

Problem A. Fit 1: The Landing

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 64 megabytes

Перед выходом из британских территориальных вод, корабль, на котором отправились в путешествие охотники за Снарком, был остановлен для досмотра. В ходе досмотра выяснилось, что Булочник не знает своего имени, а все документы он оставил в одном из сорока двух чемоданов, которые он забыл на берегу. Поэтому, с целью выяснения личности Булочника, в порт была отправлена радиограмма. К сожалению, в тот момент, когда пришла ответная радиограмма, на месте радиста находился Бобёр, который использовал ленту для своих целей. Так что лента была приведена в частичную негодность: хотя места наличия знаков на ленте были ещё различимы, только в некоторых местах можно было разобрать, точка там или тире. Кроме того, Бобёр правильно запомнил длины последовательностей подряд идущих тире, от точки до точки, а про точки он как-то не подумал, считая их фоном узора. Восстановите радиограмму целиком или, если это невозможно, восстановите те знаки, которые восстанавливаются однозначно. Если не существует ни одной радиограммы, удовлетворяющей условиям, выведите "BEAVERERROR"

Input

Испорченная лента на первой строке в формате $A_1 \dots A_N$, где A_i — или "." (точка), или "-" (тире, знак минуса), или "0" (нечитаемый символ, цифра ноль). Длина ленты не превосходит 10000 знаков. Далее, второй строкой, следует список длин последовательностей подряд идущих тире, которые запомнил Бобёр, в формате L_1, \dots, L_K в порядке слева направо. Например, для радиограммы "----.--" список будет "3 2 1".

Output

Восстановленная радиограмма в формате $A_1 \dots A_N$, где A_i есть или . (точка), или - (тире), или 0 (знак не восстанавливается однозначно), или строка "BEAVERERROR".

Example

standard input	standard output
.-0.- 2 1
000 2	0-0
.. 1	BEAVERERROR

Problem B. Fit 2: The Bellman’s Speech

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 64 megabytes

На карте, которую Благовзон предложил экспедиции, не было нанесено земли, мелей, тропиков, полюсов и экватора. На ней был только океан и своеобразная “случайная координатная сетка”. Случайная координатная сетка представляет собой N отрезков. Концы отрезков могут совпадать. Никакие два отрезка не пересекаются по отрезку. Благовзон сказал, что внутри каждой из частей, на которую отрезки сетки разбивают карту, находится по острову. Выясните, сколько островов представлено на заданной карте.

Input

В первой строке записано число отрезков N — натуральное число, не превосходящее 1000. Затем для каждого отрезка с новой строки через пробел записаны декартовы координаты концов отрезков — целые числа x_1, y_1, x_2, y_2 , не превосходящие по модулю 100000. Сама карта — прямоугольник с вершинами $(-142857, -142857)$, $(-142857, 142857)$, $(142857, 142857)$, $(142857, -142857)$.

Output

Количество частей, на которые эти отрезки разбили карту (включая “внешнюю” часть).

Example

standard input	standard output
1 1 1 2 2	1
4 0 0 0 1 0 1 1 1 1 1 1 0 1 0 0 0	2

Problem C. Fit 3: The Baker’s tale

Input file: standard input
Output file: standard output
Time limit: 10 seconds
Memory limit: 64 megabytes

Булочник сообщил экспедиции важную информацию о методах охоты на Снарка. Однако, для него не это было главным. Он знал, что если вместо Снарка ему встретится Буджум, шансы оставить от себя хотя бы мокрое место у Булочника ничтожно малы. Узнав про это, его товарищи по экспедиции решили выяснить у Благозвона, чем отличаются породистые Снарки от не менее породистых Буджумов. После ответов из серии “Увидите... Узнаете...”, Барристер, мобилизовав всё своё красноречие, смог вытащить следующую информацию. Следы, оставляемые обоими существами, представляют собой набор точек на земле, некоторые из которых соединены между собой отрезками. Любые две точки соединены не более, чем одним отрезком.

