

## Задача А. Another Game

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Вася играет с друзьями в настольную игру Каркассон. В ней игроки по очереди тянут игровую карточку (тайл) и должны пристроить ее к уже имеющейся карте мира, которая представляет собой некоторое связное по стороне множество тайлов, находящихся на игровом столе.

Тайл представляет собой квадрат, на лицевой стороне которого содержится описание ландшафта. Квадрат состоит из клеток, представляющих один из следующих элементов ландшафта: дорогу (обозначается «X», ASCII код 88), город (обозначается «\*», ASCII код 42) или зеленый луг (обозначается «0», ASCII код 79). На обратной стороне тайла ничего не нарисовано. Тайл можно пристроить к имеющейся на столе карте только если будут выполнены следующие условия:

- тайл обязательно должен граничить по стороне хотя бы с одним тайлом на игровом столе;
- тайл должен граничить с имеющимися тайлами без противоречий ландшафта: дорога должна входить в дорогу, город должен продолжаться городом, зеленый луг должен находиться рядом с лугом;
- тайл должен граничить без смещений: с каждой стороны он должен граничить не более чем с одним тайлом; длина границы должна равняться длине тайла;
- тайл не должен накладываться на другие тайлы.

Пристраиваемый тайл можно поворачивать на угол кратный 90 градусам, но нельзя переворачивать. Игровой стол является бесконечным.

Следующий ход делает Вася. Он хочет узнать по имеющейся карте мира и описанию тайла, находящегося у него в руках, сколько различных ходов он может сделать. Два хода считаются различными, если после них получаются различные карты на столе. Если получившиеся карты совпадают при ненулевом смещении и/или повороте, то такие ходы считаются различными (карта «прибита» к игровому столу и двигать ее нельзя) — см. пример №3. Если ходы различаются только поворотом исходного тайла и дают одинаковые карты, то такие ходы считаются одинаковыми — см. пример №4.

### Формат входных данных

В первой строке задано единственное число  $N$  — количество строк в представлении тайла ( $3 \leq N \leq 10$ ). Далее следуют  $N$  строк по  $N$  символов в каждой — представление тайла, находящегося у Васи. Представление тайла содержит рамку ширины 1 и описание ландшафта лицевой стороны тайла (итого, ландшафт тайла представляет собой квадрат  $(N - 2) \times (N - 2)$  клеток). Четыре угла рамки обозначены символом «+» (ASCII код 43), верхняя и нижняя стороны обозначены символами «=» (ASCII код 61), правая и левая стороны — символами «|» (ASCII код 124).

В следующей строке находятся два числа —  $H$  и  $W$  — высота и ширина представления текущей карты мира соответственно, ( $N \leq W, H \leq 100$ ). Далее следуют  $H$  строк по  $W$  символов в каждой — описание карты мира. Между границами соседних тайлов, а также на сторонах тайлов, не имеющих соответствующих соседей, имеется рамка ширины 1, такая же, как у тайла Васи. Для удобства считывания данных пустые места на карте обозначены символами «.» (ASCII код 46). Поскольку игровой стол является бесконечным, добавляемый тайл может выходить за пределы текущего описания карты. Гарантируется, что непустые клетки карты образуют множество связанных по стороне тайлов, каждый из которых имеет такой же размер — содержит  $(N - 2) \times (N - 2)$  клеток. Гарантируется, что на карте уже присутствует хотя бы один тайл, и все тайлы (кроме первого добавленного) были добавлены на карту по очереди, в соответствии с правилами, описанными в условии. Гарантируется, что представление карты не содержит полностью пустых строк и столбцов.

### Формат выходных данных

Вывести единственное число — количество различных способов добавить тайл к игровому полю.

