

## Задача А. Золотая осень

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 256 megabytes

Какое прекрасное время года — золотая осень! Девочка Алена ходит по лесу и собирает букет из опавших листьев. Алена очень избирательна — она не берет листочек, если он совпадает одновременно и по виду, и по цвету с одним из уже имеющихся у нее. Выясните, сколько листьев собрала Алена.

### Формат входного файла

В первой строке содержится целое число  $n$  ( $1 \leq n \leq 100$ ) — количество листьев, попавшихся Алене. Следующие  $n$  строк содержат описания листьев. Каждый лист характеризуется видом дерева, с которого он упал, и цветом. Виды деревьев и цвета заданы названиями, состоящими из не более 10 строчных латинских букв. Название не может быть пустой строкой. Вид дерева и цвет задаются в каждой строке через пробел.

### Формат выходного файла

Выведите одно число — количество листьев у Алены.

### Примеры

| <code>stdin</code>  | <code>stdout</code> |
|---|---------------------|
| 5<br>birch yellow<br>maple red<br>birch yellow<br>maple yellow<br>maple green | 4                   |
| 3<br>oak yellow<br>oak yellow<br>oak yellow                                   | 1                   |

## Задача В. Кола

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 256 megabytes

По торжественному случаю открытия Зимней Компьютерной Школы организаторы решили купить  $n$  литров колы. Однако в магазине возникло неожиданное затруднение: оказалось, что кола продается в бутылках по 0.5, 1 и 2 литра. При этом имеется ровно  $a$  бутылок по 0.5 литра,  $b$  литровых бутылок и  $c$  — двухлитровых. У организаторов достаточно денег, чтобы купить любое количество колы. Ожесточенные споры вызвало то, сколько каких бутылок покупать, ведь этот вопрос имеет принципиальное значение с точки зрения распределения колы между участниками (и организаторами тоже).

Итак, пока организаторы спорят, перебирая различные варианты закупки колы, Зимняя Компьютерная Школа не начнется. Ваша задача — посчитать количество всех возможных способов закупить ровно  $n$  литров колы и убедить организаторов, что это количество слишком велико и что если они будут продолжать свой спор, то Зимнюю Компьютерную Школу придется проводить летом.

Все бутылки колы одного объема считаются неразличимыми. Т.е. два варианта закупки отличаются друг от друга только в случае, когда они отличаются количеством бутылок хотя бы одного вида.

### Формат входного файла

В первой строке заданы четыре целых числа —  $n, a, b, c$  ( $1 \leq n \leq 10000, 0 \leq a, b, c \leq 5000$ ).

### Формат выходного файла

Выведите единственное число — ответ на задачу. Если купить ровно  $n$  литров колы невозможно, выведите 0.

### Примеры

| <code>stdin</code> | <code>stdout</code> |
|--------------------|---------------------|
| 10 5 5 5           | 9                   |
| 3 0 0 2            | 0                   |

## Задача С. Каникулы

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 256 megabytes

В Берляндии наступают каникулы, которые будут длиться  $n$  дней. Ученики школы № $N$  отдыхают, а учительница информатики Марина Сергеевна, которая все лето была занята проверкой БГЭ (Берляндского государственного экзамена), наконец-то взяла отпуск! Для ежедневного полива цветов в классе был составлен график дежурств. Однако составлявшая график Марина Сергеевна так устала от работы и была поглощена мечтами о предстоящем отпуске, что, возможно, допустила некоторые ошибки. А именно, могло случиться так, что согласно графику в некоторые дни каникул цветы не будут политы или будут политы несколько раз. Помогите Марине Сергеевне найти ошибку.

### Формат входного файла

В первой строке заданы два числа  $n$  и  $m$  ( $1 \leq n, m \leq 100$ ) — количество дней в берляндских каникулах и количество дежурных соответственно. Следующие  $m$  строк содержат описание графика дежурств. Каждая строка содержит два целых числа  $a_i$  и  $b_i$  ( $1 \leq a_i \leq b_i \leq n$ ), означающих, что  $i$ -й дежурный должен поливать цветы с  $a_i$ -го по  $b_i$ -й день включительно, по одному разу в день. Дежурства описываются по порядку, т.е.  $b_i \leq a_{i+1}$  для всех  $i$  от 1 до  $n - 1$  включительно.

