

Задача А. ABCD-код

Имя входного файла: abcd.in
Имя выходного файла: abcd.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вася часто ходит в гости к Пете. Для того, чтобы попасть к Пете во двор, надо ввести код, состоящий из четырех цифр. Обычно друзья ходили вместе, но в этот раз Вася пришел один, а Петя ждет его у себя.

Вася не помнит код, но у него есть несколько вариантов. Кроме того, Васе почему-то запомнился факт, что квадрат числа, составленного из первых двух цифр кода, в сумме с квадратом числа, состоящего из последних двух цифр кода, имеет при делении на семь остаток один. То есть, если код представляет собой « $ABCD$ », где « A », « B », « C », « D » — некоторые цифры, тогда $AB^2 + CD^2$ имеет остаток 1 при делении на 7. Например, код 2843, является одним из возможных кодов, поскольку $28^2 + 43^2 = 2633 = 376 \cdot 7 + 1$, а 8243 — нет, поскольку $82^2 + 43^2 = 8573 = 1224 \cdot 7 + 5$.

У Васи есть несколько вариантов того, каким может быть код. Помогите ему определить, какие из вариантов могут быть кодом от входа в Петин двор.

Формат входных данных

В первой строке входного файла находится число t ($1 \leq t \leq 10\,000$) — число вариантов кода, которые помнит Вася. В следующих t строках содержится по четыре цифры — варианты кода.

Формат выходных данных

В выходной файл выведите t строк. В i -й строке выведите «YES», если i -й код может быть кодом для входа в Петин двор, иначе выведите «NO».

Примеры

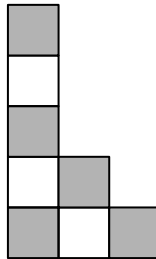
abcd.in	abcd.out
3	YES
2843	NO
8243	YES
0100	

Задача В. Шахматы

Имя входного файла: `chess.in`
Имя выходного файла: `chess.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Кроме школы и математического кружка, Вася ходит на шахматный кружок. Но играть в шахматы на обычной доске 8×8 ему кажется не очень интересным. Недавно он придумал свою версию шахмат, в которой игра происходит на доске, имеющей другую форму.

Васина доска состоит из n столбцов, i -й из которых содержит a_i клеток. Нижние клетки всех столбцов образуют один горизонтальный ряд, причем длины столбцов упорядочены слева направо по невозрастанию. На рисунке ниже приведен пример доски, в которой три столбца, содержащих 5, 2 и 1 клетку, соответственно.



Сегодня на шахматном кружке занятие было посвящено ладейным окончаниям, и Васю заинтересовал вопрос: как расставить минимальное число ладей на его доске так, чтобы каждую клетку поля была хотя бы одна ладья. Ладья бьет те клетки, которые расположены с ней на одной вертикали или одной горизонтали.

Помогите Васе расставить на его доске минимальное число ладей требуемым образом.

Формат входных данных

В первой строке входного файла задано целое число n — количество столбцов доски ($1 \leq n \leq 1000$). Следующая строка содержит n чисел a_1, a_2, \dots, a_n — количество клеток в столбцах ($1 \leq a_i \leq 1000, a_1 \geq a_2 \geq \dots \geq a_n$).

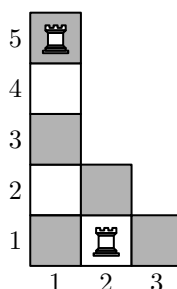
Формат выходных данных

В первой строке выведите число k — минимальное число ладей, которое можно расставить на доске так, чтобы каждую клетку доски была хотя бы одна ладья. Следующие k строк должны содержать описание позиций ладей, по одной на каждой строке. Позиция ладьи задается двумя числами: номером столбца, в котором стоит ладья, и номером клетки в столбце. Столбцы нумеруются, начиная с 1, слева направо, клетки в столбцах нумеруются снизу вверх, также начиная с 1.

