A. Кроссворд

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 256Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Решение кроссвордов — популярное времяпрепровождение. Новый сервис Яндекс.Кроссоворды будет предлагать пользователям подобрать кроссворд, распечатать его и решить на бумаге. Однако? в таком случае проверить правильность решения кроссворда непросто.

Кроссворд расположен на клетчатом поле, состоящим из R*R* строк и C*C* столбцов. Каждая клетка покрашена в белый цвет (туда можно вписать букву) или в черный. Пользователь сервиса будет решать кроссворд и выписывать слова в белые клетки горизонтально (слева-направо) или вертикально (сверху-вниз) по одной букве в клетке. После решения кроссворда незаполненных белых клеток не остается. Словом называется горизонтальная или вертикальная последовательность клеток с буквами, ограниченная по краям черными клетками или границами поля.

Для проверки правильности решения кроссворда пользователь должен ввести в качестве ответа лексикографически минимальное слово (первое в алфавитном порядке), причем длина слова должна быть не меньше 2 символов.

Ваша задача состоит в том, чтобы определить ответ по разгаданному кроссворду.

Формат ввода

В первой строке вводится два целых числа R*R* и C*C* (1≤R,C≤201≤*R*,*C*≤20) — количество строк и столбцов соответственно.

В следующих R*R* строках вводится по одному слову, каждое слово состоит из прописных латинских букв и символов ”#”, обозначающих черную клетку.

Гарантируется, что существует хотя бы одно слово длиной в 2 или более символов.

Формат вывода

Выведите лексикографически минимальное слово, длина которого не меньше 2 символов.

Пример 1

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 3 3  yan  d#e  ##x | nex |

Пример 2

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 4 2  c#  on  te  st | cots |

Пример 3

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 2 5  inter  nship | ei |

Примечания

В первом примере в кроссворде всего 3 слова: yd, yan и nex. Минимальным является nex

Во втором примере по вертикали есть 2 слова: cont и net. По горизонтали 3 слова: on, te, st. Обратите внимание, что слово ”с” в первой горизонтали не подходит — оно состоит только из одной буквы.

В третьем примере все слова по горизонталям и вертикалям состоят из 2 или более букв, минимальным является ei.

B. Медианы подотрезков

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 256Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Рассмотрим перестановку чисел длины N*N*. Перестановкой чисел называется последовательность целых чисел от 1 до N*N*, в которой каждое число встречается ровно один раз.

Подотрезком последовательности A*A* называется последовательность чисел, получаемая из A*A* путём отбрасывания некоторого (возможно, нулевого) количества чисел из начала и конца последовательности.

Медианой последовательности нечетной длины называется такое число, которое будет стоять на центральном месте после сортировки последовательности. Например, в последовательности [7,1,20][7,1,20] медианой является число 77.

Определите количество подотрезков заданной перестановки, имеющих нечетную длину, медиана которых в точности равна заданному числу B*B*.

Формат ввода

В первой строке задаются два числа N*N* и B*B* (1≤N≤1051≤*N*≤105, 1≤B≤N1≤*B*≤*N*).

В следующей строке задаются N*N* чисел Ai*Ai*​ (1≤Ai≤N1≤*Ai*​≤*N*) — перестановка чисел.

Формат вывода

Выведите одно число — количество подотрезков нечетной длины, медиана которых равна числу B*B*.