### Формат выходного файла

Выведите «ОК» (без кавычек), если график не содержит ошибок. Иначе нужно найти наименьший номер дня, в который цветы не будут политы или будут политы несколько раз, и вывести два целых числа — номер дня и сколько раз будут политы цветы в этот день.

### Примеры

| stdin                                    | stdout |
|--|--------|
| 10 5<br>1 2<br>3 3<br>4 6<br>7 7<br>8 10 | OK     |
| 10 5<br>1 2<br>2 3<br>4 5<br>7 8<br>9 10 | 2 2    |
| 10 5<br>1 2<br>3 3<br>5 7<br>7 7<br>7 10 | 4 0    |

### Note

Обратите внимание, во втором примере из условия ошибка происходит не только во второй день, но и в шестой, когда цветы никто не поливает. Однако выдать нужно именно второй день, т.е. день с наименьшим номером.

## Задача D. Гипердвигатель

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 256 megabytes

В далекой-далекой галактике  $n$  обитаемых планет, пронумерованных числами от 1 до  $n$ . Они расположены на огромных расстояниях друг от друга, поэтому сообщение между ними было крайне затруднено, пока на планете с номером 1 не был изобретен гипердвигатель. Как только произошло это знаменательное событие, на планете 1 был построен  $n - 1$  космический корабль, и эти корабли были разосланы на остальные планеты, чтобы сообщить о революционном изобретении.

Гиперпространство, как это ни парадоксально, представляет собой обычное трехмерное евклидово пространство. Обитаемые планеты можно считать неподвижными точками в нем, причем никакие две точки не совпадают и никакие три точки не лежат на одной прямой. Движение корабля, оснащенного гипердвигателем, между двумя планетами осуществляется по прямой с постоянной скоростью, одинаковой для всех кораблей. Поэтому расстояния в гиперпространстве измеряются в гипергодах (корабль, оснащенный гипердвигателем, преодолевает расстояние в  $s$  гипергодов за  $s$  лет).

Когда корабль прилетает на обитаемую планету, жители планеты разбирают его, создают по его образу и подобию  $n - 2$  корабля с гипердвигателем и рассылают их на остальные  $n - 2$  планеты (кроме той, с которой прилетел корабль). Время создания новых кораблей по сравнению со временем их движения от одной планеты к другой настолько мало, что им можно пренебречь. Новые корабли абсолютно идентичны разосланным первоначально: они движутся с той же постоянной скоростью по прямолинейной траектории и, прибывая на планету, выполняют ту же миссию, т.е. их разбирают, строят новые  $n - 2$  корабля и рассылают их на все планеты, кроме той, с которой прилетел корабль. Таким образом, процесс распространения важной новости по галактике продолжается.

Однако создатели гипердвигателя так торопились распространить известие о своем открытии, что не изучили до конца, что происходит при столкновении двух кораблей в гиперпространстве. Если два движущихся корабля оказываются в одной точке, то происходит взрыв колоссальной силы, приводящий к уничтожению галактики!

Ваша задача — вычислить время жизни галактики с момента запуска кораблей с первой планеты.

### Формат входного файла

В первой строке задано число  $n$  ( $3 \leq n \leq 5000$ ) — количество обитаемых планет в галактике. Следующие  $n$  строк содержат целочисленные координаты планет в формате « $x_i y_i z_i$ » ( $-10^4 \leq x_i, y_i, z_i \leq 10^4$ ).

### Формат выходного файла

Выведите единственное число — ответ на задачу с абсолютной или относительной погрешностью, не превосходящей  $10^{-6}$ .

### Примеры

| stdin | stdout       |
|-------|--------------|
| 4     | 1.7071067812 |
| 0 0 0 |              |
| 0 0 1 |              |
| 0 1 0 |              |
| 1 0 0 |              |

## Задача E. Обезьянка Анфиса

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 256 megabytes

Обезьянка Анфиса учится печатать на клавиатуре. Пока она не знакома с клавишей «пробел» и умеет набирать только строчные латинские буквы. Набрав достаточно длинную строку, Анфиса поняла, что неплохо бы разбить ее на  $k$  строк длины не меньше  $a$  и не больше  $b$ , чтобы ее текст стал более похож на человеческий. Помогите Анфисе.