## Примеры

стандартный ввод	стандартный вывод
<pre>4 +==+  **   XO  +==+ 7 10 ...+==+... ... O* ... ... ** ... +==+==+==+  X* ** *O   *X X* *X  +==+==+==+</pre>	3
<pre>4 +==+  **   XO  +==+ 7 10 ...+==+... ... O* ... ... ** ... +==+==+==+  X* ** *X   *X X* *X  +==+==+==+</pre>	4
<pre>6 +====+  XX*X   OXXO   *XX*   XXOX  +====+ 6 11 +====+====+  XX*X XX*X   OXXO OXXO   *XX* *XX*   XXOX XXOX  +====+====+</pre>	2
<pre>4 +==+  OO   OO  +==+ 4 4 +==+  OO   OO  +==+</pre>	4

## Замечание

В первом примере существует 3 возможных различных хода.

При повороте исходного тайла на 180 градусов — 1 вариант:

```
.....+==+...
.....|0*|...
.....|**|...
+==+==+==+==+
|0X|X*|**|*0|
|**|*X|X*|*X|
+==+==+==+==+
```

При повороте на 90 градусов по часовой стрелке — 2 варианта:

```
...+==+...
...|X*|...
...|0*|...
...+==+...
...|0*|...
...|**|...
+==+==+==+
|X*|**|*0|
|*X|X*|*X|
+==+==+==+
...+==+...
...|0*|...
...|**|...
+==+==+==+
|X*|**|*0|
|*X|X*|*X|
+==+==+==+
...|X*|...
...|0*|...
...+==+...
```

Во втором примере кроме аналогичных ходов также допустим один ход при повороте исходного тайла на 90 градусов против часовой стрелки:

```
...+==+==+
...|0*|*0|
...|**|*X|
+==+==+==+
|X*|**|*X|
|*X|X*|*X|
+==+==+==+
```

В третьем примере исходная карта состоит из двух тайлов, равных тайлу Васи. Этот тайл можно добавить к карте слева или справа. Получившиеся карты состоят из трёх одинаковых тайлов, расположенных в ряд. Эти карты совпадают с **точностью до переноса**, но ходы считаются различными.

В четвертом примере исходный тайл можно повернуть четырьмя способами и присоединить к имеющемуся на карте тайлу с любой из четырёх сторон. Поскольку поворот исходного тайла не влияет на вид получившейся карты, то в данном примере допустимо всего 4 **различных** хода.

## Задача В. Bomb Has Been Planted

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Вася играет в популярный шутер CS:NO. Карта уровня представляет собой множество ключевых точек и прямых переходов между ними. Если две ключевые точки соединены переходом, то Вася может перейти из одной точки в другую ровно за 4 секунды. По переходам можно передвигаться в обе стороны. Также на карте существуют две специальные точки —  $A$  и  $B$ , в одной из которых заложена бомба, но Вася не знает, в какой именно. На данный момент Вася находится в точке  $C$ . Ему необходимо добраться до точки, в которой заложена бомба, и обезвредить её. За какое время Вася гарантированно сможет это сделать, если будет действовать оптимально? Учтите, что для обезвреживания бомбы Васе необходимо потратить дополнительно 10 секунд после того, как он доберется до нее.

### Формат входных данных

В первой строке задано два числа  $N$  и  $M$  — количество ключевых точек на карте уровня и количество переходов между ними соответственно ( $3 \leq N \leq 100$ ,  $2 \leq M \leq N \cdot (N - 1)/2$ ). Каждая из следующих  $M$  строк содержит два различных числа — номера точек, соединяемых очередным переходом (точки пронумерованы числами от 1 до  $N$ ). Гарантируется, что все переходы различны, и по ним можно добраться из любой точки карты в любую другую. В последней строке заданы три различных числа — номера точек  $A$ ,  $B$  и  $C$ .

### Формат выходных данных

Вывести единственное число — наименьшее время в секундах, за которое Вася гарантированно сможет обезвредить бомбу.