Если подходящих расстановок несколько, можно вывести любую.

Примеры

<code>chess.in</code>	<code>chess.out</code>
3	2
5 2 1	1 5 2 1



Задача С. Завоевание

Имя входного файла: `conquest.in`
Имя выходного файла: `conquest.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Лорд Петир собирает армию для похода на соседнее королевство. Он хочет, чтобы в его армию вошли все воины каждого из n городов его королевства. Петир выяснил, что в i -м городе ищут работу a_i воинов, которых он может завербовать в свою армию.

Исходно в армии Лорда нет ни одного воина. Чтобы воин вошел в армию, Петир может заплатить этому воину. Для вербовки одного воина из i -го города, необходимо заплатить ему c_i золотых монет. При этом воины из больших городов ценят свою работу дороже, поэтому если для i -го и j -го города выполнено $a_i < a_j$, то $c_i \leq c_j$.

Однако есть еще один способ добиться того, чтобы воины присоединились к армии. Если в какой-то момент оказывается, что в армии Лорда Петира уже строго больше воинов, чем осталось в некотором городе, то все воины этого города бесплатно присоединяются к армии Лорда.

Помогите Лорду Петису выяснить, какое минимальное количество золотых монет он должен заплатить воинам, чтобы все воины из всех городов оказались в его армии.

Формат входных данных

В первой строке входного файла находится целое число n ($1 \leq n \leq 1000$) — количество городов, в которых Лорд Петир намерен набирать себе воинов. В следующих n строках входного файла находится по два целых числа a_i и c_i ($1 \leq a_i \leq 100$, $1 \leq c_i \leq 10\,000$) — количество воинов в i -м городе и число монет, которое необходимо заплатить одному воину в этом городе, чтобы он присоединился к армии. Для всех пар i и j выполнено условие, что если $a_i < a_j$, то $c_i \leq c_j$.

Формат выходных данных

В выходной файл выведите одно целое число — минимальное количество монет, которые Лорду Петису придется заплатить, чтобы все воины вошли в его армию.

Примеры

<code>conquest.in</code>	<code>conquest.out</code>
3	5
1 1	
2 2	
4 3	

В приведенном примере Лорду необходимо действовать следующим образом. Сначала он платит 2 монеты воину из второго города, и 3 монеты воину из третьего города, чтобы они присоединились к его армии.

Теперь в армии Лорда 2 воина, а в городах осталось 1, 1 и 3 воина, соответственно. Воины из первого и второго городов бесплатно присоединяются к армии Лорда Петира, в его армии становится 4 воина, после чего и оставшиеся 3 воина из третьего города бесплатно присоединяются к его армии.

Задача D. Дюны

Имя входного файла: `dunes.in`
Имя выходного файла: `dunes.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Географ Григорий Георгиевич исследует образование песчаных дюн. Он выбрал очень длинную дюну и разбил его на огромное число участков, которые пронумеровал от 1 до 10^9 .

Теория Григория Георгиевича гласит, что изначально высота песка относительно некоторой условной отметки на всех участках была равна нулю. После этого произошло n сильных порывов ветра, которые могли изменить ландшафт.

Порыв ветра номер i имел силу x_i и действовал на участки с l_i -го по r_i -й. В результате этого порыва высота участка номер l_i увеличилась на x_i , высота участка номер $l_i + 1$ уменьшилась на x_i , следующего — снова увеличилась на x_i , и так далее до участка номер r_i , включительно.

Зная всю информацию о всех n порывах ветра, Григорий Георгиевич хочет узнать установившуюся в итоге высоту некоторых интересующих его m участков. Помогите ему.

Формат входных данных

В первой строке входного файла содержатся два натуральных числа n и m ($1 \leq n, m \leq 1000$) — количество порывов ветра и количество участков, итоговая высота которых интересует Григория Георгиевича.

