

Задача А. Системы счисления

Имя входного файла: bases.in
Имя выходного файла: bases.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Чак Норрис досчитал до бесконечности. Дважды.

Интернет-фольклор

Юный математик Вовочка столь неленив, что выписал в ряд все натуральные числа в порядке возрастания. В порыве трудолюбия он перевел все числа ряда в систему счисления с основанием b . Потом Вовочка вычеркнул все числа ряда, которые являются записью какого-либо числа в системе счисления с основанием c .

Трудолюбие Вовочки закончилось, и он попросил Вас, как своего друга-программиста, найти n -ое невычеркнутое число ряда. Помогите юному математику Вовочке!

Формат входного файла

Единственная строка входного файла содержит три числа n , b и c ($1 \leq n \leq 10^7$, $2 \leq c < b \leq 36$). В системах счисления с основаниями, большими 10 в качестве цифр кроме 0, ..., 9 используются также строчные буквы латинского алфавита a, ..., z.

Формат выходного файла

В выходной файл выведите (в десятичной системе счисления) n -ое невычеркнутое число ряда.

Примеры

bases.in	bases.out
2 3 2	5
1 3 2	2
1 36 35	35

Задача В. Передача данных

Имя входного файла: `data.in`
Имя выходного файла: `data.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Данные передаются от передатчика к приемнику по каналу связи в виде последовательности битов. Особенности канала связи таковы, что в процессе передачи последовательности битов в нее могут быть внесены некоторые искажения. Таким образом, последовательность битов t , принятая приемником, может отличаться от последовательности s , переданной передатчиком. Один из типов искажений, который может быть внесен в процессе передачи, — так называемая *транспозиция соседних битов*. При таком искажении s и t отличаются ровно в двух соседних позициях i и $i + 1$, причем $s_i = t_{i+1}$, $s_{i+1} = t_i$, а для всех $j \neq i, j \neq i + 1$ верно равенство $s_j = t_j$.

Вы являетесь одним из исследователей, занимающихся проблемами передачи информации. В настоящее время Вы работаете над системой кодирования для важного правительственного канала связи. Разрабатываемый Вами код должен обладать свойством исправления одной транспозиции соседних битов. Для того, чтобы разрабатываемый код был как можно более эффективным, Вам необходимо получить информацию об уровне надежности этого канала. Надежность тестируется следующим образом — по каналу передается несколько последовательностей битов s_1, s_2, \dots, s_n , последовательности t_1, t_2, \dots, t_n , которые были приняты приемником, запоминаются, а затем сравниваются с теми, которые были переданы. На основании того, какая доля последовательностей передана без искажений или с внесением одной транспозиции и делается вывод о надежности канала.

Так как выбранные для передачи последовательности достаточно длинные, то выполнять их сравнение вручную очень трудоемко. Поэтому Вы решили написать программу, выполняющую это сравнение.

Необходимо написать программу, которая по двум заданным последовательностям s и t , определяет, можно ли получить t из s , выполнив не более одной транспозиции соседних символов.

Формат входного файла

Первая строка входного файла содержит последовательность s , вторая строка — последовательность t . Обе последовательности непусты, содержат только нули и единицы, и имеют длину не более 100000 символов.

Формат выходного файла

Если последовательность t может быть получена из последовательности s указанным образом, выведите в выходной файл слово YES, в противном случае — выведите в выходной файл слово NO.

Примеры

data.in	data.out
0001010 0010010	YES
0001010 0010101	NO
0001010 0001010	YES

Задача С. Строки Фибоначчи

Имя входного файла: `fib1.in`
Имя выходного файла: `fib1.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В математике достаточно часто применяются так называемые рекуррентные соотношения. Обычно они применяются для задания числовых последовательностей, но могут применяться и для задания последовательностей строк.

Одним из примеров строк, задаваемых рекуррентным соотношением являются *строки Фибоначчи* F_0, F_1, \dots . Они задаются следующим образом: $F_0 = a, F_1 = b, F_i = F_{i-2}F_{i-1}, i > 1$. Первые семь строк Фибоначчи выглядят следующим образом: `a, b, ab, bab, abbab, bababbab, abbabbababbab`.

Дима занимается в кружке олимпиадного программирования и интересуется алгоритмами на строках. Недавно он узнал о строках Фибоначчи. Он быстро понял, что их длина с увеличением номера i растет очень быстро, поэтому задача нахождения всех символов строки F_i требует слишком большого объема памяти. Поэтому он решил ограничиться задачей нахождения некоторых символов.

Напишите программу, которая находит k -ый символ строки F_i .

Формат входного файла

Входной файл содержит несколько наборов входных данных. Первая строка входного файла содержит целое число T наборов входных данных ($1 \leq T \leq 100$). Каждая из последующих T строк описывает один набор входных данных и содержит по два целых числа: n и k ($0 \leq n \leq 45, 1 \leq k \leq |F_n|$, как $|F_n|$ обозначена длина строки F_n , позиции символов в строке нумеруются с единицы).

Формат выходного файла

Выведите в выходной файл T строк, каждая из которых должна содержать ровно один символ — ответ для соответствующего набора входных данных.