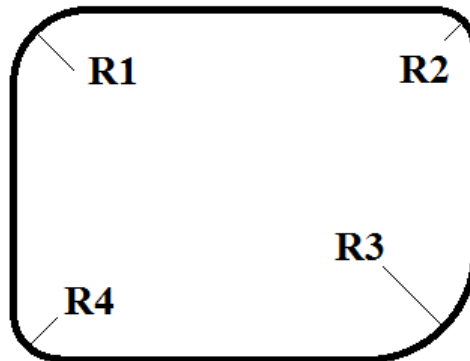
### Примеры

стандартный ввод	стандартный вывод
4 3 1 2 2 3 3 4 2 4 1	22
4 3 1 2 2 3 3 4 1 4 2	26
4 4 1 2 2 3 3 4 4 1 1 4 2	18

## Задача C. CSS Is Awesome

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Вася разрабатывает веб-страницу для отображения результатов ВКОШП. Главный элемент страницы — таблица результатов, которая будет отображаться в прямоугольнике  $W \times H$ . Вася хочет, чтобы его страница выглядела стильно и современно, поэтому он использует возможности CSS для создания дополнительных эффектов. В частности, он хочет сделать так, чтобы исходный прямоугольник имел скругленные углы. Вася задал радиусы скругления —  $R_1, R_2, R_3, R_4$ . Помогите Васе найти площадь получившейся фигуры.



### Формат входных данных

В первой строке заданы два натуральных числа  $W$  и  $H$  — ширина и высота исходного прямоугольника ( $1 \leq W, H \leq 1000$ ). Во второй строке заданы четыре целых неотрицательных числа — радиусы скругления  $R_1, R_2, R_3, R_4$  ( $0 \leq R_1, R_2, R_3, R_4 \leq \min(W/2, H/2)$ ).

### Формат выходных данных

Вывести единственное число — площадь получившейся фигуры. Ответ необходимо выдать с абсолютной или относительной погрешностью не более  $10^{-6}$ .

### Примеры

стандартный ввод	стандартный вывод
2 2 0 0 0 0	4.000000000
2 2 1 1 1 1	3.141592654

### Замечание

В первом примере все радиусы равны 0, значит, фигура представляет собой квадрат  $2 \times 2$ , площадь равна 4.

Во втором примере фигура вырождается в окружность радиуса 1, площадь равна  $\pi$ .

## Задача D. Decompressing

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Вася играет в The Elder Trolls III: Windomorr. Игра эта довольно старая, когда-то она была чрезвычайно популярной, но сейчас графика уже не радует глаз современного геймера. Вася решил реализовать свой собственный алгоритм улучшения текстур игры. Текстура представляет собой  $N$  рядов по  $M$  пикселей в каждом. Каждый пиксель характеризуется яркостью — целым числом от 0 до 9. Вася хочет растянуть текстуру по горизонтали ровно в  $K$  раз (то есть, текстура будет содержать  $N$  рядов по  $M \cdot K$  пикселей). При этом должны соблюдаться следующие свойства:

- каждый пиксель исходной текстуры заменяется на горизонтальный отрезок из  $K$  пикселей, при этом порядок получившихся отрезков соответствует порядку исходных пикселей;
- каждый пиксель новой текстуры должен также обладать яркостью от 0 до 9;
- средняя яркость пикселей каждого отрезка новой текстуры должна быть равна яркости соответствующего исходного пикселя;
- красота новой текстуры должна быть максимальной.

Красота текстуры равна сумме значений красоты всех пар соседних пикселей одного ряда (составляющие пару пиксели не обязательно принадлежат одному отрезку, то есть могут соответствовать разным пикселям исходной текстуры). Пара соседних пикселей одного ряда с яркостью  $P_1$  и  $P_2$  имеет красоту, равную  $100 - (P_1 - P_2)^2$ . Однако, существует  $T$  исключений, характеризующихся тройками чисел  $A, B, C$ . Красота пары пикселей с яркостями  $A$  и  $B$  (неважно, какой из пикселей пары имеет яркость  $A$ , а какой —  $B$ ), равна  $C$ , а не величине, вычисляемой по указанной выше формуле. Пары пикселей, соседние по вертикали, не влияют на красоту текстуры.

По заданной исходной текстуре помогите Васе найти соответствующую ей растянутую по горизонтали в  $K$  раз текстуру максимальной красоты. Если оптимальных решений несколько, то выведите любое.