Пример 1

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 5 2  5 4 3 2 1 | 2 |

Пример 2

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 6 3  3 6 5 4 2 1 | 1 |

Пример 3

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 7 4  5 7 2 4 3 1 6 | 4 |

Примечания

В первом примере подходят подотрезки [3,2,1][3,2,1] и [2][2]

Во втором примере подходит только подотрезок [3][3]

В третьем примере подходят подотрезки [5,7,2,4,3,1,6][5,7,2,4,3,1,6], [5,7,2,4,3][5,7,2,4,3], [7,2,4][7,2,4] и [4][4]

C. Альтернативная история

|  |  |
| --- | --- |
| Ограничение времени | 1 секунда |
| Ограничение памяти | 256Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Профессор математики Ерёменко разработал теорию, согласно которой реальных цивилизаций гораздо меньше, чем считают историки. В его теории есть основная цивилизация A*A* про которую известна последовательнось из N*N* исторических событий. Каждое событие обозначается числами от 11 до N*N*, каждое число встречается ровно один раз. В i*i*-й год в цивилизации происходило событие Ai*Ai*​.

Кроме цивилизации A*A* существовали также две ”ложные” цивилизации B*B* и C*C*, для них профессор Ерёменко также выписал случившиеся с ними исторические события, происходившие синхронно с событиями в цивилизации A*A*. В i*i*-й год в цивилизации B*B* происходило событие Bi*Bi*​, а в цивилизации C*C* — событие Ci*Ci*​. Эти события также обозначены числами от 11 до N*N* (однако для этих цивилизаций числа могут повторяться).

В теории профессора Ерёменко порядок событий не важен, главное чтобы у всёх трех цивилизаций A*A*, B*B* и C*C* множества событий совпадали. Помогите профессору Ерёменко вычеркнуть информацию за некоторые годы (т.е. удалить из последовательностей элементы Ai*Ai*​, Bi*Bi*​, Ci*Ci*​ для некоторых i*i*) так, чтобы множества событий стали совпадать. Чтобы сенсационность открытия профессора была выше, необходимо минимизировать количество вычеркнутых годов.

Формат ввода

В первой строке задается число N*N* (1≤N≤1000001≤*N*≤100000) — количество событий для каждой из цивилизаций.

В следующих трёх строках задаются описания исторических событий, случившиеся с цивилизациями A*A*, B*B* и C*C* соответственно. Все последовательности имеют длину N*N* и состоят из чисел от 1 до N*N*. В последовательности A*A* все числа различны.

Формат вывода

Выведите одно число — минимальное количество лет, информацию о которых необходимо вычеркнуть.

Пример 1

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 7  7 6 1 2 3 4 5  7 4 3 1 1 5 5  2 6 5 4 1 7 3 | 4 |

Пример 2

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 9  7 4 2 6 8 9 5 3 1  7 4 3 9 4 6 5 1 2  7 8 2 6 8 9 1 5 3 | 2 |

Примечания

В первом примере необходимо удалить информацию за 1, 2, 4 и 6 годы (при нумерации с единицы). Тогда в каждой цивилизации останется множество событий [1,3,5][1,3,5]

Во втором примере необходимо удалить информацию за 2 и 5 годы (при нумерации с единицы). Тогда в каждой цивилизации останется множество событий [1,2,3,5,6,7,9][1,2,3,5,6,7,9]

D. Правильная последовательность

|  |  |
| --- | --- |
| Ограничение времени | 2 секунды |
| Ограничение памяти | 256Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Правильной скобочной последовательностью назовём последовательность, состоящую из символов ”(”, ”)”, ”[”, ”]”, ”{” и ”}” и обладающую следующими свойствами:

1. Пустая последовательность является правильной скобочной последовательностью
2. Если S*S* — правильная скобочная последовательность, то (S)(*S*), [S][*S*] и {S}{*S*} также правильные скобочные последовательности
3. Если A*A* и B*B* правильные скобочные последовательности, то AB*AB* (к содержимому последовательности A*A* приписано содержимое последовательности B*B*) также правильная скобочная последовательность

Например, последовательности [](), [{()}]() — правильные, а [}, ([)] — нет.

В скобочной последовательности длины N*N* некоторые символы стерлись (обозначим из как ”?”). Определите количество способов поставить на место ”?” какую-либо скобку, чтобы последовательность стала правильной скобочной последовательностью. Так как это количество может быть очень большим, требуется вывести его по модулю 109+7109+7 (остаток от деления количества на 109+7109+7.

