

## Задача А. Выравнивание вещественных чисел

Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Вася придумал новый алгоритм сортировки вещественных чисел, который, по его мнению, будет работать быстрее, даже чем QuickSort! Он очень хочет запатентовать его, но для этого сначала необходимо представить в патентное бюро работающую версию алгоритма. К сожалению, Вася не силен в языках программирования, поэтому попросил друзей реализовать различные куски алгоритма. Вам досталась часть программы, отвечающая за выравнивание чисел.

Задана последовательность  $A_i$  положительных вещественных чисел, не превосходящих  $10^{1000}$ . Количество цифр после точки в представлении каждого числа не превосходит 1000. Требуется выписать все числа в исходном порядке в один столбец таким образом, чтобы все точки в их десятичной записи находились друг под другом. Для этого перед некоторыми числами необходимо приписать один или несколько символов #. Если решений несколько, то выведите то, в котором количество добавленных символов минимально.

### Формат входных данных

В первой строке входных данных находится целое число  $N$  — количество чисел в последовательности  $A_i$  ( $1 \leq N \leq 1000$ ). Последующие  $N$  строк содержат по одному вещественному числу последовательности  $A_i$  в десятичной записи. Числа не содержат ведущих нулей. В каждом числе присутствует десятичная точка.

### Формат выходных данных

Выходные данные должны содержать ровно  $N$  строк, по одному числу из последовательности  $A_i$  в строке. Числа должны быть выровнены по точке десятичной записи с помощью #.

### Примеры

stdin	stdout
3 3.1415926 12345.6789 2.71	###3.1415926 12345.6789 ###2.71
3 2014.09 0.001 10000.0	#2014.09 ###0.001 10000.0

## Задача В. Игра в 9

Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

В то время как Вася все пытается реализовать свой алгоритм сортировки вещественных чисел, Петя и Гена играют в занимательную игру. Они по очереди добавляют в конец числа, записанного на доске, по цифре. Игра заканчивается после того, как на доске будет написано  $N$  цифр. Если получившееся число делится на 9, то выигрывает Гена, в противном случае — Петя.

Мальчики договорились, что могут использовать не все 10 цифр, а только некоторые из них. Определите, кто выиграет при правильной игре обоих игроков.

Перед началом игры на доске не написано ни одной цифры. Петя ходит первым. Число на доске не должно содержать ведущих нулей. Гарантируется, что нет теста, в котором единственной допустимой цифрой является 0.

### Формат входных данных

Первая строка содержит целое число  $N$  ( $1 \leq N \leq 10\,000$ ).

Вторая строка содержит количество цифр  $M$  ( $1 \leq M \leq 10$ ), которые ребята договорились использовать.

Третья строка содержит  $M$  различных цифр.

### Формат выходных данных

Единственное слово — `Petya`, если выигрывает Петя, и `Gena` в случае победы Гены.

### Примеры

stdin	stdout
4 1 9	Gena
2 2 4 5	Gena
4 2 1 2	Petya

### Замечание

В первом примере, ребята договорились использовать только цифру 9. В конце игры на доске будет записано число 9999, которое делится на 9, поэтому победителем будет Гена.

Во втором примере, если Петя сначала напишет 4, то Гена допишет 5 и получится 45, которое делится на 9. Если же Петя первым ходом напишет 5, то Гена допишет 4 и получится 54, которое также делится на 9. Таким образом, всегда побеждает Гена.

В третьем примере, могут получиться следующие числа: 1111, 1112, 1121, 1122, 1211, 1212, 1221, 1222, 2111, 2112, 2121, 2122, 2211, 2212, 2221, 2222. Ни одно из них не делится на 9, поэтому победит Петя.

## Задача С. Преобразование числа

Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Гена и Петя поспорили на мороженое, кто из них двоих умнее. Для того, чтобы это выяснить Гена написал на доске целое положительное число  $N$  и хочет, чтобы Петя получил из него число  $M$ , выполняя следующую операцию. Разрешается представить число на доске в виде суммы двух натуральных чисел  $X$  и  $Y$ , а затем заменить его на произведение  $X$  и  $Y$ . Если Петя сможет получить число  $M$ , то он выиграет спор, в противном случае выиграет Гена.

Гена хочет узнать, есть ли у Пети шанс выиграть или же победа Гены неизбежна.