### Формат входного файла

В первой строке заданы три целых числа  $k$ ,  $a$  и  $b$  ( $1 \leq k \leq 200$ ,  $1 \leq a \leq b \leq 200$ ). Во второй строке содержится последовательность строчных латинских букв — текст, набранный Анфисой. Гарантируется, что заданная строка не пустая и ее длина не превосходит 200 символов.

### Формат выходного файла

Выведите  $k$  строк, каждая из которых содержит не менее  $a$  и не более  $b$  символов — текст Анфисы, разбитый на строки. Производить любые изменения текста: удалять или добавлять символы, менять их порядок и т.п. не разрешается. Если решений несколько, выведите любое. Если решения не существует, выведите «No solution» (без кавычек).

### Примеры

| stdin                | stdout              |
|----------------------|---------------------|
| 3 2 5<br>abrakadabra | ab<br>rakad<br>abra |
| 4 1 2<br>abrakadabra | No solution         |

## Задача F. VerPaint

|                         |                     |
|-------------------------|---------------------|
| Имя входного файла:     | <code>stdin</code>  |
| Имя выходного файла:    | <code>stdout</code> |
| Ограничение по времени: | 5 seconds           |
| Ограничение по памяти:  | 256 megabytes       |

Обезьянка Анфиса разочаровалась в текстовых редакторах, потому что они недостаточно хороши, чтобы передать всю полноту переживаемых ею эмоций, поэтому она решила переключиться на графические редакторы. Открыв редактор VerPaint, она увидела белый прямоугольник размера  $W \times H$ , на котором можно рисовать. Первым делом Анфиса освоила чертежный инструмент, позволяющий рисовать отрезки, и быстро набросала на этом прямоугольнике некоторое число отрезков черного цвета. Получившаяся картина показалась Анфисе недостаточно яркой, поэтому она обратила внимание на инструмент «Заливка», который позволяет указать точку в прямоугольнике для рисования и выбрать цвет, после чего вся область, имеющая тот же цвет, что и выбранная точка, и содержащая ее, полностью окрашивается в выбранный цвет. Применяв заливку много раз в разных точках, Анфиса полностью выразила всю свою экспрессию и бросила рисование. От вас же требуется по информации о нарисованных отрезках и проведенных заливках определить для каждого цвета, какова суммарная площадь областей, окрашенных в этот цвет после всех заливок.

### Формат входного файла

В первой строке заданы два целых числа  $W$  и  $H$  ( $3 \leq W, H \leq 10^4$ ) — размеры изначально белой прямоугольной области для рисования. Во второй строке записано целое число  $n$  — количество черных отрезков ( $0 \leq n \leq 100$ ). В следующих  $n$  строках описаны сами отрезки, каждый из которых задается координатами своих концов  $x_1, y_1, x_2, y_2$  ( $0 < x_1, x_2 < W, 0 < y_1, y_2 < H$ ). Все отрезки имеют ненулевую длину. В следующей строке задано число заливок  $m$  ( $0 \leq m \leq 100$ ). Каждая из следующих  $m$  строк задает операцию заливки в виде « $x y color$ », где  $(x, y)$  это координаты выбранной точки ( $0 < x < W, 0 < y < H$ ), а  $color$  — строка из маленьких латинских букв длиной от 1 до 15 символов, задающая цвет. Изначально прямоугольник имеет цвет «white», а отрезки наносятся цветом «black». Координаты концов всех отрезков и координаты точек заливки целые.

### Формат выходного файла

Для каждого цвета, который присутствует на итоговой картинке, на отдельной строке выведите название этого цвета и суммарную площадь областей, окрашенных в этот цвет, с точностью  $10^{-6}$ . Цвета выдавайте в любом порядке.

