left (almost) blank.

Problem A
Assassins
Problem ID: assassins
Time limit: 1 second



Picture by Cristian V. from flickr, cc by-nd

In the cut-throat world of assassins for hire, the rivalry is ruthless and everyone is fighting to get an edge. To eliminate the competition, many assassins even go so far as to assassinate other assassins. The question is: with several assassins trying to do each other in, which ones will remain alive and kicking, and which ones will kick the bucket?

Assassins generally lay careful plans before executing them, including planning multiple attempts to dispose of the same target, with the second attempt being a backup in case the first attempt fails, the third attempt being a secondary backup, and so on. Using their great annihilytical skills, assassins can also very accurately determine the probability that any given assassination attempt will succeed.

Given the list of planned assassination attempts for a group of assassins, what are the probabilities that each assassin is alive after all these attempts? Performing an assassination attempt requires that the assassin is still alive, so if the assassin is indisposed due to already having been assassinated, the attempt is cancelled.

Input

The first line of input contains two integers n and m, where n ($1 \leq n \leq 15$) is the number of assassins, and m ($0 \leq m \leq 1000$) is the number of planned assassination attempts. The assassins are numbered from 1 to n.

Then follow m lines, each containing two integers i, j, and a real number p, indicating that assassin i plans to attempt to assassinate assassin j ($1 \leq i, j \leq n$, $j \neq i$), and that this attempt would succeed with probability p ($0 \leq p \leq 1$, at most 6 digits after the decimal point). The planned attempts are listed in chronological order from first to last, and no two attempts happen simultaneously.

Output

Output n lines, with the i'th containing the probability that assassin i is alive after all m assassination attempts have taken place. You may assume that none of the n assassins will die of any other cause than being assassinated in one of these m attempts. The probababilities should be accurate to an absolute error of at most 10^{-6}.

Sample Input 1	Sample Output 1
4 3 1 2 0.25 1 4 0.42 2 3 1.0	1 0.75 0.25 0.58

Sample Input 2	Sample Output 2
``2 3 1 2 0.23 2 1 0.99 1 2 0.99```	```0.2377000000 0.7623770000```

Problem B
Espresso Bucks
Problem ID: espressobucks
Time limit: 1 second

Image by talinamalu from Pixabay

The big café chain Espresso Bucks is expanding to the country of Gridland. Since you are an expert on Gridland culture and geography, Espresso Bucks have hired you to decide where to put their coffee shops for maximum profit. Gridland consists of an n by m grid, where some cells are land, and some are water. At most one coffee shop can be built on each land cell. Nothing can be built on the water cells, but on the other hand, no one lives in the water. After a lot of long meetings with the Espresso Bucks people, you have come to the conclusion that there are only two constraints the placement of coffee shops has to satisfy:

1. Each land cell must have a coffee shop directly on it, or adjacent to it.
2. No two coffee shops can be adjacent to each other.

Two cells are adjacent if one is immediately to the west, north, east, or south of the other. Find any placement of coffee shops that satisfies these constraints.

Input

The first line of input consists of two integers n and m ($1 \leq n, m \leq 100$). The following n lines each contain a string of length m consisting only of the characters '`.`' (land) and '`#`' (water). This is the map of gridland. It is guaranteed that the map contains at least one land cell.

Output

Output a copy of the map of gridland, where some of the land cells have been replaced with the letter '`E`', meaning that a coffee shop was placed on the corresponding land cell. This placement should satisfy the constraints above. If there are many solutions, any one will be accepted.

Sample Input 1	Sample Output 1
5 6 ....#. ....#. #..... ...... ##...#	E..E#. ..E.#E #..E.. .E...E ##E.E#

Sample Input 2

```
10 16
########.#....##
######......####
#####.....######
###.......######
##.......#######
##.........#####
##......########
##....##########
###.############
##...###########
```

Sample Output 2

```
########E#.E.E##
######E...E.####
#####E.E.E######
###E..E.E.######
##E..E.E.#######
##.E..E..E.#####
##E..E.E########
##.E..##########
###.############
##E.E###########
```

Problem C
Loo Rolls
Problem ID: loorolls
Time limit: 1 second

Image by Karen Arnold from Pixabay.

Your friend Nick needs your help with a hard problem that he came across in real life. Nick has a loo roll of length ℓ centimetres in his bathroom. Every time he visits the toilet, he uses exactly n centimetres of loo roll. When the roll runs out, Nick always goes to the store and buys a new one of length ℓ directly afterwards. However, sometimes the roll runs out even though Nick still needs a non-zero amount of paper. Let us call such an event a *crisis*.