### Формат входных данных

В единственной строке входных данных находятся целые числа  $N$  и  $M$ , разделенные пробелом ( $1 \leq N, M \leq 10\,000$ ).

### Формат выходных данных

Выведите YES, если Петя может победить, и NO — в противном случае.

### Примеры

stdin	stdout
5 9	YES
2 10000	NO

### Замечание

В первом примере, последовательность преобразований следующая: представить 5 как  $2 + 3$  и заменить на  $6 = 2 \times 3$ , затем представить 6 как  $3 + 3$  и заменить на  $9 = 3 \times 3$ .

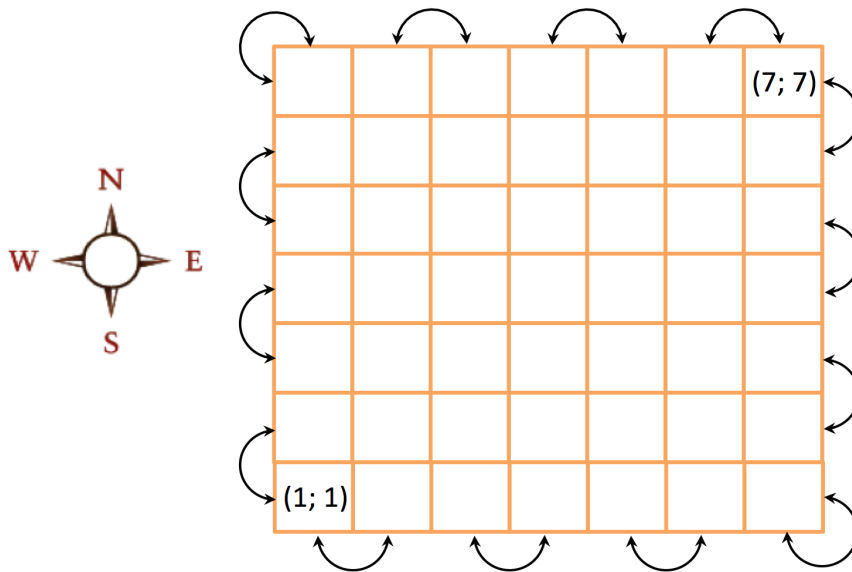
Во втором примере, число 2 можно представить только как  $1 + 1$ , поэтому на доске будет записано число  $1 = 1 \times 1$ . Число 1 нельзя представить в виде суммы двух натуральных чисел, поэтому число 10 000 получить нельзя. Значит, победит Гена.

## Задача D. Марракеш

Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Вася играет с Петей в настольную игру Марракеш. В этой игре участники по очереди передвигают по клетчатому полю  $7 \times 7$  клеток фигуру торговца Ассам. Перед броском кубика игрок выбирает, в каком направлении он хочет переместить Ассам. Торговца можно либо оставить в прежнем положении, либо повернуть на  $90^\circ$  по или против часовой стрелки (он не может поворачиваться на  $180^\circ$ ).

После выбора направления игрок бросает кубик и перемещает Ассам на столько клеток, сколько тапок изображено на выпавшей стороне кубика. Ассам перемещается по прямой (не по диагонали) в том направлении, в котором он был повернут в начале хода. Если Ассам пересекает край игрового поля, то он следует по разворотным стрелкам (при подсчете шагов стрелки не считаются за клетки).



Некоторые клетки игрового поля пусты, а на некоторых находятся ковры, принадлежащие одному из игроков. Если Ассам заканчивает свое движение на ковре противника, то игрок должен заплатить за это владельцу ковра золотые монеты. Сумма выплаты определяется путем подсчета количества взаимосвязанных клеток с коврами одного цвета, начиная с клетки, на которой остановился Ассам. Клетки считаются взаимосвязанными, если у них есть общая сторона.

Игрок ничего не платит, если Ассам заканчивает свой ход на пустой клетке или на клетке, на которой находится ковер, принадлежащий этому же игроку.

Игральный кубик имеет 6 граней, на которых нарисовано 1, 2, 2, 3, 3 и 4 тапки соответственно.

Вася хочет узнать, в какую сторону выгоднее повернуть Ассам (или оставить текущее направление движения), чтобы минимизировать среднюю сумму, которую ему придется заплатить после своего хода. Если несколько вариантов направления движения являются оптимальными, то вывести тот идентификатор направления, который следует раньше в английском алфавите.