## Примеры

| stdin  | stdout                                |
|--|---------------------------------------|
| 4 5<br>6<br>1 1 1 3<br>1 3 3 3<br>3 3 3 1<br>3 1 1 1<br>1 3 3 1<br>1 1 3 3<br>2<br>2 1 red<br>2 2 blue                                   | blue 0.00000000<br>white 20.00000000  |
| 5 5<br>5<br>1 1 2 2<br>2 2 4 2<br>4 2 4 4<br>4 4 2 4<br>2 4 2 2<br>2<br>3 3 black<br>3 3 green   | green 4.00000000<br>white 21.00000000 |
| 7 4<br>9<br>1 2 2 3<br>2 3 3 2<br>3 2 2 1<br>2 1 1 2<br>3 2 4 2<br>4 2 5 3<br>5 3 6 2<br>6 2 5 1<br>5 1 4 2<br>2<br>2 2 black<br>2 2 red | red 2.00000000<br>white 26.00000000   |

## Note

Изначально черные отрезки, которые нарисовала Анфиса, также могут быть окрашены, если какая-то из выбранных точек попадет на отрезок. Отрезки имеют нулевую площадь. Поэтому, если в итоговой картинке в какой-то цвет будут окрашены только лишь части отрезков, то площадь, окрашенная в этот цвет, равна 0.

## Задача G. Тир

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 10 seconds  
Ограничение по памяти: 256 megabytes

Тир в Берляндском парке культуры и отдыха по праву считается одним из лучших в мире. Ежедневно лучшие стрелки страны оттачивают там свое мастерство, а многочисленные посетители соревнуются в стрельбе по мишеням в борьбе за достойные призы. А недавно директор парка решил сделать онлайн-версию тира. В процессе разработки выяснилось, что нужна программа, эффективно моделирующая процесс стрельбы. Чтобы сформулировать требования к программе, тир был описан формально. Была введена трехмерная декартова система координат, в которой ось  $X$  проходила по полу тира по линии, вдоль которой стоят стрелки, ось  $Y$  проходила вертикально по стене тира, а положительное направление оси  $Z$  совпадало с направлением стрельбы. Плоскость  $XOY$  назовем плоскостью стрельбы и будем считать, что все пули выходят из дул в точках этой плоскости и летят параллельно оси  $Z$ . Каждая мишень представляет собой прямоугольник, стороны которого параллельны осям  $X$  и  $Y$ , имеющий положительную  $z$ -координату. Все мишени находятся на разном расстоянии от плоскости стрельбы. Пуля попадает в мишень, если она проходит через внутренность или границу соответствующего ей прямоугольника. Когда пуля поражает мишень, мишень падает вертикально вниз, в подпол тира, и больше по ней стрелять нельзя. Мишени достаточно прочные, поэтому пуля не может пробить мишень насквозь, и в случае попадания пуля не летит дальше. На вход программе-симулятору дается расположение всех мишеней, а также координаты всех выстрелов в порядке их появления. Программа должна определить, какая мишень была сбита каждым выстрелом. Если вы еще не догадались, то написать такую программу поручается вам.

### Формат входного файла

В первой строке записано целое число  $n$  ( $1 \leq n \leq 10^5$ ) — количество мишеней. Каждая из последующих  $n$  строк содержит описание одной мишени. Мишень описывается пятью целыми числами  $x_l, x_r, y_l, y_r, z$ , задающими ее расположение в пространстве ( $0 \leq x_l < x_r \leq 10^7, 0 \leq y_l < y_r \leq 10^7, 0 < z \leq 10^7$ ). В следующей строке записано целое число  $m$  ( $1 \leq m \leq 10^5$ ), задающее количество выстрелов. Далее в  $m$  строках описаны выстрелы. Каждый выстрел задается координатами пули в плоскости стрельбы  $(x, y)$  ( $0 \leq x, y \leq 10^7$ , координаты пули целые). Выстрелы задаются в том порядке, в каком они производились. Интервалы между выстрелами достаточно большие, а мишень падает очень быстро, поэтому считайте, что падающая мишень не может преградить путь выстрелам, следующим за тем, который ее сбил.

### Формат выходного файла

Для каждого выстрела в отдельной строке выведите номер мишени, которую поразил этот выстрел, или 0, если пуля не попала ни в какую мишень. Мишени нумеруются с 1 в том порядке, в каком они заданы во входных данных.

### Примеры

| stdin     | stdout |
|-----------|--------|
| 2         | 0      |
| 1 4 1 4 1 | 1      |
| 2 5 2 6 2 | 2      |
| 4         | 0      |
| 0 0       |        |
| 3 3       |        |
| 4 5       |        |
| 3 5       |        |



## Задача N. Номер телефона

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 2 seconds  
Ограничение по памяти: 256 megabytes

Найти свою любовь, увы, непросто. Маше пока в этом не везет. Подружка Даша рассказала Маше способ гадания, с помощью которого можно определить номер телефона своего прекрасного принца.

