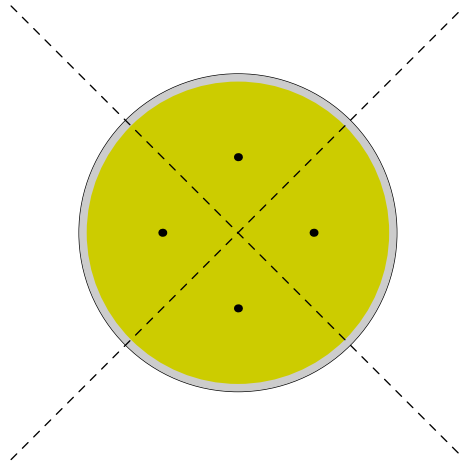


Задача А. Тортик

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

Винни-Пух, Кролик и Пятачок пришли в гости к Кристоферу Робину на его день рождения. Главный подарок — огромный круглый медовый торт с четырьмя свечками. Однако, чтобы съесть торт, его нужно сначала разрезать на четыре части. Зная, как сильно Винни-Пух любит мед, можно было ожидать, что он сразу же заберет себе самый большой кусок торта и наверняка застрянет в дверях, как это случилось, когда он и Пятачок ходили в гости к Кролику.



Чтобы избежать этого Кристофер Робин решил разрезать торт на четыре одинаковые части, а чтобы подчеркнуть торжественность, ведь это не обычный торт, а праздничный, и не усложнять разрезание, торт нужно разрезать двумя перпендикулярными разрезами, проходящими через его центр, причем на каждом из получившихся кусков должно быть ровно по одной свечке. При этом, чтобы никто не получил нецелую свечку и не обиделся, Кристофер Робин хочет, чтобы разрезы не проходили ни через одну свечку.

Формат входного файла

Во входном файле содержится описание торта. Центр торта принят за начало координат. Первая строка содержит одно целое число R ($1 \leq R \leq 10^9$) — радиус торта. В следующих четырех строках содержатся координаты свечек x_i, y_i . Гарантируется, что все свечки находятся строго в тортe.

Формат выходного файла

В выходной файл выведите «YES», если Кристофер Робин может разрезать торт так, как он хочет, и «NO» — если не может.

Примеры

| <code>cake.in</code> | <code>cake.out</code> |
|----------------------------------|-----------------------|
| 10 0 1 1 0 0 -1 -1 0 | YES |

Задача В. Шахматы

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

Вчера мудрая Сова в своем великолепном замке «Каштаны» нашла шахматы. Но выслушать и запомнить правила игры полностью смог только Кролик. В связи с этим, Сова решила упростить правила. В результате нововведений на доске остались только 2 фигуры, которые, для простоты запоминания, было решено назвать в честь жителей Волшебного Леса. Одна из фигур — это конечно же ужасный Слонопотам, в честь которого, кстати, назван обычный шахматный слон. А вторую фигуру единогласно было решено назвать в честь Тигры (кто же еще сможет схватить Ужо... пото... Слонопотама?). Тигра недоволен ходами существующих фигур и хочет ходить (а точнее прыгать), по-своему, ведь больше всего на свете Тигра любит прыгать, прыгать и прыгать. А точнее он хочет, чтобы его фигура могла прыгать либо на две клетки по вертикали или горизонтали, либо на одну клетку по диагонали. Также Сова решила, что будет справедливо, если фигура Слонопотама не будет ходить, поскольку настоящий Слонопотам обычно прячется в засаде и не двигается, чтобы его не заметили.

Помогите жителям Волшебного Леса понять, сумеет ли Тигра схватить, то есть попасть в одну клетку со Слонопотамом в ходе шахматной партии.