### Формат входных данных

В первой строке заданы три целых числа  $N, M$  и  $K$  — количество рядов пикселей исходной текстуры, количество пикселей в ряду и коэффициент растяжения текстуры по горизонтали ( $1 \leq N, M \leq 50, 2 \leq K \leq 50$ ). Далее идут  $N$  строк по  $M$  символов в каждой — описание исходной текстуры. Каждый символ представляет собой цифру от 0 до 9. В следующей строке дано число  $T$  — количество исключений в правиле подсчета красоты ( $0 \leq T \leq 10$ ). Затем следуют  $T$  строк, содержащие по три числа  $A, B, C$  — описания исключений ( $0 \leq A, B \leq 9, 0 \leq C \leq 100$ ). Гарантируется, что исключения не противоречат друг другу, то есть каждая пара яркостей  $A, B$  встречается не более одного раза.

### Формат выходных данных

Вывести  $N$  строк по  $M \cdot K$  символов в каждой — описание растянутой текстуры максимальной красоты.

## Примеры

стандартный ввод	стандартный вывод
3 3 2 010 111 234 0	001100 111111 223344
3 3 2 010 111 234 1 2 4 100	001100 111111 222444
5 10 3 1111881111 111888111 1188118811 188888881 8811111188 1 1 8 99	111111111111888888111111111111 11111111118888888888881111111111 111111888888111111888888111111 11188888888888888888888888111 888888111111111111111111888888

## Задача E. Equation

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Вася любит красивые математические равенства. Сегодня он захотел написать  $N$  различных целых чисел так, чтобы их произведение было равно их сумме. Помогите Васе!

### Формат входных данных

В первой строке задано единственное число  $N$  — количество чисел, которые нужно написать ( $2 \leq N \leq 100$ ).

### Формат выходных данных

Вывести искомые  $N$  целых чисел. Числа должны быть различны и не превосходить  $10^6$  по абсолютному значению. Если решения не существует, нужно вывести единственное число 0.

### Примеры

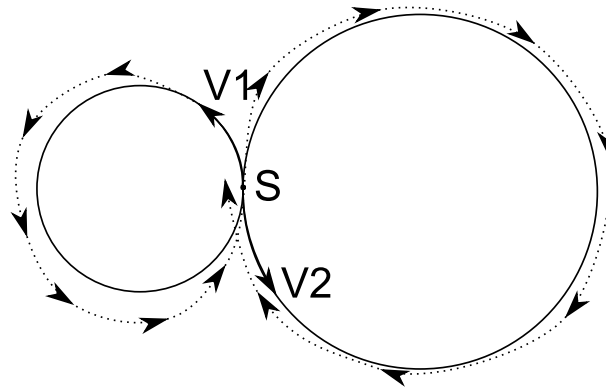
стандартный ввод	стандартный вывод
3	1 2 3



## Задача F. Formula 8

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Вася играет в гоночный симулятор Formula 8: BumpIn. Одна из трасс этой игры выглядит как восьмерка — состоит из двух кругов, имеющих одну общую точку. Два участника стартуют из центральной точки в противоположных направлениях и начинают описывать восьмерки (на рисунке показан путь первого участника). Если участники встречаются в любой точке, двигаясь навстречу друг другу, то они мирно разъезжаются. Если же они встречаются в центральной точке, двигаясь в «перпендикулярных» направлениях, то происходит столкновение.



Первый участник движется с постоянной скоростью  $V_1$  м/с, а второй — со скоростью  $V_2$  м/с. Левый круг имеет длину  $L_1$  метров, а правый —  $L_2$  метров. Оба участника стартуют в один и тот же момент и набирают скорость мгновенно. Васю интересует — столкнутся ли участники?

### Формат входных данных

На первой строке входного файла находятся два целых числа  $V_1$  и  $V_2$  ( $1 \leq V_1, V_2 \leq 10^9$ ) — скорости первого и второго участников соответственно. Во второй строке: два целых числа  $L_1$  и  $L_2$  ( $1 \leq L_1, L_2 \leq 10^9$ ) — длина левого и правого кругов соответственно.

### Формат выходных данных

Выведите YES, если участники когда-нибудь столкнутся или NO в противном случае.