Этот способ заключается в следующем. Сначала нужно выписать свой номер телефона. Например, предположим, что номер телефона Маши 12345. После этого нужно написать свою любимую цифру от 0 до 9 под первой цифрой своего номера. Это будет первая цифра желаемого номера. Например, Машина любимая цифра — 9. Вторая цифра определяется как среднее арифметическое второй цифры Машиного номера и уже записанной первой цифры номера ее возлюбленного. В данном случае среднее арифметическое равно  $(2 + 9)/2 = 5,5$ . Маша может округлить это число или вверх, или вниз, по собственному усмотрению. Например, она выбирает цифру 5. Записав полученную цифру под второй цифрой своего номера, Маша переходит к получению третьей цифры тем же способом, т.е. находя полусумму третьей цифры своего номера и второй цифры нового номера. Получается  $(5 + 3)/2 = 4$ . В данном случае ответ однозначен. Таким образом, каждая  $i$ -я цифра определяется как среднее арифметическое  $i$ -й цифры Машиного номера и  $i - 1$ -й цифры номера ее возлюбленного, при необходимости возможно округление в любую сторону. Например, Маша может получить:

12345

95444

К сожалению, звонок по этому номеру разочаровал Машу: выяснилось, что абонент временно недоступен или находится вне зоны действия сети. Но Маша не отчаивается. Возможно, она произвела округления в неправильную сторону или неудачно выбрала первую цифру. Поэтому она продолжает получать все новые и новые номера описанным способом и звонить по ним. Вычислите количество номеров, по которым позвонит Маша. Маша звонит по всем возможным номерам, которые можно получить при помощи описанного гадания, кроме, возможно, своего собственного.

### Формат входного файла

В первой строке задана непустая последовательность из цифр от 0 до 9 — номер телефона Маши. Длина последовательности не превосходит 50.

### Формат выходного файла

Выведите одно число — количество номеров, по которым позвонит Маша.

### Примеры

| <code>stdin</code> | <code>stdout</code> |
|--------------------|---------------------|
| 12345              | 48                  |
| 09                 | 15                  |

## Задача I. Игрушки

Имя входного файла: `stdin`  
Имя выходного файла: `stdout`  
Ограничение по времени: 5 seconds  
Ограничение по памяти: 256 megabytes

Маленькая девочка Маша любит раскладывать свои игрушки в кучки на полу. А еще она очень не любит, когда кто-нибудь трогает ее игрушки. Как-то раз Маша аккуратно разложила все свои  $n$  игрушек в несколько кучек, а затем ее старший брат Саша пришел и собрал все кучки в одну. Маша увидела это, страшно расстроилась и принялась плакать. У Саши никак не получается успокоить Машу, а скоро придет мама и ему влетит за то, что сестренка плачет. Поэтому он решает восстановить разложение игрушек по кучкам. Однако он совершенно не помнит, как лежали игрушки. Маша, конечно, помнит это, но она еще не умеет говорить и может помочь Саше лишь тем, что радостно завопит, когда он выложит игрушки именно так, как они и лежали. Значит, Саше придется раскладывать игрушки всеми возможными способами до тех пор, пока Маша не узнает нужное разложение. Взаимное расположение кучек и игрушек в каждой кучке не важно, поэтому два способа разложить игрушки считаются различными, если найдутся две такие игрушки, которые при одном из способов лежат в одной и той же кучке, а при другом нет. Саша ищет наиболее быстрый путь перебора всех способов, потому что скоро придет мама. За одно действие Саша может взять игрушку из любой кучки и переложить ее в любую другую кучку (возможно, при этом появится новая кучка или одна из кучек исчезнет). Саша хочет найти такую последовательность действий, что в процессе все способы разложить игрушки на кучки будут перебраны ровно по одному разу. Помогите Саше. Изначально, как мы помним, все игрушки находятся в одной кучке.

### Формат входного файла

В первой строке записано целое число  $n$  ( $1 \leq n \leq 10$ ) — число игрушек.