Формат входного файла

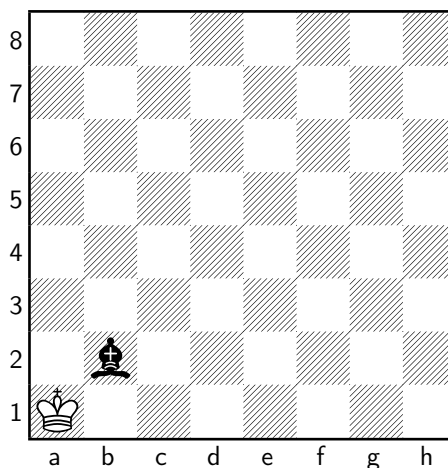
Во входном файле содержатся координаты двух клеток, с местонахождением Тигры и Слонопотама, каждая в отдельной строке. Координаты клеток задаются в виде пары буквы (от *a* до *h*) и цифры (от 1 до 8) без пробелов.

Формат выходного файла

В выходной файл выведите «YES», если фигура Тигры может схватить фигуру Слонопотама, и «NO» — если не может.

Примеры

| <code>chess.in</code> | <code>chess.out</code> |
|-----------------------|------------------------|
| a1 b2 | YES |



Задача С. Враг моего врага — мой друг!

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

Винни-Пух зарегистрировался в новой социальной сети, которая называется ВЛесу. В этой социальной сети у каждого пользователя, кроме списка его друзей, был также список его врагов. В этот список можно было добавить любого пользователя, но при этом удовлетворялись некоторые условия:

1. Если пользователь v является врагом пользователя u , то u **не обязательно** является врагом v
2. Пользователь не может быть врагом самого себя

Винни-Пуху очень понравилась эта социальная сеть. Он целыми днями сидел и записывал, кто же становился чьим врагом, так как хотел знать все, что происходит в их лесу. Он считал, что никто не пойдет в гости к своему врагу. Также, по его мнению, враг врага является другом, а любая уважающая своих друзей персона должна пойти в гости к своему другу. Винни-Пуху очень интересно узнать — сколько же у пользователя под номером v друзей. Пользователь u является другом пользователя v по версии Винни-Пуха, если выполняются некоторые условия:

1. u является врагом некоторого врага v
2. u не является врагом v

Заметим также, что никакой пользователь сам не является своим другом.

Формат входного файла

В первой строке входного файла задано числа n и m ($1 \leq n, m \leq 2000$) — количество пользователей, зарегистрированных в социальной сети и количество запросов соответственно.

В следующих m строках заданы запросы двух видов:

1. $+ v u$ — пользователь v начал считать пользователя u своим врагом
2. $? v$ — узнать количество друзей пользователя v по версии Винни-Пуха

Гарантируется, что входные данные корректны — пользователь не начнет считать себя своим врагом и никакой пользователь не станет врагом другого более одного раза.

Формат выходного файла

Для каждого запроса $? v$ выведите одно целое число — ответ на него в отдельной строке.

Примеры

| <code>enemy.in</code> | <code>enemy.out</code> |
|-----------------------|------------------------|
| 5 5 | 1 |
| + 1 2 | |
| + 2 4 | |
| + 2 5 | |
| + 1 5 | |
| ? 1 | |

Задача D. Нумерация дробей

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

Кролик очень сильно любит дроби. Особенно складывать их, умножать и сокращать. Именно поэтому ему недавно был подарен большой набор дробей.

Вчера Кролик прочитал статью в очень авторитетном математическом издании, в которой рассказывался способ нумерации дробей. Кролик очень любил порядок, и поэтому после прочтения этой статьи он сразу понял, что нужно упорядочить подаренные ему дроби именно так.

Чтобы упорядочить дроби необходимо их положить в таблицу, как показано на рисунке. В строку с номером i этой матрицы по порядку записаны дроби с числителем i , а в столбец с номером j дроби с знаменателем j .

$$\begin{pmatrix} \frac{1}{1} & & \frac{1}{2} & \rightarrow & \frac{1}{3} & & \dots \\ \downarrow & \nearrow & & \swarrow & & \nearrow & \\ \frac{2}{1} & & \frac{2}{2} & & \frac{2}{3} & & \dots \\ & \swarrow & & \nearrow & & \swarrow & \\ \frac{3}{1} & & \frac{3}{2} & & \frac{3}{3} & & \dots \\ \downarrow & \nearrow & & \swarrow & & \nearrow & \\ \vdots & & \vdots & & \vdots & & \ddots \end{pmatrix}$$

Дальше необходимо выписать все дроби в том порядке, как показано на рисунке стрелками. Получится такая последовательность: $\frac{1}{1}, \frac{2}{1}, \frac{1}{2}, \frac{1}{3}, \frac{2}{2}, \frac{3}{1}, \dots$

Кролик проделал все эти операции и получил последовательность дробей, как и было описано в прочитанной им статье.