В каждой из следующих n строк содержится описание очередного порыва ветра — три целых числа l_i, r_i, x_i ($1 \leq l_i \leq r_i \leq 10^9$; $1 \leq x_i \leq 1000$).

В каждой из следующих m строк содержится целое число q_i ($1 \leq q_i \leq 10^9$) — номер участка, для которого требуется узнать его итоговую высоту. Номера участков приведены в возрастающем порядке.

Формат выходных данных

Для каждого из m запросов выведите одно целое число — высоту соответствующего участка.

Примеры

<code>dunes.in</code>	<code>dunes.out</code>
2 6	7
1 6 7	-7
3 7 2	9
1	-9
2	2
3	0
6	
7	
8	

Задача E. Игра

Имя входного файла:	game.in
Имя выходного файла:	game.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

На уроке физкультуры первоклассники Петя и Вася играют в увлекательную игру. Перед ребятами в ряд стоит n столбиков разной высоты. У мальчиков есть m колец, которые они по очереди кидают на столбики, причем если на столбике уже есть кольцо, то кидать кольцо на этот столбик нельзя. Петя кидает первым.

Ребята выяснили, что Петя может закинуть кольцо на столбик только, если высота этого столбика не меньше l_1 и не больше r_1 . На слишком высокий или слишком низкий столбик он закинуть кольцо не может. Зато, если столбик имеет подходящую высоту, бросок гарантированно заканчивается успехом. Аналогично, Вася может закинуть кольцо только на столбики с высотой не меньше l_2 и не больше r_2 и гарантированно закидывает кольцо на любой такой столбик.

Физрук Андрей Сергеевич обещал поставить пятерку тому из ребят, кто по итогам игры закинет больше колец на столбики. Помогите ребятам выяснить, кто из них выиграет при оптимальной игре.

Формат входных данных

В первой строке входного файла находятся два целых числа n и m — количество столбиков и колец, соответственно ($1 \leq m \leq n \leq 10^5$). Следующие две строки содержат числа l_1 , r_1 и l_2 , r_2 — минимальную и максимальную высоту столбиков, на которые могут кидать колечки Петя и Вася, соответственно ($1 \leq l_1 \leq r_1 \leq 10^9$, $1 \leq l_2 \leq r_2 \leq 10^9$). В последней строке содержится n чисел, описывающих высоту столбиков, высота каждого столбика является целым положительным числом и не превышает 10^9 .

Формат выходных данных

В выходной файл выведите «Petya», если выиграет Петя, «Vasya», если выиграет Вася, или «Draw», если при оптимальной игре оба мальчика закинут на столбики равное число колец.

Примеры

game.in	game.out
4 3 1 2 2 4 1 2 3 4	Petya
4 4 1 4 1 4 1 2 3 4	Draw
4 4 1 2 1 4 1 2 3 4	Vasya

В первом примере Петя сначала кидает кольцо на столбик высоты 2. Вася может в ответ закинуть кольцо на столбики высотой 3 или 4, но какой бы из них он не выбрал, Петя закинет третье кольцо на столбик высотой 1 и выиграет — он закинул 2 кольца, а Вася только одно.

Во втором примере каждый из игроков может закинуть кольцо на любой столбик, поэтому оба закинут по два кольца и игра закончится вничью.

В третьем примере Петя первым ходом закидывает кольцо на один из двух доступных ему столбиков, а Вася вторым ходом закидывает кольцо на второй из этих столбиков. Теперь у Пети нет столбиков, на который он может закинуть кольцо, он кидает третье кольцо, но не попадает. Вася же закидывает последнее кольцо на любой из столбиков высоты 3 или 4.

Задача F. НОД и НОК

Имя входного файла: `gcm.in`
Имя выходного файла: `gcm.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Сергея очень любит математические задачи. Недавно на математическом кружке ему рассказали, что такое НОД и НОК.