### Формат входных данных

В первой строке заданы два целых числа  $1 \leq y, x \leq 7$  — текущее положение Ассам (номер строки и столбца соответственно). Во второй строке — одна буква-идентификатор, соответствующая направлению, в котором смотрит Ассам в настоящий момент: «N», «S», «W» или «E» (север, юг, запад или восток соответственно). Нумерация клеток начинается с 1, с левого нижнего угла.

Следующие 7 строк, по 7 символов каждая, соответствуют клеткам игрового поля. Свободной от ковров клетке соответствует символ «.», ковро Васи соответствует буква «V», ковро Пети соответствует буква «P». Никаких других символов быть не может.

## Формат выходных данных

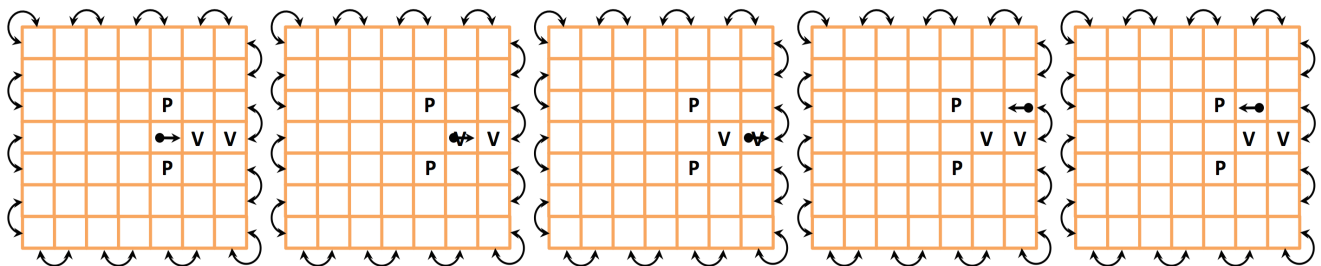
Единственную букву — направление, куда необходимо повернуть Ассаму, чтобы минимизировать среднюю величину выплаты.

## Примеры

stdin	stdout
<pre>4 5 E ..... ..... ....P.. .....VV ....P.. ..... .....</pre>	E
<pre>4 1 E ..... ..... P..... ...PP.. P..... ..... .....</pre>	N

## Замечание

В первом примере, если Вася повернет Ассаму на  $90^\circ$  против или по часовой стрелке (т.е. пойдет на север или на юг соответственно) и на кубике выпадет 1, то Вася заплатит Пете 1 монету. Если выпадет не 1, то Ассам встанет на одну из клеток, на которой нет ковра Пети, и поэтому Вася ничего не заплатит. В том случае, если Вася не будет менять направление движения Ассамы (этот случай проиллюстрирован на рисунке ниже) и на кубике выпадет 1 или 2, Ассам окажется в клетках с координатами (4, 6) или (4, 7) соответственно, и Вася ничего не заплатит, так как на этих клетках лежат его ковры. При выпадении же 3 или 4, Ассам окажется в клетках с координатами (5, 7) или (5, 6) соответственно, и Вася так же ничего не заплатит, поскольку торговец встанет на пустые клетки.



Во втором примере, если Вася повернет Ассаму в любую из сторон, и на кубике выпадет 1, то он заплатит Пете одну золотую монету, а в остальных случаях — ничего не заплатит. Если же Вася не будет менять направление движения и на кубике выпадет 3 или 4, то Вася заплатит по 2 золотые монеты в каждом случае. Получается, что одинаково выгодно повернуть Ассаму на юг («S») и на север («N»). Так как буква «N» идет раньше в английском алфавите чем «S», то она и является ответом.

## Задача Е. Мухи на плоскости

Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Школьник Петя сидит на уроке физики и скучает. Он начертил в своей тетрадке в клетку две координатные оси и смотрит, как по странице ползают мухи.

На нарисованной Петей координатной плоскости сидят три маленькие мухи. Муха с номером  $i$  сидит в точке с координатами  $x_i, y_i$ . Никакие две точки из этих трех не совпадают. В момент времени  $t = 0$  мухи начинают двигаться с постоянными скоростями — скорость  $i$ -й мухи задана вектором с координатами  $v_i^x, v_i^y$  (координаты вектора скорости — это то же самое, что и значения его проекций на соответствующие оси координат). В момент времени  $t = t_e$  секунд звенит звонок, Петя хватается тетрадь, мухи взлетают и покидают координатную плоскость.