Сегодня к Кролику в гости пришел Винни-Пух. Ему тоже очень понравились дроби, особенно его удивила последовательность, в которой они расположены. Он захотел узнать, какая дробь стоит на позиции n в этом порядке.

Вам требуется по числу n найти числитель и знаменатель n -ой дроби.

Формат входного файла

Во входном файле дано число n ($1 \leq n \leq 10^{18}$) — порядковый номер дроби в последовательности Кролика.

Формат выходного файла

В выходной файл требуется вывести через символ «/» два числа: числитель и знаменатель соответствующей дроби.

Примеры

| <code>fracnum.in</code> | <code>fracnum.out</code> |
|-------------------------|--------------------------|
| 1 | 1/1 |
| 6 | 3/1 |
| 2 | 2/1 |

Задача Е. Ямы

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

У Винни-Пуха большая радость — он купил автомобиль. И теперь, чтобы опробовать покупку, он собрался съездить на нем за медом.

От дома Винни-Пуха до дерева с правильными пчелами, дающими правильный мед, ведет дорога длиной n километров. Казалось бы, скорость передвижения на автомобиле там, где живет Винни, не регламентирована, и ехать можно сколь угодно быстро. Однако, если бы все было так просто, Винни бы не стал обращаться к Вам.

В некоторых местах на дороге расположены ямы. И, если Винни-Пух хочет сохранить свое новое авто в целости и сохранности (а он, конечно же, хочет), проезжать километр, на котором расположена яма, нужно со скоростью, не превышающей k километров в час. Конструкция автомобиля такова, что после каждого проеханного километра он может мгновенно ускориться на один километр в час, замедлиться на эту же скорость или оставить скорость своего движения без изменений. При этом после изменения он будет ровно километр ехать с новой скоростью, после чего снова сможет ее изменить.

Еще одна особенность автомобиля заключается в следующем: он не может мгновенно разогнаться до нужной скорости, зато может мгновенно остановиться (выбросив тормозной парашют). Соответственно, первый километр пути он всегда проедет со скоростью 1 километр в час, а скорость на последнем километре может быть ограничена только наличием на этом километре ям.

Винни любит быструю езду, а еще больше он любит мед. Соответственно, добраться от своего дома до дерева он хочет как можно быстрее. Вас же он попросил определить, за сколько он сможет добраться до дерева и при этом сохранить целым автомобиль.

Формат входного файла

В первой строке входного файла даны два целых числа n и k ($1 \leq n, k \leq 1000$) — длина дороги и максимальная скорость на участке с ямой, соответственно. В следующей строке дано n чисел, каждое из которых равно 0 или 1 — описание дороги. 0 означает, что соответствующий километр дороги ровный и его можно проезжать с любой скоростью, 1 же, напротив, означает, что километр содержит яму и его можно проехать со скоростью, не превышающей k километров в час.

Формат выходного файла

Выведите в выходной файл одно вещественное число — минимальное количество часов, за которое Винни-Пух доедет до дерева. Ответ должен отличаться от правильного не больше, чем на 10^{-6} .

Примеры

| <code>holes.in</code> | <code>holes.out</code> |
|------------------------|------------------------|
| 7 1 0 0 1 0 0 0 1 | 4.8333333 |
| 8 2 0 0 0 0 0 0 1 0 | 3.5 |

Задача F. Кто длиннее?

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

Как-то раз Тигра весело прыгал по лесу и придумывал новую игру. Вообще, как вам известно, Тигра очень любит различные игры, и чаще всего подвижные. Но в тот день над лесом постепенно сгустились тучи, и потому все мысли Тигры были о тихой и спокойной игре.