### Примеры

стандартный ввод	стандартный вывод
1 3 5 10	YES
11 12 25 26	NO

### Замечание

В первом примере участники столкнутся спустя 5 секунд после начала гонки: первый участник преодолеет левый круг и прибедет в начальную точку двигаясь вверх-вправо, в то время как второй участник проедет восьмерку целиком и прибедет в начальную точку двигаясь вниз-вправо.

Во втором примере участники никогда не оказываются на перпендикулярных курсах в одной точке.

## Задача G. Grab Your Seat!

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

В тексте этой задачи Вася не играет ни в какую игру. На этот раз он осуществил свою детскую мечту и попал на самую крупную в мире конференцию, посвященную компьютерным играм, — ГеймзКонф! Это мероприятие настолько популярно среди геймеров, что в этом году для его проведения был построен огромный павильон, вмещающий  $M$  человек, и теперь для каждого посетителя найдется кресло. Все кресла в павильоне расположены в один ряд, для того чтобы все участники могли хорошо видеть сцену.

Конференция еще только начинается и лишь  $N$  кресел занято. Вася заметил, что если все зрители сидят на подряд идущих местах (например, с пятого по десятое или с восьмого по тысячу двадцать пятое), то вошедший в павильон участник легко выберет себе место: с левого или с правого конца подряд идущих кресел. В противном случае, человек долго мучается с выбором места, из-за чего образуется очередь, и зрители начинают нервничать. Нет ничего хуже нервного геймера: на YouTube по запросу «нервный геймер» вы найдете тысячи видео, доказывающих это.

Вася решил пересадить участников так, чтобы они расположились на подряд идущих местах. При этом Вася хочет побеспокоить как можно меньше присутствующих. Помогите ему в этом.

### Формат входных данных

В первой строке входных данных два целых числа  $N$  ( $0 \leq N \leq 10^5$ ) и  $M$  ( $N \leq M \leq 10^9$ ) — количество занятых кресел и вместимость павильона соответственно.

Во второй строке находятся  $N$  различных натуральных чисел  $a_i$  — номера занятых кресел ( $1 \leq a_i \leq M$ ).

### Формат выходных данных

Единственное число: минимальное количество участников, которых надо пересадить, чтобы все зрители сидели на подряд идущих креслах.

### Примеры

стандартный ввод	стандартный вывод
3 10 1 3 10	1

### Замечание

В примере достаточно пересадить участника, сидящего на десятом кресле, на второе кресло, тогда все люди будут находиться на подряд идущих местах: с первого по третье.

## Задача N. Heal

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Вася играет в MMORPG World of Foodcraft. В финальном сражении он со своим другом Петей должны убить Босса. У персонажа Васи на момент начала битвы  $H_v$  очков жизни, у Босса —  $H_b$  очков. Каждые  $T_v$  секунд Вася наносит Боссу урон в  $D_v$  очков. Каждые  $T_b$  секунд он получает урон от Босса в размере  $D_b$  очков. Персонаж Пети — клерик, поэтому он не может сражаться. Вместо этого каждые  $T_p$  секунд он заклинанием восстанавливает  $H_p$  очков жизни персонажу Васи. Если здоровье персонажа Васи падает до нуля или ниже, то персонаж погибает, и битва считается проигранной. Если Вася успевает довести здоровье Босса до нуля или ниже — он побеждает. Механика игры такова, что персонажи выполняют действия не чаще, чем раз в секунду, первым наносит удар Вася, затем — Босс, последним применяется лечебный эффект заклинания Пети. Персонаж Васи может иметь неограниченно много здоровья.

### Формат входных данных

На первой строке ввода находятся три числа, разделенные пробелами — характеристики персонажа Васи: начальное здоровье  $H_v$ , урон от одной атаки  $D_v$ , и минимальный интервал между атаками  $T_v$ . На второй строке — характеристики Босса в таком же формате: начальное здоровье  $H_b$ , урон от одной атаки  $D_b$ , интервал между атаками  $T_b$ . На третьей строке находятся два числа — характеристики персонажа Пети: количество очков жизни, которое восстанавливает его заклинание  $H_p$  и минимальный интервал между заклинаниями  $T_p$ .