Видел ли Петя во время урока, что мухи находятся на одной прямой? Размерами самих мух в этой задаче можно пренебречь и считать их точками.

### Формат входных данных

В первой строке задано одно целое число  $t_e$  ( $1 \leq t_e \leq 3600$ ) — время в секундах до конца урока.

Затем три строки, по четыре целых числа в каждой:  $x_i, y_i, v_i^x, v_i^y$  ( $-1000 \leq x_i, y_i, v_i^x, v_i^y \leq 1000$ ) — начальные координаты и компоненты скорости каждой мухи соответственно.

### Формат выходных данных

Выведите самое раннее время (в секундах) от момента начала наблюдений Петей, когда он мог увидеть всех мух на одной прямой. Ваш ответ должен отличаться от правильного не более чем на  $10^{-6}$  секунды.

Если же Петя ни разу не увидел всех мух на одной прямой во время урока, то выведите  $-1$ .

### Примеры

stdin	stdout
2 0 0 1 2 0 7 3 -3 8 7 -2 -1	1.06314329

## Задача F. Фоторамка

Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Вася выиграл в школьной викторине приз — рамку для фотографий, причем не простую, а такую, в которую можно вставить сразу несколько фотографий. У Васиной рамки есть четыре места под фотографии, расположенные в ряд.

В век цифрового фото у Васи не так много бумажных фотографий, которые он мог бы вставить в рамку — всего  $N$  штук. Мальчик планирует менять их состав каждый день, и ему интересно, как долго он сможет это делать. Для этого ему надо подсчитать, сколько есть способов заполнить рамку имеющимися у него фотографиями. Отметим, что Вася считает различными и способы, отличающиеся только порядком фотографий в рамке.

### Формат входных данных

На единственной строке записано одно целое число  $N$  ( $1 \leq N \leq 20$ ) — количество фотографий, которые есть у Васи.

### Формат выходных данных

Выведите единственное число — количество различных способов заполнить Васиными фотографиями рамку.

### Примеры

stdin	stdout
4	24
6	360

## Задача G. Сны о скобках

Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Однажды учитель информатики Петр Алексеевич рассказывал на уроке про правильные скобочные последовательности.

— Скобочной последовательностью, дети, называется любая строка, которая состоит только из открывающих и закрывающих скобок «(» и «)». Правильной же скобочной последовательностью или, сокращенно, ПСП называется такая последовательность, которую можно получить вычеркиванием из какого-нибудь математического выражения всего, кроме скобок. Проще говоря, для каждой открывающей скобки найдется закрывающая и наоборот. Например «()», «(())()», «(())» являются ПСП, а «(())» или «)()» — нет.

Вася многократно участвовал в олимпиадах по программированию и уже знал все про ПСП, поэтому ему стало скучно и он заснул. Ему приснилось, что он в математической стране, разгружает вагон со скобочными последовательностями. Очередная последовательность вырвалась у него из рук и упала. Вася заметил, что одна из скобок разбилась. К счастью, неподалеку лежит кучка запасных скобок. Но откуда выпала разбившаяся скобка? Вася не заметил. Впрочем, он надеется, что поврежденная скобочная последовательность была правильной и он сможет подобрать место, куда нужно вставить скобку. Но каковы его шансы на успех?

Чтобы ответить на этот вопрос, Васе нужно посчитать, сколько различных правильных скобочных последовательностей он может получить из данной произвольной скобочной последовательности, дописав всего одну открывающую или закрывающую скобку.

### Формат входных данных

В первой строке входных данных находится число  $N$  ( $1 \leq N \leq 100\,000$ ) — длина данной Васе строки. Вторая строка состоит из  $N$  открывающих и закрывающих скобок «()».

### Формат выходных данных

Выведите единственное число — количество различных правильных скобочных последовательностей, которые может получить Вася.