И вот тут Тигра вспомнил о очень веселой и занимательной игре, в которую они с Кроликом играли в далеком детстве. Правила ее хоть и не были просты, но игра занимала большое количество времени и приносила массу удовольствий.

Через пару минут Тигра встретил ослика Иа-Иа и предложил ему сыграть. Ослик согласился и тогда Тигра поведал ему правила игры: игроки по очереди приписывают к строчке одну из известных им букв английского алфавита, но при этом в строке не должно образоваться двух одинаковых комбинаций из двух подряд идущих символов. Тот, кто не может сделать ход — проигрывает.

Иа-Иа очень заинтересовался этой игрой и захотел выиграть у Тигры. Для начала он хочет придумать наидлиннейшую строку, которая может быть получена в процессе этой игры. Помогите ему в этом!

Формат входного файла

Единственная строка входного файла содержит число k ($1 \leq k \leq 26$) — количество первых букв английского алфавита, известное героям.

Формат выходного файла

В единственной строке файла должна быть строка, состоящая из k первых строчных букв английского алфавита — ответ на задачу.

Примеры

| | <code>string.in</code> | <code>string.out</code> |
|---|------------------------|-------------------------|
| 1 | | aa |

Задача G. Подпоследовательность Фибоначчи

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

Сегодня в школе Кристофер изучал последовательности и перестановки. Ему очень понравилась последовательность Фибоначчи. Последовательность чисел a_1, a_2, \dots является фибоначчиевой, если для любого $i > 2$ верно, что $a_i = a_{i-1} + a_{i-2}$.

Вечером Кристофер пришёл в гости к Кролику и увидел у него на столе набор карточек с числами. Кристофера сразу заинтересовал вопрос — можно ли составить из этих чисел фибоначчиевую последовательность.

Формат входного файла

В первой строке входного файла дано натуральное число n — количество элементов в последовательности ($1 \leq n \leq 100$). Во второй строке входного файла дано n натуральных чисел, меньших 10^9 .

Формат выходного файла

Вывести «YES» без кавычек, если из чисел можно составить фибоначчиеву последовательность, а иначе — «NO».

Примеры

| <code>subfib.in</code> | <code>subfib.out</code> |
|------------------------|-------------------------|
| 3 5 8 3 | YES |
| 3 5 6 7 | NO |

Задача Н. Часы

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

На очередной день рождения ослику Иа-Иа подарили наручные стрелочные часы. Теперь у него появилось новое развлечение — смотреть на бег стрелок. На то, как минутная догоняет часовую, обходит и тут же продолжает бежать за ней.

Вот и в этот раз Кенга застала ослика за этим занятием. Она присоединилась к наблюдением и через некоторое время ей стало интересно, сколько уже моментов, когда минутная стрелка обгоняет часовую, видел Иа-Иа. Для этого она спросила у ослика во сколько он начал смотреть на часы, записала это и текущее время и побежала к Сове с этим вопросом. Но Сова оказалось очень занята и поэтому попросила вас помочь.

Как известно, за один день часовая стрелка делает два оборота, а минутная целых 24.

Формат входного файла

Первая строка входного файла содержит описание момента времени, в который Иа-Иа начал смотреть на часы в формате НН:ММ — две первых цифры обозначают часы, потом через двоеточие две цифры минут.

Вторая строка входного файла содержит описание конечного момента времени в таком же формате.

Известно, что Иа-Иа наблюдал за часами менее суток, но вполне мог это делать и в полночь.

Формат выходного файла

В выходной файл выведите единственное целое число — количество моментов, когда минутная стрелка обгоняет часовую в процессе наблюдения.

Если такое событие происходило когда Иа-Иа начал или закончил смотреть на часы, то оно также считается.

Примеры

| <code>watches.in</code> | <code>watches.out</code> |
|-------------------------|--------------------------|
| 23:50 00:20 | 1 |
| 12:00 13:20 | 2 |