

Задача A. Beginning

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: `64 Mebibytes`
Ограничение по памяти: `2 seconds`

Во время открытия олимпиады было рассказано про несколько забавных игр с числами. Например, если IOI10 прочитать как двоичное число, получится 22 — номер олимпиады.

Вдохновлённый этим примером, один из участников предложил для открытия следующей, 23-й, олимпиады такой пример: пусть $Q(x)$ — функция, сопоставляющая целому числу сумму квадратов его цифр. Тогда $Q(23) = 2 * 2 + 3 * 3 = 13$, $Q(13) = 1 * 1 + 3 * 3 = 10$, $Q(10) = 1 * 1 + 0 * 0 = 1$ — то есть последовательно применяя $Q(x)$ к числу 23, в конечном итоге получим 1. При этом далеко не для всякого числа это справедливо — например, для 3 последовательное применение $Q(x)$ даёт последовательность 3, 9, 81, 65, 61, 37, 58, 89, 145, 42, 20, 4, 16, 37, 58... — то есть получается период (37, 58, 89, 145, 42, 20, 4, 16).

По заданному числу выясните, можно ли в результате последовательного применения функции $Q(x)$ к этому числу получить 1.

Формат входного файла

Во входном файле задано целое положительное число, не превосходящее 2×10^9 .

Формат выходного файла

В выходной файл выведите заданное во входном файле число, затем двоеточие, и затем 1, если в результате последовательного применения к этому числу функции $Q(x)$ можно получить единицу, и 'LOOP' в противном случае.

Пример

standard input	standard output
23	23:1
2	2:LOOP

Задача В. Canadian contestants

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: `64 Mebibytes`
Ограничение по памяти: `2 seconds`

Как известно, страна-организатор выставляет на IOI 8 участников. При жеребьёвке участники канадской делегации получили номера от 1 до 8. После церемонии открытия канадские участники зашли в лифт... Один из них заметил, что участники в лифте стоят в клетках квадрата 3×3 , при этом в каждой клетке, кроме одной, стоит один участник, а оставшаяся клетка является пустой.

При этом участники стоят в лифте настолько тесно, что возможно только перемещение участника, стоящего рядом со свободной клеткой, в эту самую клетку (при этом клетка, которую он занимал, становится свободной). Канадских участников заинтересовал вопрос, за какое минимальное количество перемещений они смогут добиться расстановки, при которой участник с номером 1 стоит в левом верхнем углу, далее в верхнем ряду — участники с номерами 2 и 3, в следующем ряду (слева направо) идут участники с номерами 4,5,6, участник с номером 7 стоит в левом нижнем углу, рядом с ним стоит участник с номером 8, а правый нижний угол является свободным).

Формат входного файла

Во входном файле задана изначальная расстановка участников в лифте — 3 строки по 3 символа в каждой. Участники с номерами от 1 до 8 обозначены соответствующими цифрами, пустое место обозначено знаком решётки.

Формат выходного файла

Выведите минимальное количество перемещений, необходимое участникам для того, чтобы добиться требуемой расстановки, или `'impossible'`, если такая расстановка недостижима ни за какое количество перемещений.

Пример

standard input	standard output
146 837 #52	<code>impossible</code>
123 456 78#	<code>0</code>
123 456 7#8	<code>1</code>

Задача C. Front page

Имя входного файла: **standard input**
Имя выходного файла: **standard output**
Ограничение по времени: **64 Mebibytes**
Ограничение по памяти: **2 seconds**

В этом году — в отличие от всех предыдущих IOI — на официальном сайте была организована прямая трансляция соревнований. В последний момент организаторы получили информацию, что скрипт, генерирующий HTML-страницы, работает с ошибками. Вам поручено написать простейшую программу- верификатор корректности HTML-кода.