### Примеры

stdin	stdout
3 (()	2
4 (())	0

### Замечание

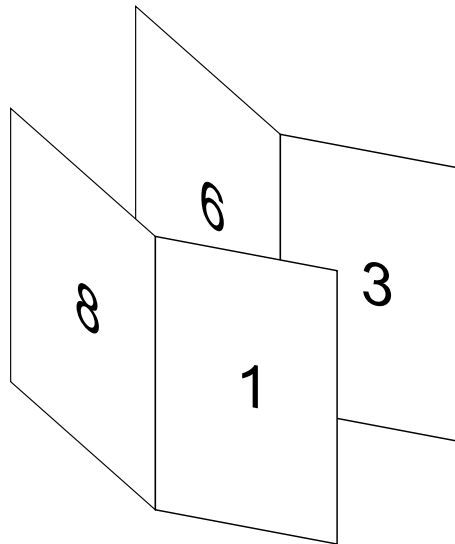
В первом примере Вася может получить две ПСП: «()()» и «(())». Во втором примере Вася не может получить ПСП из данной последовательности.



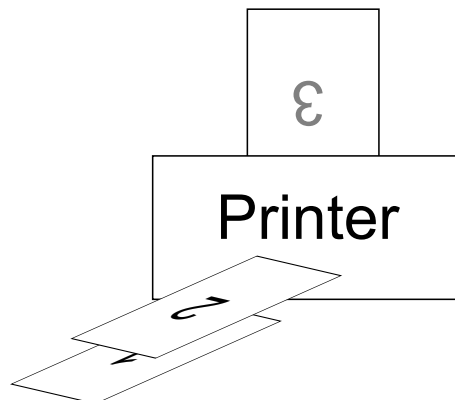
## Задача Н. Спуск полос

Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

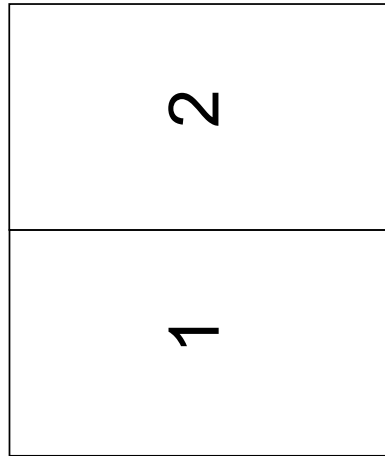
Вася помогает своей учительнице Марии Ивановне подготовить брошюру к 100-летию школы. Мария Ивановна подготовила и сверстала брошюру в своем любимом текстовом редакторе World. Но вот незадача — вся верстка сделана на листах А4, а напечатать нужно настоящую книжку, причем в формате А5. «Это не проблема», — заявляет Вася, и пробует печатать по две страницы на листе. Формат совпадает, но скрепить эти листы нельзя — в книжке они окажутся не в том порядке. Листы скрепляются как в тонкой тетрадке, посередине:



Эксперимент показал Васе, что ему надо печатать страницы в нетривиальном порядке. Но ему не под силу вручную правильно перечислить в окне настроек печати World все сотни страниц брошюры. Поэтому он попросил вас написать программу, которая подготовит это перечисление за него. Вася печатает страницы с обеих сторон листа, поэтому ему нужно будет два перечисления: для лицевых сторон и для обратных сторон. Чтобы печатать с обратной стороны, Вася берет пачку отпечатанных с лицевой стороны листов и переворачивает ее сверху вниз (по короткой стороне листа); при этом он не меняет порядок листов в пачке. Принтер Васи ориентирует бумагу следующим образом (на рисунке первая и вторая страница напечатаны, третья печатается):



При настройке печати в редакторе World достаточно перечислить страницы через пробел. Если перечислена пара страниц 1 2, то на печать будет выведено:



Если в брошюре не хватает текста, чтобы заполнить последние страницы, то их можно оставить пустыми — главное, чтобы получилась тетрадка.

### Формат входных данных

На вход вашей программе подается единственное целое число  $N$  ( $1 \leq N \leq 1000$ ) — количество страниц в брошюре.

### Формат выходных данных

Выведите две строки, содержащие последовательности страниц для принтера. В первой строке выведите номера страниц, печатаемых на лицевой стороне, во второй — на оборотной, разделяя номера пробелами. Если при печати листа требуется оставить одну из его половин пустой — выведите вместо номера страницы символ - (дефис).