Все числа во входных данных — натуральные, не превосходящие ста.

### Формат выходных данных

Выведите слово WIN, если Васе и Пете удастся одолеть Босса, или FAIL в противном случае.

### Примеры

стандартный ввод	стандартный вывод
100 5 2 10 25 1 1 1	WIN
10 5 2 11 5 1 1 2	FAIL

### Замечание

В первом примере события развиваются следующим образом:

- В первую секунду боя ( $t = 1$ ) Вася наносит Боссу урон 5, Босс наносит Васе урон 25, Петя восстанавливает Васе 1 очко жизни. У Васи 76 очков жизни, у Босса — 5.
- Во вторую секунду ( $t = 2$ ) Вася не может атаковать босса, поскольку он атаковал всего секунду назад, а может атаковать лишь каждые две секунды. Босс же наносит урон 25. Петя снова восстанавливает Васе одно очко жизни. У Васи 52 очка жизни, у Босса все еще 5.
- В третью секунду ( $t = 3$ ) Вася снова получает возможность атаковать, наносит Боссу урон 5, здоровье Босса падает до нуля и он умирает. Вася и Петя празднуют победу.

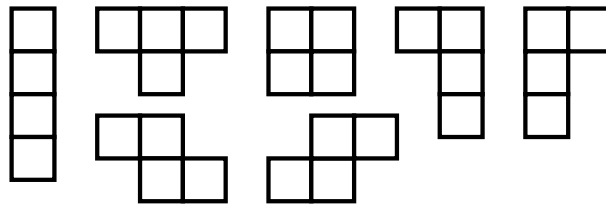
## Задача I. Is It Tetris?

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Вася поиграл в Тетрис, а теперь изучает статью про компьютерное зрение в Википедии. Его заинтересовало, можно ли научить компьютер отличать фигурки Тетриса от произвольных нагромождений клеток. Помочь в решении этого вопроса он просит вас.

Вася нарисовал в своей тетради несколько образцов, чтобы проверить ваше решение. Каждый образец — это поле  $5 \times 5$  клеток, некоторые из которых закрашены. Связные закрашенные клетки (то есть имеющие общую сторону) образуют фигурки. Некоторые из этих фигурок являются фигурками Тетриса.

Все возможные фигурки Тетриса представлены на рисунке ниже (эти фигурки также могут быть повернуты на 90, 180 и 270 градусов).



Ваша программа должна по описанию поля определять, сколько фигурок Тетриса на нем нарисовано.

### Формат входных данных

На вход программе подается описание поля — пять строчек, по пять символов в каждой. Символом «.» обозначены пустые клетки, а символом «#» — закрашенные.

### Формат выходных данных

Выведите одно число — количество фигурок Тетриса, которые находятся на поле.

### Примеры

стандартный ввод	стандартный вывод
<pre>...#. ..### ..... .##.. .##..</pre>	2
<pre>..... ..#.. .###. ..#.. .....</pre>	0

### Замечание

В первом примере на поле находится две фигурки — «Т»-образная (перевернутая) и квадрат. Во втором примере ни одной фигурки нет — потому что только часть фигурки совпадает с фигуркой Тетриса.

## Задача J. Jump!

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Вася играет в платформер Jump! В этой игре надо переправить лягушку с одного берега реки на другой, прыгая по кувшинкам, растущим на мелководье. Сложность заключается в том, что кувшинки то уходят под воду, то снова появляются на поверхности. Пока кувшинка под водой, на нее прыгать нельзя.

Схематически мелководье можно представить в виде прямоугольника размером  $W \times H$ , каждая клетка которого содержит ровно одну кувшинку. Для каждой кувшинки известны  $N$  пар подряд идущих временных интервалов: сколько секунд она скрыта под водой, и сколько затем находится на поверхности. По истечении заданных временных интервалов кувшинка тонет и больше никогда не всплывает. Например, если задана четверка чисел “3 5 2 4”, то это значит, что первые 3 секунды кувшинка находится под водой, следующие 5 — на поверхности, затем 2 секунды под водой, а следующие 4 — снова на поверхности. Через  $3 + 5 + 2 + 4 = 14$  секунд кувшинка тонет навсегда. Интервалы являются открытыми справа, то есть правая граница не входит в интервал. В приведенном выше примере лягушка может находиться на кувшинке в моменты времени  $t$ , удовлетворяющие условию  $3 \leq t < 8$  или  $10 \leq t < 14$ . На совершение одного прыжка лягушке требуется ровно 1 секунда. Таким образом, если лягушка начнет движение в момент времени  $t$ , то закончит в момент  $t + 1$ . Перемещаться разрешено только по клеткам со смежной стороной, то есть запрещено двигаться по диагонали.