Nick has a clever way of preventing crises from happening: he uses a backup roll. The backup roll is another roll of length ℓ that is hidden somewhere in the bathroom, and when the regular roll runs out even though Nick still needs more paper, he will take that amount from the backup roll. Then he will replace the regular roll directly after the visit.

As you can imagine, this makes crises much less frequent. But still, the backup roll will also slowly run out, and eventually a crisis might still happen. So to generalize this, Nick wants to use several layers of backup rolls. First he will take paper from roll number 1 (the regular roll), if it runs out he will take from roll number 2, then if roll 2 runs out from roll number 3, and so on all the way up to roll number k. After each visit, all the rolls that have run out will be replaced. Nick managed to prove that if he picks a large enough number k, he can actually make it so that crises never happen! Your task is to find the smallest such number k.

Input

The input consists of a single line containing the two integers ℓ and n ($1 \leq n \leq \ell \leq 10^{10}$).

Output

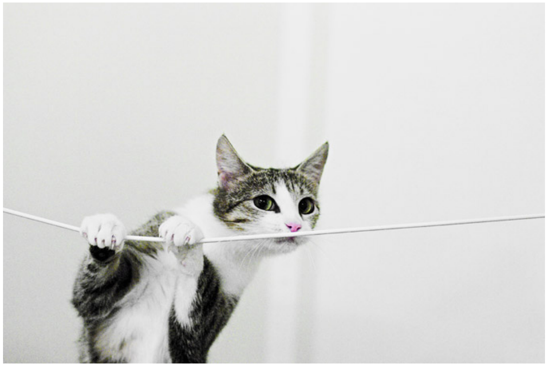
Output the smallest integer k such that crises will never happen when using k layers of rolls (including the regular roll).

Sample Input 1	Sample Output 1
`31 6`	`4`

Sample Input 2	Sample Output 2
`10000000000 17`	`3`

This page is intentionally left (almost) blank.

Problem D
Meow Factor 2
Problem ID: meowfactor2
Time limit: 2 seconds



Picture by Stefan Tell on Flickr, cc by

Strings of yarn have been popular in Catland for ages. Which cat has not spent many a lazy afternoon bouncing around a ball of yarn? Lately however, strings of yarn have gotten competition: strings of characters. It turns out that these are almost as much fun as yarn, and generally much safer as well (so far, no cat has had to call 911 on account of any character string-related entanglement accidents).

Naturally, some strings are more stylish than others, and for cool cats it is important to engage in their string-playing pastime with style. The *meow factor* of a string S is the minimum number of operations needed to transform S into a string S' which contains the word "meow" as a substring, where an operation is one of the following four:

1. Insert an arbitrary character anywhere into the string.
2. Delete an arbitrary character anywhere from the string.
3. Replace any character in the string by an arbitrary character.
4. Swap any two adjacent characters in the string.

Write a program to compute the meow factor of a string of characters.

Input

The input consists of a single line containing a string S, consisting only of lower-case letters '`a`'-'`z`'. The length of S is at least 1 and at most 10^6.

Output

Output the meow factor of S.

Sample Input 1	Sample Output 1
`pastimeofwhimsy`	`1`

Sample Input 2	Sample Output 2
`yarn`	`4`

This page is intentionally left (almost) blank.

Problem E
Pea Soup and Pancakes
Problem ID: peasoup
Time limit: 1 second

As a Swede, you hold a deep love for the traditional Thursday lunch of pea soup and pancakes. You love it so much, in fact, that you will eat it any meal it is available. You find yourself looking at the menus for all your favorite restaurants every day to see if this combination is available, and realized you can do this more easily with a program. Given a list of restaurant menus, decide where to eat.

Input

The first line of input contains a number n ($1 \le n \le 10$), the number of restaurants. Then follow the n restaurant menus. Each menu starts with a line containing a number k ($1 \le k \le 10$), the number of menu items for the day. The remainder of the menu consists of $k+1$ lines, each containing a nonempty string of at most 100 characters. The first of these lines is the restaurant name, and the rest are menu items. Strings consist only of lower case letters '`a`'-'`z`' and spaces, and they always start and end with a letter. All restaurant names are unique.