Следы Снарков могут быть самыми разными, но все они получены из “базового следа”
(AB, BC, AC, AD, BD, CD)

при помощи некоторого числа преобразований всего двух видов, а именно:

1. (AX, BX, CX) \rightarrow (AQ, BP, CR, PQ, QR, PR)
2. (AB, CD) \rightarrow (AC, BD)

Здесь буквами обозначены точки, парами букв — отрезки, соединяющие точки. При преобразовании, точки из левой части, отсутствующие в правой части преобразования, исчезают, равно как отрезки из левой части преобразования, отсутствующие в правой. Объекты, отсутствующие в левой части, но присутствующие в правой, добавляются.

Если же след не может быть получен из “базового” подобным образом, то это след Буджума. Осталось только научиться различать эти случаи, и опасность исчезнуть для Булочника резко уменьшится.

Input

$1 \leq K \leq 1000$ — число точек, $N > 0$ — число отрезков, затем сами отрезки, каждый отрезок с новой строки. Отрезки задаются номерами точек A_i и B_i для концов каждого отрезка. Точки нумеруются натуральными числами от 1 до K .

Output

Слово **Snark**, если это след Снарка, либо **Boojum**, если это след Буджума.

Example

standard input	standard output
2 1 1 2	Boojum
4 6 1 2 2 3 3 1 1 4 2 4 3 4	Snark

Problem D. Fit 4: The Hunting

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 64 megabytes

Барристера заинтересовал один из способов поимки Снарка, который он вычитал в найденном руководстве “Snark Hunting for Dummies in 21 days”. Снарка можно поймать так: “Охотники должны ночью высадиться на остров с разных сторон, и сближаться до определённого момента. Затем некоторые из участников охоты бросают друг другу верёвки так, что получается замкнутый многоугольник, внутри которого и спит Снарк. После чего поимка становится делом техники, если, конечно, все смогли вести себя достаточно тихо”.

Благозвон хочет, используя этот способ, обойтись минимальным количеством верёвки — ведь её запасы на корабле ограничены, а после поимки, Снарка, возможно, придётся связывать. К тому же, существует ещё одна проблема — максимальная длина, на которую можно бросить верёвку, ограничена и равна K . Вычислите минимальную длину верёвки, если заданы финальные координаты всех ловцов и координаты, можно надеяться, спящего Снарка.

Input

В первой строке записано число участников экспедиции N — натуральное число от 3 до 100 и K — максимальная длина куска верёвки — вещественное число, большее 0 и не превосходящее 30000. В следующей строке дана пара координат Снарка. Далее записаны N пар координат X_i, Y_i финального положения каждого из участников экспедиции, по паре координат в одной строчке. Координаты заданы с 4 знаками после запятой и по модулю не превосходят 10000.

Output

Минимальная длина требуемой верёвки с точностью два знака после запятой.

Example

standard input	standard output
3 6.00 1.0000 1.0000 0.0000 0.0000 0.0000 3.0000 4.0000 0.0000	12.00

Problem E. Fit 5: The Beaver's Lesson

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 64 megabytes

После того, как Бобёр и Бойня были атакованы птицей Джубдзуб, для защиты от атаки Бобру надо было вслух производить арифметические действия.

Так как арифметика (а особенно в экстремальной ситуации) не была сильной стороной Бобра, а иногда для построения узоров требуются некоторые расчёты, то у бобра с собой была разновидность арифмометра — “бобрифмометр”.

Бобрифмометр состоит из ленты, поделенной на K ячеек (лента позаимствована Бобром из телеграфного аппарата, и изначально в каждой ячейке записана или точка, или тире) и каретки с карандашом, на другом конце которого имеется ластик. Каретка может перемещаться вправо и влево. Если каретка находится над ячейкой, то с её помощью Бобёр может поставить в данную ячейку тире, если там находится точка, и поставить в данную ячейку точку, если там находится тире.

Программа для бобрифмометра записывается на деревянной пластинке и кодируется в виде направляющих пазов.

Запись команд бобрифмометра для базовых операций:

```
>    переход на одну ячейку вправо
<    переход на одну ячейку влево
v    поставить тире в ячейку, над которой находится каретка
x    поставить точку в ячейку, над которой находится каретка
s    завершить работу программы
```

Кроме того, в программе могут использоваться макроопределения, которые описываются в начале программы.