Лягушка начинает свое путешествие в момент времени 0, находясь на левом берегу, и может прыгать только на находящиеся на поверхности кувшинки, которым соответствуют клетки крайнего левого столбца прямоугольника. Совершить прыжок на правый берег можно только с кувшинок, которым соответствует крайний правый столбец прямоугольника. Если кувшинка, на которой находится лягушка, уходит под воду, то игра проиграна.

Вася хочет узнать, за какое минимальное время он сможет переправить лягушку с левого берега реки на правый.

### Формат входных данных

В первой строке входных данных находятся три целых числа  $H$ ,  $W$  и  $N$  ( $1 \leq W \cdot H \leq 1000$ ,  $1 \leq N \leq 100$ ) — размеры мелководья и количество пар временных интервалов для каждой кувшинки.

В последующих  $W \cdot H$  строках заданы  $N$  пар положительных целых чисел, задающих временные интервалы, сумма которых для каждой кувшинки не превосходит 1000. Сначала идет описание кувшинок первого ряда (начиная с крайней левой), затем второго и т. д. Все величины заданы в секундах.

### Формат выходных данных

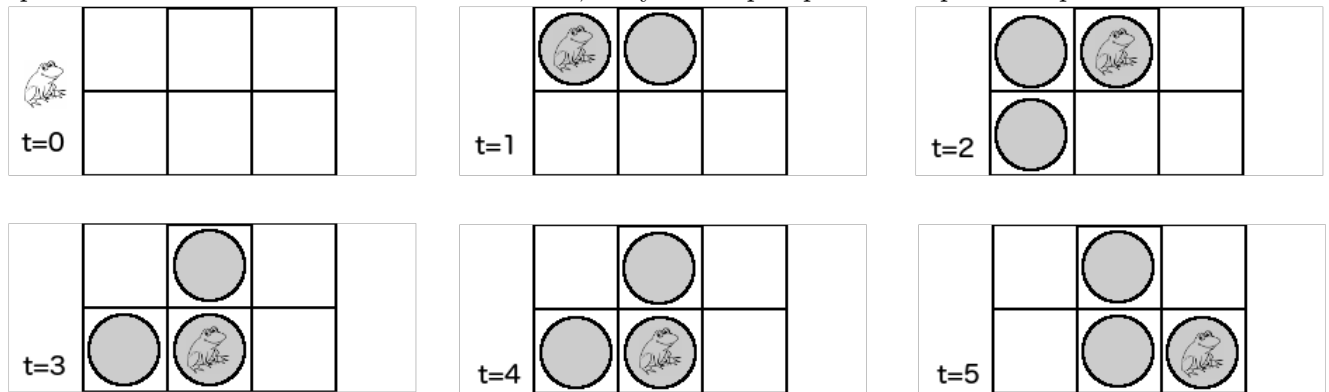
В единственную строку выходных данных выведите минимальное время в секундах, необходимое лягушке, чтобы перебраться с левого берега на правый, или  $-1$ , если перебраться на правый берег невозможно.

## Примеры

стандартный ввод	стандартный вывод
2 3 2 1 2 5 1 1 6 3 2 7 1 1 5 2 3 1 1 3 3 4 4 5 1 6 3	6
1 2 1 5 10 100 10	-1

## Замечание

В первом примере лягушке необходимо прыгнуть в момент времени  $t = 0$  на кувшинку в верхнем ряду. Это можно сделать, потому что в момент времени  $t = 1$ , когда завершится прыжок, эта кувшинка всплывет. Сразу после приземления лягушка делает второй прыжок, на следующую кувшинку в этом ряду. После этого совершает третий прыжок на вторую кувшинку второго ряда. Теперь лягушке необходимо одну секунду не двигаться, а в момент  $t = 4$  прыгнуть на кувшинку в правом нижнем углу, поскольку кувшинка всплывает именно в это время. Последним прыжком, который начнется в  $t = 5$  и закончится в  $t = 6$ , лягушка переберется на правый берег.