Output

Output a single line. If at least one restaurant has both "`pea soup`" and "`pancakes`" as menu items, output the name of the first of those restaurants, by the order in which the restaurants appear in the input. Otherwise, output "`Anywhere is fine I guess`".

Sample Input 1	Sample Output 1
2 2 q potatoes salad 3 nymble pancakes pea soup punsch	nymble

Sample Input 2

```
4
2
asian wok house
paa soup
pancakes
2
kebab kitchen
pea  soup
pancakes
2
la campus
tasty pea soup
pancakes
3
slime stand
slime
pea soup and pancakes
slime
```

Sample Output 2

```
Anywhere is fine I guess
```

Problem F
Ski Lifts
Problem ID: skilifts
Time limit: 1 second

Picture by SkiHoodoo from Wikimedia Commons, cc by-sa

Last winter, an avalanche swept away all the ski lifts from the ski resort Valen. Instead of rebuilding the lifts like they were before, the plan is to do it in a more optimized way, and you are responsible for this.

The only thing remaining from the old lift system are n pylons situated at integer coordinates in the plane. You would like to put lifts in the form of line segments between some of these pylons. The line segments must satisfy the following constraints:

1. A line segment can only go between pylons (x_1, y_1) and (x_2, y_2) if $|y_1 - y_2| = 1$.
2. There are two types of pylons, one-way and two-way pylons. The one-way pylons can be connected to at most one other pylon, and the two-way pylons can be connected to at most two other pylons. However, if a two-way pylon i is connected to two other pylons, then they must be on opposite sides of i in the y-direction. In other words, the two pylons connected to i must have different y-coordinates.
3. Two line segments may not intersect (except that the two line segments incident on a two-way pylon may touch at their endpoints).

What is the maximum number of ski lifts (line segments) you can place under these constraints?

Input

The first line contains one integer n ($1 \leq n \leq 10^5$). Each of the following n lines contains three integers x, y, and a, the coordinates and type of a pylon ($0 \leq x, y \leq 10^5$; $a = 1$ for a one-way pylon and $a = 2$ for a two-way pylon). All the pylons are situated at different coordinates.

Output

Output the maximum number of ski lift line segments that can be placed.

Sample Input 1	Sample Output 1
8 1 0 1 3 0 2 0 1 1 2 1 2 4 1 2 1 2 2 2 3 1 4 3 1	4

Sample Input 2	Sample Output 2
``4 0 0 1 100000 1 1 0 99999 1 100000 100000 1```	```2```

Problem G
Symmetric Polynomials
Problem ID: symmetricpolynomials
Time limit: 1 second

A symmetric polynomial is a polynomial in n variables that remains the same polynomial under any permutation of the variables. For example, $f(x_1, \ldots, x_n) = x_1 + \ldots + x_n$ is a symmetric polynomial (it is in fact called the first elementary symmetric polynomial). Symmetric polynomials have many important applications. They can, for instance, be used to prove that there is no formula for the roots of a general five-degree polynomial.

In this problem however, we will concern ourselves with another kind of symmetry. Consider an infinite curve P in the plane where both x and y coordinates are given by polynomials, i.e.,

$$\begin{aligned} x(t) &= a_n t^n + a_{n-1} t^{n-1} + \ldots + a_1 t + a_0, \\ y(t) &= b_m t^m + b_{m-1} t^{m-1} + \ldots + b_1 t + b_0. \end{aligned}$$

We say that such a curve is symmetric around a straight line L (given as an equation $Ax + By + C = 0$) if there exists a real number t_0 such that for all $t \in \mathbb{R}$ the point $(x(t_0 + t), y(t_0 + t))$ is the reflection of $(x(t_0 - t), y(t_0 - t))$ around the line L, and we call the line L a symmetry line for the curve P. For example, consider the curve P given by

$$\begin{aligned} x(t) &= -5t^5 - 26t^4 - 19t^3 + 59t^2 + 111t + 26, \\ y(t) &= -t^5 + 17t^3 - 9t^2 - 61t + 12. \end{aligned}$$

This curve is symmetric around the line $5x + y + 92 = 0$ with $t_0 = -1$ (see Figure G.1).