### Примеры

stdin	stdout
8	8 1 6 3 2 7 4 5
6	- 1 6 3 2 - 4 5

## Задача I. Иззи

Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Вася вернулся с международной олимпиады школьников по программированию (IOI) и привез с собой  $N$  разноцветных камней в качестве сувениров. Вася совсем не жадный мальчик, поэтому решил поделиться камнями со своими друзьями. Каждому другу Вася отдал ровно один камень. Оказалось, что у самого Васи остался тоже только один камень. Определите, сколько же у Васи друзей?

### Формат входных данных

В первой строке входного файла находится число  $N$  ( $1 \leq N \leq 100$ ) — количество камней, привезенных Васей.

### Формат выходных данных

Выведите единственное число — количество друзей Васи.

### Примеры

stdin	stdout
2	1

### Замечание

В примере Вася привез 2 камня, один из которых остался у него. Значит, второй камень Вася отдал своему единственному другу.

## Задача J. Затмение

Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Вася очень любит астрономию. Он решил создать модель затмения одной планеты другой и сфотографировать это явление. Для этого он сделал две звезды и каждой звезде добавил планету, вращающуюся вокруг этой звезды. В проекции на плоскость объектива фотоаппарата можно считать, что планеты вращаются вокруг своих звезд по окружности с центром в соответствующей звезде, причем скорость вращения для каждой планеты постоянна (однако, скорости вращения планет, а так же радиусы орбит могут различаться между собой). Планеты вращаются вокруг звезд по часовой стрелке. Каждая планета в модели Васи представляет собой круг некоторого радиуса.

Вася установил начальное положение планет и звезд, после чего планеты пришли в движение по окружностям с соответствующими скоростями. Положение планет Вася считает «интересным», если проекция центра хотя бы одной планеты лежит на прямой, соединяющей проекции центров двух звезд. В процессе движения проекции кругов-планет могут пересекаться. Помогите Васе определить, какая наибольшая площадь пересечения проекций планет может быть во время «интересного» положения планет. Можно считать, что после начального положения планеты будут двигаться бесконечно долго в соответствии с описанными условиями.

### Формат входных данных

В первой строке содержится единственное число  $D$  — расстояние между центрами звезд. Считается, что первая звезда имеет координаты  $(0, 0)$ , а вторая —  $(D, 0)$ . Во второй строке заданы четыре числа:  $X_1, Y_1$  — начальные координаты первой планеты,  $R_1$  — радиус первой планеты,  $V_1$  — линейная скорость первой планеты. В третьей строке описана вторая планета в таком же формате.

Все числа целые и не превосходят 100 по абсолютному значению. Радиусы планет и скорости планет строго положительны. Число  $D$  неотрицательно. Начальное положение центра каждой планеты отличается от положения центра соответствующей ей звезды, однако, центр звезды может находиться внутри планеты достаточно большого радиуса (см. первый пример).

### Формат выходных данных

Вывести единственное число — максимальную возможную площадь пересечения проекций планет в «интересном» положении.

### Примеры

stdin	stdout
5 1 0 2 1 3 0 2 3	4.913478794
5 0 1 2 1 5 1 2 1	0.000000000

### Замечание

В первом примере начальное положение планет дает максимально возможную площадь пересечения.

Во втором примере скорости планет и радиусы их орбит одинаковы, планеты движутся параллельно друг другу, не пересекаясь.

## Задача К. Парковка

Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Вася сделал покупки в популярном торговом центре и теперь хочет отправиться домой. Для этого ему сначала нужно выехать на своей машине с парковки. Задача эта не очень простая, поскольку на парковке находится множество машин других посетителей торгового комплекса. Сама же парковка представляет собой прямоугольную область  $N \times M$  клеток. В одной из клеток находится машина Васи. В других клетках могут находиться машины остальных покупателей. Кроме того, в некоторых клетках (по крайней мере, в одной) находятся выезды с парковки.