Корректности HTML-кода определяется следующими правилами:

```
(html_code) :: <HTML>(body)</HTML>
(body)      :: <BODY>(text)</BODY>
(text)     :: (string) | (string)(text) | (tag) | (tag)(text)
(string)   :: пустая строка или строка из символов с кодами от 32 до 127, исключая знаки > и <
(tag)     :: (bold) | (italics) | (link)
(bold)    :: <B>(text)</B>
(italics) :: <I>(text)</I>
(link)    :: <A HREF=http://(string).com>(text)</A>
```

Формат входного файла

Во входном файле задана одна строка, по длине не превосходящая 1000 символов — HTML-код, требующий проверки. При этом строчные и заглавные латинские буквы считаются различными.

Формат выходного файла

В выходной файл выведите ‘Yes’, если HTML-код удовлетворяет правилам, и ‘No’ в противном случае.

Пример

standard input
<HTML><BODY>XXII IOI RESULTS</BODY></HTML>
standard output
Yes
standard input
<HTML><BODY>IOI onboard score</HTML>
standard output
No

Задача D. Translation

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: `64 Mebibytes`
Ограничение по памяти: `2 seconds`

За время, которое понадобилось организаторам, чтобы предоставить условия для перевода, тренер одной из команд успел написать вирус, который не только размножается в памяти, но и автоматически, и, что главное, вполне адекватно переводит условия задач по программированию.

Этот вирус, существующий в N вариациях, устроен следующим образом. Ежечасно каждый экземпляр вируса, находящийся в памяти, заменяется, в зависимости от своей вариации, определённым количеством вирусов некоторых вариаций. Кроме того, в этот же час он переводит некоторое, также зависящее от вариации вируса, количество страниц из условия IOI, оставшихся непереведёнными. По заданным для каждой вариации параметрам размножения и скорости перевода выясните, за какое количество часов с момента заражения вирус целиком переведёт условия размером P страниц. Считается, что в момент заражения в памяти оказался один вирус в первой вариации.

Например, пусть вирус существует в 2 вариациях — первой и второй, а условия содержат 600 страниц. Экземпляр вируса первой вариации ежечасно заменяется тремя экземплярами вируса первой вариации и одним экземпляром вируса второй вариации, при этом переводя одну страницу. Экземпляр вируса второй вариации заменяется двумя вирусами второй вариации, при этом переводя 101 страницу. Тогда через час после заражения в памяти будут три вируса первой вариации и один вирус второй вариации, при этом одна страница условия будет переведена. Через два часа в памяти будут 9 вирусов первой вариации и 5 вирусов второй вариации, а в условиях $1 + (3 * 1 + 101) = 105$ переведённых страниц условия. Через трое суток в памяти будут 27 вирусов первой вариации, 19 вирусов второй вариации и $105 + (9 * 1 + 5 * 101) = 619$ переведённых страниц — то есть условия будут полностью переведены за три часа.

Формат входного файла

В первой строке входного файла заданы 2 целых положительных числа — количество вариаций вируса N ($1 \leq N \leq 20$) и количество страниц условия P ($1 \leq P \leq 10^9$). В последующих N строках описаны вариации вируса: в i -й строке сначала идут N целых неотрицательных чисел, не превосходящих 1000 — количество вирусов каждой из N вариаций, заменяющих каждый час один вирус вариации i , а затем — одно целое неотрицательное число, не превосходящее 1000 — количество страниц условия, которое переводит за час i -я вариация.

Формат выходного файла

Одно целое число — количество часов, за которое будет переведено условие. Если условие так и не будет переведено целиком, выведите -1 .

Пример

standard input	standard output
2 600	3
3 1 1	
0 2 101	

Задача E. Verdict

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: 64 Mebibytes
Ограничение по памяти: 2 seconds

Для того, чтобы сделать жизнь участников более интересной, организаторы XXII IOI решили несколько модифицировать стандартные сообщения об ошибках.