НОД двух натуральных чисел a и b — это их наибольший общий делитель, то есть такое максимальное число x , что a делится на x и b делится на x . Например, $\text{НОД}(24, 18) = 6$. А НОК целых чисел a и b — это их наименьшее общее кратное, то есть такое минимальное число x , что x делится на a и x делится на b . Например, $\text{НОК}(24, 18) = 72$.

Сергея сразу заметил, что может существовать несколько пар чисел с одинаковыми НОД и НОК. Теперь он заинтересовался вопросом: если заданы числа a и b , насколько близко друг к другу могут быть два числа, у которых такие же НОД и НОК.

Помогите ему по заданным двум числам a и b найти такие числа x и y , что $\text{НОД}(a, b) = \text{НОД}(x, y)$, $\text{НОК}(a, b) = \text{НОК}(x, y)$, а их разность $y - x$ минимальна.

Формат входных данных

В первой строке входного файла находятся два натуральных числа a и b ($1 \leq a, b \leq 10^9$).

Формат выходных данных

Выведите два натуральных числа x и y ($1 \leq x \leq y$), таких, что $\text{НОД}(a, b) = \text{НОД}(x, y)$, $\text{НОК}(a, b) = \text{НОК}(x, y)$, а их разность $y - x$ минимальна.

Примеры

<code>gcm.in</code>	<code>gcm.out</code>
3 4	3 4
1 12	3 4

Задача G. Мерлин

Имя входного файла: `merlin.in`
Имя выходного файла: `merlin.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Однажды, вернувшись в свою башню, Мерлин обнаружил, что Моргана наложила проклятие на все его сосуды с эликсиром мудрости.

Мерлин знает, как снять проклятие, но соответствующее заклинание требует, чтобы во всех сосудах, к которым оно применяется, было равное количество эликсира.

Чтобы добиться этого, Мерлин решил действовать следующим образом. Он выбирает несколько сосудов и переливает весь эликсир из выбранных сосудов в оставшиеся. Он может распределить переливаемый эликсир между оставшимися сосудами произвольным образом. После того, как весь эликсир из выбранных сосудов перелит, Мерлин разбивает опустошенные сосуды (с них проклятие уже не снять), выбрасывает осколки и применяет заклинание снятия проклятия к оставшимся сосудам.

Помогите волшебнику узнать, какое наименьшее количество сосудов ему придется разбить, чтобы снять проклятие Морганы.

Формат входных данных

В первой строке входного файла находится число n ($2 \leq n \leq 10^5$) — количество сосудов. Во второй строке содержатся n чисел a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) — количество литров эликсира мудрости в каждом сосуде.

Формат выходных данных

Выведите в выходной файл минимальное количество сосудов, которые Мерлину придется разбить.

Примеры

<code>merlin.in</code>	<code>merlin.out</code>
3 2 3 2	1
4 4 4 4 4	0
5 1 2 3 4 5	2

В первом примере можно, например, перелить 0.5 литра эликсира из первого сосуда во второй и 1.5 литра в третий, после чего разбить первый сосуд.

Во втором сосудах исходно содержат равное количество эликсира, можно ничего не переливать.

В третьем примере можно, например, перелить 1 литр эликсира из первого сосуда во второй, по 2 литра из пятого во второй и третий, 1 литр из пятого в четвертый, после чего разбить первый и пятый сосуды.

Задача Н. Регистрация на олимпиаду

Имя входного файла: `names.in`
Имя выходного файла: `names.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Петя и Вася проводят олимпиаду по программированию. На нее пришло так много участников, что для того, чтобы их всех зарегистрировать, Пете и Васе пришлось работать вдвоем.

Для того, чтобы зарегистрироваться, каждый участник называет свои имя, фамилию и отчество, а Петя и Вася заносят эту информацию в общую электронную таблицу. Так как участников много, а времени на организацию так мало, Петя и Вася не успели договориться о формате записи данных участника в таблицу и им пришлось импровизировать. Петя решил писать для каждого участника сначала его фамилию, затем имя, а затем — отчество, а Вася — сначала имя, затем отчество, а затем — фамилию.