Вася хочет доехать до одного из выездов с парковки. Вася может перемещаться на машине из текущей клетки в соседнюю по стороне клетки, если эта клетка находится на парковке и свободна от других машин. В процессе переезда Вася может увеличивать скорость автомобиля максимум на  $A$  или уменьшать скорость автомобиля максимум на  $B$ . Итого, если до переезда скорость автомобиля была равна  $V$ , то после переезда на новую клетку скорость может быть любым (не обязательно целым) числом в интервале  $[V - B, V + A]$  по желанию Васи, с единственным ограничением — скорость не может быть ниже 1. Если в начале переезда скорость машины была равна  $V_1$ , а после — стала равна  $V_2$ , то будем считать, что Вася потратил  $2/(V_1 + V_2)$  времени на переезд. Кроме того, Вася может менять направление движения, только если скорость автомобиля равна 1. Это значит, что если Вася уже совершил переезд из одной клетки в другую и его скорость после переезда больше 1, то он может продолжать двигаться только в том же направлении. Если же следующая клетка в этом направлении занята машиной или находится за пределами парковки, то такое движение запрещено. Однако приезжать к выезду с парковки можно на любой скорости (даже если следующая клетка в направлении движения занята).

Определите, за какое наименьшее время Вася сможет добраться до какого-либо из выездов с парковки. Покидать парковку, выходя за пределы исходной области  $N \times M$ , запрещено. Машины других покупателей не двигаются. Начальная скорость машины Васи равна 1.

### Формат входных данных

В первой строке входного файла находятся 4 целых числа — размеры парковки  $N$  и  $M$  ( $1 \leq N, M \leq 200$ ), а так же числа  $A$  и  $B$  ( $1 \leq A, B \leq 100$ ). Следующие  $N$  строк описывают начальное состояние парковки. Каждая строка имеет длину  $M$  и состоит из символов '.', 'C', 'V', 'E'. Символ '.' соответствует пустой клетке парковки, 'C' — занятой автомобилем, 'V' — клетка, в которой находится машина Васи, 'E' — клетки, представляющие собой выезды с парковки. Гарантируется, что присутствует ровно один символ 'V', не менее одного символа 'E' и что Вася может добраться хотя бы до одного выезда с парковки.

### Формат выходных данных

Вывести единственное вещественное число с абсолютной или относительной погрешностью, не превышающей  $10^{-6}$  — минимальное время, за которое Вася сможет добраться до одного из выездов с парковки.

### Примеры

stdin	stdout
2 2 2 2 VE ..	0.500000000
2 3 1 1 V.. ..E	2.000000000
2 3 1 1 VC. ..E	2.066666667

## Замечание

В первом примере Вася при переезде с клетки  $(1, 1)$  в клетку  $(1, 2)$ , содержащую выезд с парковки, может увеличить скорость с 1 до 3. При этом он потратит время, равное  $2/(1 + 3) = 0.5$ .

Во втором примере оптимально двигаться из клетки  $(1, 1)$  в клетку  $(1, 2)$ , увеличив скорость до 2. Затем — в клетку  $(1, 3)$ , снизив скорость до 1 из-за поворота. Далее — в клетку  $(2, 3)$ , увеличив скорость до 2. Итого, будет потрачено время  $2/(1 + 2) + 2/(2 + 1) + 2/(1 + 2) = 2$ .

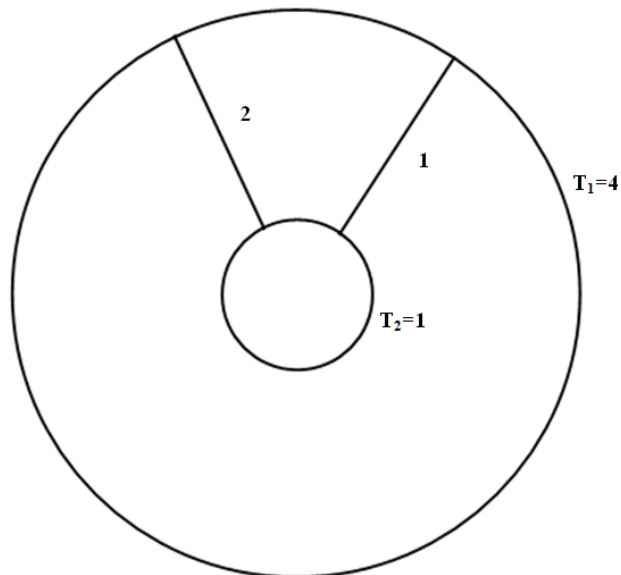
В третьем примере нельзя двигаться так же, как в предыдущем, поскольку клетка  $(1, 2)$  занята другой машиной. Необходимо двигаться сначала в клетку  $(2, 1)$  на скорости 1, затем повернуть на клетку  $(2, 2)$ , увеличив скорость до 2, далее — в клетку  $(2, 3)$ , увеличив скорость до 3. Итого, Вася потратит  $2/(1 + 1) + 2/(1 + 2) + 2/(2 + 3) = 2\frac{1}{15}$  единиц времени.