Описание макроопределения представляет собой последовательность команд, которой предшествует заголовок макроопределения M <имя>. Признаком завершения макроопределения является строка с символом E . В качестве имени используется любое натуральное число меньше 10000. В описании макроопределений нельзя использовать или определять другие макросы. Макрос может быть включен в программу строкой $\#$ <имя>. Например, следующая программа:

```
M4
>
v
E
>
#4
>
>
#4
s
```

приведёт к выполнению такой последовательности команд:

```
>
>
v
```

>
>
>
v
s

Программа должна обязательно заканчивать свою работу командой “s”. Это нормальное завершение работы. Программа завершает свою работу также тогда, когда не может выполнить действие: нельзя поставить тире туда, где уже стоит тире, нельзя поставить точку туда, где уже стоит точка и нельзя сместиться за пределы ленты. В этих случаях бобрифмометр ломается. В начале работы программы бобрифмометр находится в некотором состоянии: на ленте расставлены точки и тире, а каретка находится над определенной ячейкой.

Бобру надо с помощью бобрифмометра из одного числа, записанного точками и тире на ленте, получить заданное число, написав код программы. Но проблема в том, что деревянная пластина отнюдь не бесконечна.

Напишите программу, которая составляет код для бобрифмометра, переводящий его из одного заданного состояния в другое заданное состояние (состояние — это запись на ленте + положение каретки). При этом эта программа должна завершаться нормально, а её текст вместе с макроопределениями должен содержать не более $500 + (L/2)$ строк, где L — количество символов на ленте.

После 1.000.000 операций каретка ломается и бобрифмометр можно выбрасывать, так что количество базовых операций не должно превышать 1.000.000.

Input

В первой строчке указано количество ячеек в ленте L — натуральное число, меньшее 10000. Затем идёт описание начального и конечного состояния бобрифмометра. Состояние описывается двумя строками. Первая строка описания содержит последовательность из L точек “.” или тире “-”. Вторая строка описания содержит позицию каретки. Ячейки в первой строке описания нумеруются с единицы.

Output

Ваша программа должна вывести программу для бобрифмометра.

Example

standard input	standard output
12	M1
.....-.	<
10	<
.-.-.-.-.	v
1	<
	E
	#1
	#1
	#1
	s

Problem F. Fit 6: The Barrister's dream

Input file: **standard input**
Output file: **standard output**
Time limit: **2 seconds**
Memory limit: **64 megabytes**

Барристер, проснувшись, осознал, что Благовзвон звонил не по делу, после чего снова заснул... и его сон продолжился.

Дело о свинье-дезертире, которое выиграл Снарк, было воспринято как прецедент и вошло в анналы юриспруденции. В соответствии с этим, материалы дела должны быть размещены на специальном стенде, который должен быть установлен в каждом суде (для ознакомления судей, присяжных и особенно адвокатов со столь революционным прецедентом).

Тексты выступлений должны размещаться на прямоугольных листах бумаги высоты 1 на специальном стенде заданной ширины W . Выступления со стороны защиты и обвинения (показания свидетелей, реплики сторон и так далее) должны быть размещены без перекрытий по разные стороны вертикальной перегородки. Для удобства ознакомления присяжных в каждой из частей на одной горизонтали могут помещаться не более двух выступлений (иначе говоря, никакая горизонтальная линия не может пересекать более 2 прямоугольников, находящихся в одной части).

Нужно выбрать положение разделительной перегородки и разместить тексты выступлений без перекрытий таким образом, чтобы реплики защиты оказались в правой части, а реплики обвинения — в левой. При этом, высота стенда должна быть минимальной (стенд обрежут внизу там, где заканчивается самое нижнее выступление) с целью экономии материала.

Input

В первой строке задана ширина стенда, целое число W , $1 \leq W \leq 50000$. В следующей — количества реплик защиты N_1 и обвинения N_2 , разделенные пробелом. Затем идет строка с N_1 числами — длины реплик защиты, и строка с N_2 числами — длины реплик обвинения, $0 \leq N_1, N_2 \leq 1000$. Длины реплик являются натуральными числами, не превосходящими 10000.