В частности, сообщение, состоящее из заглавных латинских букв, предлагалось зашифровать следующим способом: каждому символу ставится во взаимно однозначное соответствие некоторый (возможно, тот же самый) символ, и затем все буквы в сообщении заменяются на им соответствующие. Получившаяся после замены строка также состоит только из заглавных латинских букв (однако замена отсутствующей в исходном сообщении буквы другим символом — например, знаком вопроса — вполне допустима). При этом функцию, которая шифрует строки, предлагалось открыть для доступа участникам, разрешив её вызывать не более одного раза в минуту.

Один из членов жюри заметил, что в таком случае участникам будет достаточно применить функцию шифрования некоторое количество раз, чтобы получить оригинальное сообщение. Организаторы поручили вам выяснить по заданным двум сообщениям: оригинальному и однократно зашифрованному, какое минимальное количество раз к зашифрованному сообщению нужно применить функцию шифрования, чтобы в результате получилось оригинальное, или сообщить, что у Вас недостаточно данных для того, чтобы это определить.

Формат входного файла

В первой строке входного файла задано исходное сообщение. Во второй задана строка той же длины — однократно зашифрованное сообщение. Строки непустые, состоят только из заглавных латинских букв, длина каждой строки не превосходит 1000.

Формат выходного файла

В выходной файл выведите целое неотрицательное число — минимальное количество последовательных применений функции шифрования к зашифрованному сообщению, которое бы давало в результате оригинальное сообщение. Если информации для того, чтобы определить это число, у Вас не хватает, выведите -1 .

Пример

standard input	standard output
SOLVED SVEDLO	5
A A	0
A B	-1

Задача F. Yet another coins game

Имя входного файла: `standard input`
Имя выходного файла: `standard output`
Ограничение по времени: `64 Mebibytes`
Ограничение по памяти: `2 seconds`

После того, как результаты соревнований были объявлены фактически сразу же после второго тура, участники могли спокойно ждать закрытия олимпиады. Чтобы занять оставшееся перед закрытием время, двое призёров IOI решили сыграть в следующую игру.

На большом листе клетчатой бумаги по одной из линий проведена вертикальная полоса. Первый участник отмечает какую-то клетку справа от вертикальной полосы. Второй расставляет некоторое количество монет слева от этой полосы. Далее он пытается достичь клетки, отмеченной первым, по следующим правилам: можно «перепрыгнуть» одной из монет другую, если монеты находятся в клетках, граничащих по стороне, и клетка, в которую «перепрыгивает» первая монета, свободна. При этом та монета, через которую перепрыгивают, снимается с доски.

Например, пусть вертикальная линия разделяет клетки с x -координатами -1 и 0 , первым игроком отмечена клетка $(1, 0)$, а второй участник расставил монеты в клетках $(-2, 0)$, $(-1, 0)$, $(-1, 1)$ и $(-1, 2)$. Тогда он сначала «перепрыгивает» первой монетой вторую, получая монеты в клетках $(0, 0)$, $(-1, 1)$ и $(-1, 2)$. Затем он «перепрыгивает» монетой из $(-1, 2)$ монету на $(-1, 1)$, получая 2 монеты в клетках $(0, 0)$ и $(-1, 0)$. Последним ходом он «перепрыгивает» монету на $(0, 0)$, достигая отмеченной клетки.

Ваша задача — определить, какое минимальное количество монет потребуется второму игроку, чтобы достичь отмеченной клетки, или выяснить, что отмеченная клетка недостижима ни при каком количестве монет.

Формат входного файла

Во входном файле задано одно целое неотрицательное число x , не превосходящее 31 — расстояние от отмеченной первым клетки до разделяющей прямой.

Формат выходного файла

Выведите одно число — минимальное количество монет, которое потребуется второму игроку для достижения отмеченной клетки, или -1 , если ни при каком начальном расположении монет второму игроку отмеченной клетки достигнуть не удастся.

Пример

standard input	standard output
1	2