Во втором примере лягушка может находиться на первой кувшинке в моменты времени  $5 \leq t < 15$ , однако вторая кувшинка все это время находится под водой и, поэтому добраться на правый берег невозможно.

## Задача К. Key Number

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Вася играет в RPG Fallin, действие которой происходит в постапокалиптическом мире, пережившем ядерную войну. Главный Герой игры, выходец Поселения 31, выполняет крайне важную задачу по поиску воздушного чипа, с помощью которого можно очищать загрязненный воздух окружающей среды. На это у него есть не более 51 дня. Если за это время чип не будет найден и доставлен в Поселение 31, то все жители поселения погибнут.

В ходе поисков Герою нужно посетить множество живописных мест: деревню Санни-Хилс, Поселение 51, город Трэжестаун, торговый город Бах, поселение Райские Кущи и многие другие. Несмотря на жесткие ограничения по времени, Герой может выполнять дополнительные задания, которые ему дают встречаемые во время путешествия персонажи. В награду он получает бесценный опыт, оружие и боеприпасы, которые так нужны ему в этом опасном мире. Однако самым важным бонусом за выполнение дополнительных заданий является карма. Чем выше карма у Героя, тем дружелюбнее жители, тем дешевле можно купить у них товары и тем больше полезной информации они могут рассказать. Если же Герой ведет себя по-свински, то его карма ухудшается, как и отношение людей к нему.

За выполнение одного дополнительного задания Герой получил карту, на которой был отмечен тайник с одним из самых мощных оружий — винтовкой Тесла. Она способна одним выстрелом испепелить даже самого сильного противника! О таком оружии можно только мечтать!

После долгих и опасных странствий по пустошам Герой наконец нашел заветный тайник, в котором он обнаружил сейф с оружием. Для того чтобы открыть сейф, необходимо ввести правильную комбинацию из четырех цифр. Владелец карты дал Герою клочок бумаги с кодом, но по какой-то неведомой причине код записан задом наперед. Помогите Герою не запутаться, и ввести нужный код. В случае если будет набрана неправильная комбинация, раздастся взрыв!

### Формат входных данных

В первой строке находится целое число  $N$  ( $1000 \leq N \leq 9999$ ) — код, записанный на клочке бумаги.

### Формат выходных данных

В единственную строку выходных данных выведите правильную последовательность из четырех цифр без пробелов.

### Примеры

стандартный ввод	стандартный вывод
3247	7423
3165	5613

## Задача L. Looking for Next String

Ограничение по времени: 1 секунда  
Ограничение по памяти: 256 мегабайт

Вася играет в очередную игру-квест. Последние три дня он без особого успеха пытается разгадать очередную головоломку.

Вася знает, что ответом на головоломку является слово длиной  $N$  букв (в игре используются только маленькие латинские). Отчаявшись найти ответ честным способом, Вася пытается подобрать его, перебирая все такие слова. Чтобы не повторяться, он придумал перебирать слова в алфавитном порядке. Но Васе уже надоело вычислять каждое следующее слово руками, поэтому он просит вас помочь ему автоматизировать процесс — написать программу, которая по введенному слову выдает следующее в алфавитном порядке слово (такой же длины).

### Формат входных данных

В первой и единственной строке ввода находится последнее слово, которое попробовал Вася — строка из маленьких латинских букв длиной не более 1000 символов.

### Формат выходных данных

Выведите следующую строку, которую Вася должен попробовать в качестве разгадки. Если такой строки не существует — выведите строку `NO SOLUTION`.

### Примеры

стандартный ввод	стандартный вывод
abcd	abce
zzz	NO SOLUTION