## Задача Л. Пушка Гаусса

Ограничение по времени: 2 секунды  
Ограничение по памяти: 256 мегабайт

Вася очень любит физику. Освоив основы электродинамики, он решил создать собственную модель гаусс-пушки. Ключевыми элементами пушки являются ускорители, позволяющие снаряду приобрести огромную начальную скорость. Каждый ускоритель содержит две концентрические окружности — основные кольца. Кольца ускорителя соединены некоторым количеством резисторов. Каждый из резисторов представляет собой отрезок, соединяющий два кольца, который лежит на прямой, проходящей через центры колец. Для проверки качества ускорителя необходимо уметь определять, за какое время заряд может пройти от одной точки основного кольца, до другой точки (возможно, другого) основного кольца. Заряд может двигаться по первому основному кольцу, тратя  $T_1$  единиц времени на прохождение одного радиана, по второму кольцу — тратя  $T_2$  единиц времени на радиан. Кроме того, заряд может проходить через резисторы, тратя  $T$  единиц времени на каждый резистор. Как по кольцам, так и по резисторам допускается движение в любом направлении. Разумеется, при переходе через резистор заряд окажется на другом основном кольце.

Конфигурация ускорителя, соответствующая примеру, приведена на картинке ниже.



Зная параметры  $T_1$ ,  $T_2$  и  $T$ , а также конфигурацию ускорителя (положение резисторов), определите время прохождения заряда между указанными парами точек. Толщиной колец и резисторов можно пренебречь.

### Формат входных данных

В первой строке находятся 3 целых числа —  $T_1$ ,  $T_2$  и  $T$  ( $1 \leq T_1, T_2, T \leq 1000$ ). Во второй строке — единственное число  $N$  — количество резисторов в ускорителе ( $1 \leq N \leq 200000$ ). В следующей строке находятся  $N$  вещественных чисел  $A_i$ , содержащих ровно 4 знака после десятичной точки, — углы положения резисторов. Углы заданы в радианах,  $0 \leq A_i < 2\pi$ , все числа  $A_i$  различны.

В следующей строке задано единственное число  $M$  — количество запросов на нахождение времени прохождения заряда ( $1 \leq M \leq 100000$ ). В каждой из  $M$  следующих строк находятся 4 числа —  $C_1$ ,  $X_1$ ,  $C_2$ ,  $X_2$  — описание начальной и конечной точек движения заряда.  $C_i$  равно 1, если точка находится на первом кольце, 2 — если на втором кольце.  $X_i$  — угол (в радианах) положения точки на кольце,  $0 \leq X_i < 2\pi$ ,  $X_i$  содержит ровно 4 знака после десятичной точки.

## Формат выходных данных

Для каждого из  $M$  запросов нужно вывести одно вещественное число на отдельной строке — минимальное время, которое потребуется заряду для перехода между двумя соответствующими точками. Ответ должен быть дан с абсолютной или относительной погрешностью, не превышающей  $10^{-6}$ .

## Примеры

stdin	stdout
4 1 2	6.000000000
2	4.000000000
1.0000 2.0000	4.800000000
3	
1 0.0000 1 1.5000	
1 1.6000 2 1.6000	
1 1.6000 2 0.6000	

## Замечание

В примере имеется 2 резистора, в позиции 1 радиан и 2 радиана.

В первом запросе нужно пройти между двумя точками первого кольца. Расстояние между точками — 1.5 радиан. Поэтому потребуется время  $4 \cdot 1.5 = 6$ .

Во втором запросе нужно из точки первого кольца перейти в точку второго кольца. Быстрее всего сделать это, пройдя через второй резистор. Для этого потребуется  $4 \cdot (2 - 1.6) + 2 + 1 \cdot (2 - 1.6) = 4$  единицы времени.

В последнем запросе, в отличие от предыдущего, выгоднее двигаться через первый резистор. Искомое время равно  $4 \cdot (1.6 - 1) + 2 + (1 - 0.6) = 4.8$ .