По окончании регистрации стало понятно, что для подведения итогов олимпиады использовать данную таблицу невозможно: участнику будет неудобно себя искать. Было решено привести таблицу к следующему виду:

- для всех участников сначала написана фамилия, затем имя, а затем — отчество;
- участники в таблице упорядочены лексикографически по фамилии.

Петя и Вася заметили, что фамилии у всех участников различны, а вот каждое имя встречается хотя бы два раза. При этом никакое имя не является ни фамилией, ни отчеством никакого из участников, аналогично никакие фамилия и отчество не совпадают.

Пользуясь этой информацией, помогите им привести таблицу к желаемому виду.

Формат входных данных

В первой строке задано число n ($2 \leq n \leq 1000$) — общее число записей в электронной таблице. Далее, в n строках записано по три слова $s_{1,i}$, $s_{2,i}$, $s_{3,i}$. Каждое из слов содержит от 1 до 20 латинских букв, первая буква является заглавной, а все остальные — строчными. Каждая строка соответствует одной из записей, сделанных Петей или Васей. Слова разделены одним пробелом.

Формат выходных данных

Выведите n строк — электронную таблицу, в которой для каждого участника идет сначала фамилия, потом имя, потом отчество, причем все записи отсортированы лексикографически.

Лексикографический порядок соответствует порядку в словарях: слова сначала сравниваются по первой букве, затем по второй и т.д. Если очередная буква в одном из слов идет раньше в алфавите, то это слово лексикографически меньше другого. Если же расхождение так и не найдено, то есть одно из слов является префиксом другого, то считается, что слово, являющееся префиксом, лексикографически меньше.

Примеры

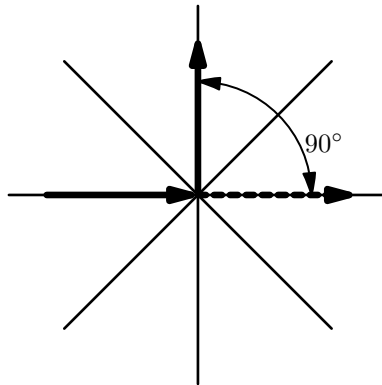
<code>names.in</code>	<code>names.out</code>
4	Ivanov Ivan Ivanovich
Ivanov Ivan Ivanovich	Pavlov Sergey Borisovich
Ivan Borisovich Petrov	Petrov Ivan Borisovich
Sergey Ivanovich Sidorov	Sidorov Sergey Ivanovich
Pavlov Sergey Borisovich	

Задача I. Безопасный путь

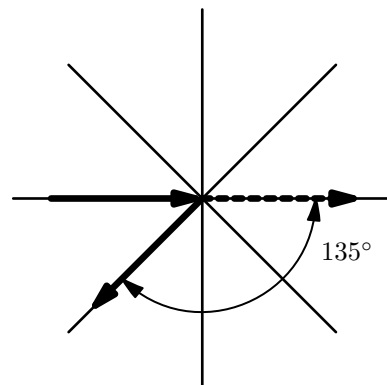
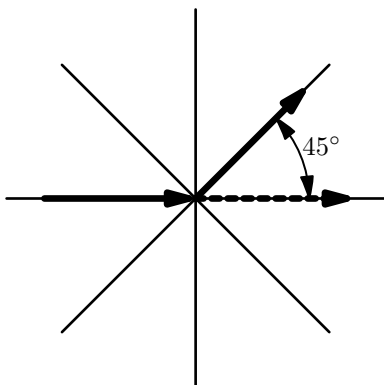
Имя входного файла:	saferoute.in
Имя выходного файла:	saferoute.out
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

Петя и Вася — хорошие друзья. Поэтому они часто ездят друг к другу в гости. Недавно Петя получил водительские права и собирается навестить своего друга. Для простоты будем считать, что все дороги в городе, в котором они живут, являются бесконечными прямыми. В месте пересечения двух или более дорог находятся перекрестки. Дома Пети и Васи расположены возле некоторых дорог города, но не на перекрестках.