Формат ввода

В первой строке вводится чётное число N*N* (1≤N≤2001≤*N*≤200) — длина последовательности.

Во второй строке вводится последовательность, состоящая из символов ”(”, ”)”, ”[”, ”]”, ”{”, ”}” и ”?”.

Формат вывода

Выведите одно число — количество способов заменить знаки ”?” на скобки, чтобы последовательность стала правильной скобочной последовательностью, по модулю 109+7109+7.

Пример 1

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 4  []() | 1 |

Пример 2

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 10  ?{?[(?])?) | 3 |

Пример 3

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 4  ]??( | 0 |

Примечания

В первом примере нет знаков вопроса, но последовательность является правильной, поэтому существует единственный способ замены.

Во втором примере возможны следующие замены ({([()])})({([()])}), ({}[()])()({}[()])(), ({}[([])])({}[([])]).

В третьем примере нет ни одного способа получить правильную скобочную последовательность.

E. Неэффективный поиск

|  |  |
| --- | --- |
| Ограничение времени | 2 секунды |
| Ограничение памяти | 256Mb |
| Ввод | стандартный ввод или input.txt |
| Вывод | стандартный вывод или output.txt |

Наибольший общий префикс двух слов — это самое длинное слово, которое является началом как первого слова, так и второго. Например, слова ”хобот” и ”хорошо” имеют наибольший общий префикс ”хо”.

Наибольшие общие префиксы широко используются в поисковых технологиях, например, в подсказках при поиске. База данных содержит N*N* слов, которые нужно показывать в качестве поисковых подсказок. Стажёру было поручено проверить, входит ли введённое пользователем слово в базу данных. Стажёр реализовал поиск следующим образом: он идёт по словам из базы данных в том порядке, в котором они там записаны, и сравнивает очередное слово со словом из запроса буква за буквой, до тех пор, пока не найдутся отличающиеся буквы. Если слово из запроса совпало со словом из базы данных — поиск прекращается. Количество действий для этого алгоритма можно определить как количество слов из базы данных, с которыми проводилось сравнение, плюс сумма длин всех наибольших общих префиксов сравниваемых слов из словаря и слова из запроса.

Вам необходимо подсчитать количество действий этого алгоритма для Q*Q* различных запросов.

Формат ввода

В первой строке содержится число N*N* (1≤N≤300001≤*N*≤30000) — количество слов в базе данных.

В каждой из следующих N*N* строк записано по одному слову. Слова состоят из прописных английских букв, их длина не превосходит 30.

В следующей строке записано число Q*Q* (1≤Q≤300001≤*Q*≤30000) — количество запросов.

В каждой из следующих Q*Q* строк записано по одному запросу. Запросы состоят из прописных английских букв, их длина не превосходит 30.

Формат вывода

Выведите Q*Q* чисел по одному в строке — количество действий алгоритма для каждого из запросов.

Пример

| **Ввод**  Скопировать ввод | **Вывод**  Скопировать вывод |
| --- | --- |
| 3  ba  ab  abc  3  cd  ba  ab | 3  3  4 |

Примечания

В примере для запроса ”cd” будет выполнено три действия, т.к. запрос будет сверяться со всеми тремя словами из словаря, а суммарная длина наибольших общих префиксов равна нулю.

Для запроса ”ba” будет выполнено три действия, т.к. сравнение будет происходить с одним словом из словаря, а длина наибольшего общего префикса равна двум, что даёт в сумме три действия. Т.к. произошло совпадение со словом из словаря, то дальнейшая работа алгоритма прекращается.

Для запроса ”ab” будет выполнено 4 действия: сравнение с двумя словами из словаря и сумма длин наибольших общих префиксов также равна двум. Для первого слова длина наибольшего общего префикса составляет 0, а для второго — 2. После совпадения работа алгоритма прекращается.