### Формат выходного файла

В первой строке выведите количество различных способов разложить игрушки на кучки. Далее выведите все способы разложить игрушки на кучки в том порядке, в каком их должен перебирать Саша (т.е. каждый следующий способ должен получаться из предыдущего описанной в условии операцией). Каждый способ надо выводить следующим образом. В каждой кучке игрушки надо упорядочить по возрастанию номера. После этого надо упорядочить кучки по возрастанию номеров первых игрушек в них. Каждый способ выводите в отдельной строке. См. пример для уточнения формата выходных данных. Если решений несколько, выведите любое из них.

### Примеры

| <code>stdin</code> | <code>stdout</code>  |
|--------------------|--|
| 3                  | 5<br>{1, 2, 3}<br>{1, 2}, {3}<br>{1}, {2, 3}<br>{1}, {2}, {3}<br>{1, 3}, {2} |

## Задача J. Триминошки

|                         |                     |
|-------------------------|---------------------|
| Имя входного файла:     | <code>stdin</code>  |
| Имя выходного файла:    | <code>stdout</code> |
| Ограничение по времени: | 2 seconds           |
| Ограничение по памяти:  | 256 megabytes       |

Бывает много интересных задач о замощении доминошками. Например, известен такой замечательный факт. Возьмем стандартную шахматную доску ( $8 \times 8$ ) и вырежем из нее ровно две клетки. Оказывается, что получившуюся доску всегда можно замостить доминошками  $1 \times 2$ , если две вырезанные клетки разного цвета, и нельзя в противном случае.

Пете наскучили доминошки, поэтому он взял шахматную доску (не обязательно  $8 \times 8$ ), вырезал из нее некоторые клетки и пытается замостить ее триминошками. Триминошки — это прямоугольники  $1 \times 3$  (или  $3 \times 1$ , потому что триминошки можно беспрепятственно поворачивать), причем две крайние клетки триминошки обязательно белые, а клетка в середине — черная. Триминошки разрешается класть на шахматную доску таким образом, чтобы их клетки совпадали с невырезанными клетками доски, причем должно выполняться соответствие цветов: черные клетки можно совмещать только с черными, а белые — с белыми. Триминошки не должны выступать за пределы доски или накладываться друг на друга. Все невырезанные клетки доски должны быть покрыты триминошками.

Помогите Пете определить, можно ли замостить его доску триминошками описанным способом и выведите один из вариантов замощения.

### Формат входного файла

В первой строке заданы два целых числа  $n$  и  $m$  ( $1 \leq n, m \leq 1000$ ) — размеры доски. Следующие  $n$  строк содержат по  $m$  символов каждая и представляют собой описание доски. Если в некоторой позиции содержится «.», то клетка в этой позиции вырезана. Белой клетке соответствует символ «w», черной — «b». Гарантируется, что путем добавления вырезанных клеток можно получить корректную шахматную доску (т.е. с чередованием черных и белых клеток), хотя, возможно, и нестандартного размера.

### Формат выходного файла

Если хотя бы одно корректное замощение существует, выведите в первой строке «YES» (без кавычек), а затем — описание замощения. Описание должно содержать  $n$  строк, по  $m$  символов в каждой. Вырезанные клетки, как и во входных данных, обозначаются символом «.». Для обозначения триминошек можно использовать символы «a», «b», «c», «d», причем все три клетки каждой триминошки должны обозначаться одним символом. Если две триминошки граничат по стороне, то они должны обозначаться разными символами. Две триминошки, не граничащие по стороне, могут обозначаться одним и тем же символом (см. пример).

Если корректных замощений несколько, разрешается выводить любое. Если замостить доску триминошками невозможно или не существует корректного замощения, для описания которого было бы достаточно четырех символов «a», «b», «c», «d», выведите в первой строке «NO» (без кавычек).

## Примеры

| stdin   | stdout   |
|---|--|
| 6 10<br>.w.wbw.wbw<br>wbwbw.w.w.<br>bw.wbwbwbw<br>w.wbw.wbwb<br>. . .wbw.w.w<br>. .wbw.wbw. | YES<br>.a.aaa.ccc<br>baccc.c.a.<br>ba.dddcbab<br>b.aaa.cbab<br>. . .bbb.b.b<br>. .ccc.ddd. |
| 2 2<br>wb<br>bw   | NO   |
| 1 3<br>wbw  | YES<br>bbb   |
| 1 3<br>. . .  | YES<br>. . .   |