Петя начинает путь на дороге возле своего дома. При этом он может выбрать любое из двух направлений. Когда Петя подъезжает к перекрестку, он может повернуть на любую другую проходящую через него дорогу или продолжить ехать по текущей. Поскольку Петя не очень опытный водитель, каждый поворот, который он совершает, заставляет его волноваться. Причем волнение Пети равно величине угла, на который он поворачивает, в градусах. Например, при повороте на прямой угол волнение Пети равно 90° .



При менее крутом повороте Петя волнуется меньше, а при более крутом — сильнее.



Будем считать, что волнение Пети в течении всего маршрута равно сумме величин в градусах углов, на которые ему придется повернуть в течении движения. Конечно, Петя хочет воспользоваться маршрутом, который заставит его волноваться как можно меньше.

Помогите Пете выяснить, чему равно минимальное суммарное волнение, которое он испытает, добравшись до дома Васи.

Формат входных данных

В первой строке входного файла находится целое число n ($1 \leq n \leq 50$) — количество дорог в городе. В следующих n строках находится описание дорог.

Каждая дорога описывается четверкой целых чисел x_1, y_1, x_2, y_2 , которые задают координатами двух различных точек (x_1, y_1) и (x_2, y_2) , через которые проходит дорога.

Гарантируется, что никакие две дороги не совпадают. В следующих двух строках заданы координаты домов Пети и Васи. Гарантируется, что каждый дом находится ровно на одной дороге, а также, что Петя и Вася живут в разных местах.

Координаты всех точек во входном файле являются целыми числами и не превосходят 100 по абсолютному значению.

Формат выходных данных

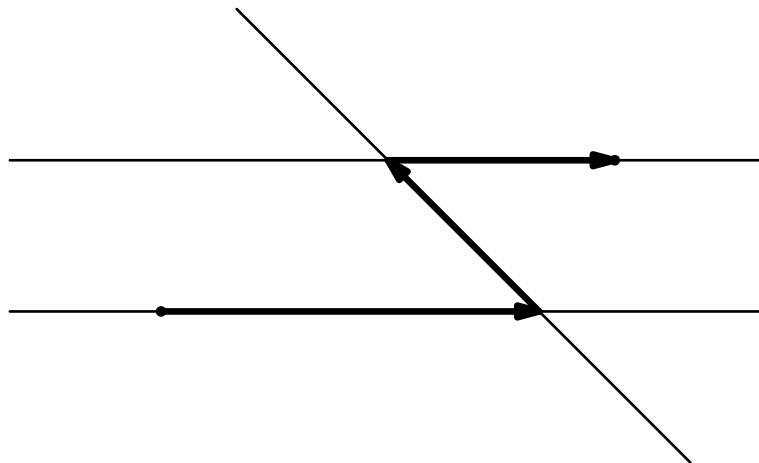
В выходной файл выведите единственное число — суммарный угол в градусах, на который придется повернуть Пете при оптимальном выборе маршрута. Ответ считается правильным, если его относительная или абсолютная погрешность не превосходит 10^{-9} .

Если Петя никак не сможет добраться до дома Васи, выведите число -1 .

Примеры

saferoute.in	saferoute.out
3 0 0 2 0 1 1 0 2 1 2 3 2 -3 0 3 2	270.0
1 0 0 2 0 0 0 2 0	0.0
5 0 0 1 0 0 0 1 1 0 0 0 1 0 0 -1 1 0 1 1 1 5 0 0 5	90.0

Следующий рисунок соответствует первому примеру. Петя совершает два поворота на 135 градусов, его суммарное волнение равно 270.