Output

Выведите H — минимальную высоту стенда. Если требуемое размещение невозможно, выведите 0.

Example

standard input	standard output
10	2
3 2	
5 4 4	
1 1	

Problem G. Fit 7: The Banker's Fate

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 64 megabytes

После того, как Злопастный Брандашмыг похитил Банкира, Банкир предложил ему скидку и некоторую сумму выкупа. Для того, чтобы объяснить похитителю выгоду своего предложения, Банкир произвел ряд нетривиальных арифметических действий. Банкир знает, что Брандашмыг совершенно не умеет пользоваться десятичной системой счисления, тем более ему не интересно знать тонкостей британской денежной системы. Торг происходил в некоей произвольной системе счисления, принятой на данный момент в Зазеркалье. Спустя некоторое время после инцидента с Банкиром Брандашмыг оказался пойман и предстал перед судом.

Барристер, защищавший Брандашмыга, заявил, что его подзащитному не было смысла сводить Банкира с ума, приведя в качестве вещественного доказательства выкладки Банкира. Выкладки представляли собой вычисления, сделанные Банкиром, а именно, операции сложения и умножения. Однако обвинитель заявил, что данные вычисления бессмысленны, и попросил Барристера предоставить доказательства, а именно, основание системы счисления, использованной Банкиром. Ваша задача — проверить, может ли Барристер предоставить такое основание.

Input

В первой строке N — число тождеств (N — натуральное число от 1 до 1000 включительно), затем в последующих N строках — тождества в формате

$$X_1 \dots X_l * Y_1 \dots Y_m = Z_1 \dots Z_k$$

или

$$X_1 \dots X_l + Y_1 \dots Y_m = Z_1 \dots Z_k$$

где все X_i, Y_i, Z_i — цифры от 0 до 9 или латинские буквы от a до z (цифры от 10 до 35). Числа m, l, k натуральные и меньше 11. В случае, если основание восстанавливается однозначно, все числа в примерах не превосходят $2^{64} - 1$.

Output

Вывести основание используемой во всех данных примерах системы счисления, если такая существует и восстанавливается однозначно. Выведите "0", если такой системы не существует или "-1", если система счисления не восстанавливается однозначно по данным примерам. Основание необходимо вывести в десятичном виде.

Example

standard input	standard output
3 2*2=4 25*25=625 10+10=20	10
1 2*2=5	0
1 2*2=4	-1

Problem H. Fit 8: The Vanishing

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 64 megabytes

В играх на базе D&D (Dungeons and Dragons) параметры хода определяются с помощью набора игровых костей (dice) с разным количеством граней. На грани каждой кости нанесены натуральные числа от 1 до N , где N есть число граней данной кости. Повреждение, наносимое некоторым оружием, определяется по формуле $(MdN + K)$, где MdN обозначает сумму результатов независимых бросков N -гранной кости, K — некоторое натуральное число. Например, $3d6 + 4$ обозначает, что 3 раза бросается стандартная игральная кость с 6 гранями, выпавшие очки суммируются, а затем к сумме прибавляется 4.

В игре Вы встретили хитрого монстра Буджум со следующими свойствами: если по нему нанести удар с повреждением ровно Q единиц, где Q — натуральное число, то Буджум будет уничтожен. Если же будет нанесено повреждение, отличное от Q , то Буджум выживет и в свой ход сделает так, что Вы исчезнете. Однако есть и приятные новости — Вы ходите раньше Буджума.

По заданным M , N , K и Q определите, какова вероятность уничтожения Буджума данным оружием.

Input

Параметр оружия в виде $MdN + K$, затем со следующей строки необходимое повреждение Q . Числа M , N натуральные и не больше 30. Числа K и Q натуральные и по модулю не превосходят 1000.

Output

Вероятность того, что результирующее повреждение будет равно Q в процентах, с точностью два знака после десятичной точки.

Example

standard input	standard output
1d6+1 2	16.67