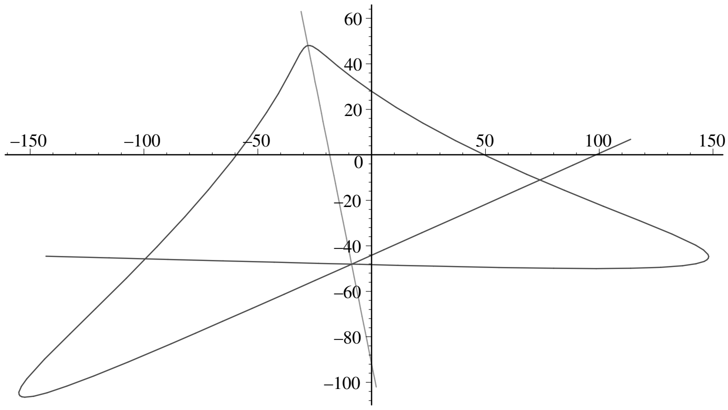


Figure G.1: Illustration of Sample Input 1, drawn from $t = -3.9$ to $t = 1.9$.

Now, your task is to write a program that, given the two polynomials $x(t)$ and $y(t)$, finds a symmetry line of the curve (if one exists).

Input

The first line of input contains an integer n ($0 \leq n \leq 10$), the degree of x. Then follows a line with $n + 1$ integers $a_n, \ldots, a_1, a_0$, where a_i is the degree i coefficient of x. Then follow two lines describing the polynomial y in the same format.

If either of $x(t)$ or $y(t)$ is the zero polynomial, its degree is given as 0. The coefficients have absolute values bounded by $1\,000$. You may assume that the leading coefficient a_n of each polynomial is non-zero, except in the case when the polynomial is the zero polynomial.

Output

Output three real numbers A, B and C, indicating that $Ax + By + C = 0$ is a symmetry line for the given curve. If there is no symmetry line, let $A = B = C = 0$. If the curve has more than one symmetry line, any one will be accepted.

In the case when a symmetry line exists, the provided line must satisfy the following conditions:

- $0.5 \leq \max(|A|, |B|) \leq 10^{100}$.
- The direction of the provided line is within 10^{-6} radians of some symmetry line.
- The value of $\frac{C}{\max(|A|,|B|)}$ is correct within an absolute or relative error of 10^{-6}.

Sample Input 1	Sample Output 1
5 -5 -26 -19 59 111 26 5 -1 0 17 -9 -61 12	5 1 92

Sample Input 2	Sample Output 2
1 1 0 3 1 0 0 0	0 0 0

Sample Input 3	Sample Output 3
1 1 0 0 0	2.718281828 0 0

Problem H
Type Charts
Problem ID: typecharts
Time limit: 2 seconds

You are playing the game Nudgémon, a common element of which is Nudgémon battles. In a battle, you and your opponent each start by sending out a Nudgémon of your choice, and then take turns attacking the opposing Nudgémon.

Each attack has a type (an integer between 1 and n), and the opposing Nudgémon also has either one or two types. Depending on these types, the attack will do different amounts of damage.

When an attack of type i hits a Nudgémon with single type j, the attack gets a damage multiplier $a(i, j)$, where a is a type matchup table consisting of entries in $\{0, 0.5, 1, 2\}$. If it hits a Nudgémon with double types j and k, it gets a damage multiplier of $a(i, j) \cdot a(i, k)$.

Depending on the value v of the damage multiplier, the game will emit different messages:

$v = 0$	It had no effect.	x
$0 < v < 1$	It's not very effective…	−
$v = 1$	*<no message>*	=
$v > 1$	It's super effective!	+

You are new to this game and do not know what the table a looks like. Trying to learn its first row, you have gathered some observations about the game's output when attacking various Nudgémon with attacks of type 1. Now you are trying to reconstruct the first row in a way that is consistent with this data.

Input

The first line of input contains two integers n and m ($1 \leq n \leq 10^5$, $1 \leq m \leq 10^5$), where n is the number of types and m is the number of observations.

Then follow m lines, each containing two integers i, j and a character c ($1 \leq i, j \leq n$ and c is one of x,−,= or +), where c is the observed effect when attacking a Nudgémon with types i and j, as indicated in the table above. If $i = j$, the Nudgémon has just a single type.

Output

Output a single line with n characters, each either x, −, = or +. The ith character should describe the effect of attacking a Nudgémon of type i with an attack of type 1.

If there are multiple valid solutions, you may output any one of them. It is guaranteed that at least one solution exists.

Sample Input 1

```
5 5
1 2 -
2 4 -
4 5 x
2 3 =
3 4 +
```

Sample Output 1

```
=-+=x
```

This page is intentionally left (almost) blank.