Задача J. Преобразование последовательности

Имя входного файла: `seq.in`
Имя выходного файла: `seq.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Учительница математики очень не любит Колю и всегда заставляет его отвечать у доски самые сложные задачи.

Вот и сегодня она написала на доске последовательность из n целых неотрицательных чисел a_1, a_2, \dots, a_n и вызвала Колю к доске. За одно действие учительница разрешает Коле стереть любое число и на его место записать число на единицу больше. Учительница требует от Коли за минимальное число действий добиться того, чтобы где-нибудь в этой последовательности встречались подряд в возрастающем порядке числа от 1 до h .

Помогите Коле понять, за какое минимальное число действий ему удастся добиться того, что для некоторого i будет выполнено $a_i = 1, a_{i+1} = 2, \dots, a_{i+h-1} = h$, или выясните, что это невозможно и учительница опять безнаказанно издевается над бедным Колей.

Формат входных данных

Первая строка входного файла содержит два натуральных числа: n и h ($1 \leq h \leq n \leq 200\,000$). Вторая строка содержит n чисел a_i — исходные значения элементов выписанной последовательности ($0 \leq a_i \leq n$).

Формат выходных данных

В единственной строке выходного файла выведите минимальное количество действий, за которое Коля сможет выполнить задание, или -1 в случае, если выполнить его невозможно.

Примеры

<code>seq.in</code>	<code>seq.out</code>
4 3 1 1 0 2	3
3 2 1 3 2	-1

В первом примере Коле надо дважды увеличить на 1 третье число и один раз — четвертое. Тогда последовательность примет вид 1, 1, 2, 3, для $i = 2$ выполнено искомое условие.

Во втором примере получить в последовательности подряд 1 и 2 невозможно.

Задача К. Крестики-нолики

Имя входного файла: `tictactoe.in`
Имя выходного файла: `tictactoe.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Пете и Васе стало очень скучно на уроке биологии, и они решили поиграть в любимую всеми школьниками игру в крестики-нолики до пяти в ряд на бесконечном поле.

Рассмотрим кратко правила игры. Игра происходит на бесконечном клетчатом поле, два игрока делают ходы по очереди, первый игрок ходит крестиками, а второй — ноликами. В свой ход игрок выбирает свободную клетку поля и ставит туда свой символ. Если после хода очередного игрока на поле есть пять его символов подряд по вертикали, горизонтали или диагонали, то сделавший такой ход игрок объявляется победителем и игра заканчивается.

Петя и Вася уже довольно долго играют в игру. Сейчас должен ходить Петя, который играет крестиками. Петя надеется побыстрее завершить игру и хочет выиграть не более чем за два, а лучше за один ход. Петя называет ход оптимальным, если для этого хода выполнено одно из двух:

- этот ход приводит к немедленной победе Пети;
- не существует хода, который приводит к немедленной победе Пети, но если Петя сделает этот ход, то Вася не выиграет следующим ходом и, вне зависимости от ответного хода Васи, у Пети будет следующий ход, который приведет к его немедленной победе.

Помогите Пете найти количество оптимальных ходов.

Формат входных данных

В первой входного файла находятся два натуральных числа n, m ($1 \leq n, m \leq 200$) — размеры прямоугольника, содержащего все уже поставленные на поле крестики и нолики.

Следующие n строк содержат по m символов, каждый из которых равен одному из следующих: «.» (точка), «X» (заглавная латинская буква «икс») или «0» (ноль). При этом «.» обозначает пустую клетку, «X» обозначает крестик, а «0» обозначает нолик. Гарантируется, что на поле находится равное число крестиков и ноликов, и ни один игрок еще не одержал победу.

Формат выходных данных

Выведите одно число — количество оптимальных ходов Пети.

Примеры

<code>tictactoe.in</code>	<code>tictactoe.out</code>
5 3 ... 000 XXX	2
4 4 ..0. .XX0 .0X.	0
5 6XXX.. .0000. ..X...	0