Примеры

<code>fib1.in</code>	<code>fib1.out</code>
4	a
0 1	b
1 1	a
3 2	a
7 7	

Задача D. Дробные фокусы

Имя входного файла: `frac.in`
Имя выходного файла: `frac.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Мальчик Сережа очень хочет научиться жонглировать и показывать фокусы. Недавно его друг показал ему один фокус — попросил Сережу загадать некоторую положительную дробь $\frac{a}{b}$, затем назвать ее знаменатель и две дроби $\frac{p_1}{q_1}$ и $\frac{p_2}{q_2}$, такие, что $\frac{p_1}{q_1} < \frac{a}{b} < \frac{p_2}{q_2}$, причем между ними нет меньшей дроби с тем же знаменателем, что и у дроби, загаданной Сережей. После этого Сережин друг отгадал числитель загаданной Сережей дроби.

Теперь Сережа хочет научиться сам показывать этот фокус. Помогите ему в этом — напишите программу, находящую числитель загаданной дроби.

Формат входного файла

Во входном файле через переводы строк заданы три дроби в формате <числитель>/<знаменатель> ($p_1/q_1, a/b, p_2/q_2$), где вместо числа a указан вопросительный знак «?». Все числители и знаменатели являются положительными целыми числами и не превышают 1000.

Формат выходного файла

В выходной файл выведите минимальное число a , такое, что $\frac{p_1}{q_1} < \frac{a}{b} < \frac{p_2}{q_2}$. Если такого не существует, выведите «-1» (без кавычек).

Примеры

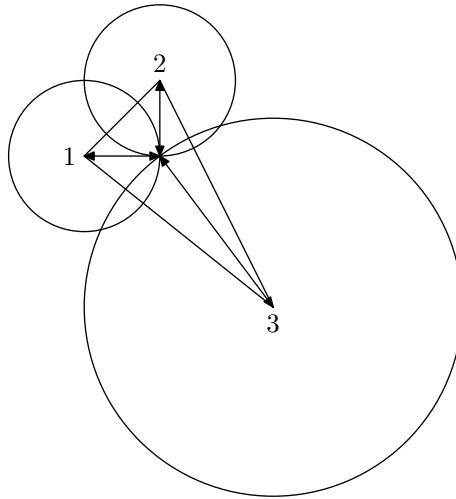
<code>frac.in</code>	<code>frac.out</code>
1/2 ?/8 3/4	5/8
3/20 ?/2 4/10	-1

Задача Е. Система глобальнейшего позиционирования

Имя входного файла: `gps.in`
Имя выходного файла: `gps.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Недавно во Флатландии было решено создать *Новейшую Систему Глобальнейшего Позиционирования*. Поскольку страна занимает бесконечно большой участок плоскости, то вывод спутников очень затруднителен, поэтому было решено ограничиться наземным методом позиционирования.

Для этого во Флатландии было построено три радиовышки, не находящиеся на одной прямой. Объект, который хочет узнать свое местоположение, посылает вышкам сигнал. По силе сигнала, дошедшего до вышек, определяется расстояние между вышками и объектом.



Напишите программу, которая реализует последний компонент системы, который, получая координаты вышек и расстояния от объекта до каждой из них, находит координаты объекта.

Формат входного файла

В первой строке входного файла три пары чисел x_1, y_1, x_2, y_2, x_3 и y_3 — координаты вышек. Во второй строке три неотрицательных числа — расстояния до соответствующих вышек. Все числа во входном файле целые и по модулю не превышают 50.

Формат выходного файла

Если не существует такого местоположения объекта, что расстояния до вышек соответствовали бы данным, то выведите в выходной файл единственное слово «Impossible». Иначе выведите два числа — координаты объекта. Ответ будет проверяться с точностью до шести знаков после запятой.

Примеры

<code>gps.in</code>	<code>gps.out</code>
0 4 2 6 5 0 2 2 5	2.000000 4.000000
0 0 0 3 1 -4 4 5 5	4.000000 0.000000
0 0 1 0 0 1 2 2 2	Impossible

Задача F. Преобразование Капрекара

Имя входного файла: `kaprekar.in`
Имя выходного файла: `kaprekar.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Индийский математик Д. Р. Капрекар известен своими работами по теории чисел. Одна из его работ посвящена так называемому преобразованию Капрекара. Рассмотрим следующую операцию. Пусть задано число x . Пусть M — наибольшее число, которое можно получить из x перестановкой его цифр, а m — наименьшее число (это число может содержать ведущие нули). Обозначим как $K(x)$ разность $M - m$, дополненную при необходимости ведущими нулями так, чтобы число цифр в ней было равно числу цифр в x .

Например, $K(100) = 100 - 001 = 099$, $K(2414) = 4421 - 1244 = 3177$.

Капрекар доказал, что если начать с некоторого четырехзначного числа x , в котором не все цифры равны между собой, и последовательно применять к нему эту операцию (вычислять $K(x)$, $K(K(x))$, ...), то рано или поздно получится число 6174. Для него верно равенство $K(6174) = 7641 - 1467 = 6174$, поэтому на нем процесс заиклится.

Ваша задача состоит в том, чтобы написать программу, вычисляющую $K(x)$ по числу x .

Формат входного файла

Входной файл содержит целое число без ведущих нулей x ($1 \leq x \leq 10^9$).

Формат выходного файла

В выходной файл выведите $K(x)$.

Примеры

<code>kaprekar.in</code>	<code>kaprekar.out</code>
100	099
2414	3177

Задача G. Налеее-во!

Имя входного файла: leftturn.in
Имя выходного файла: leftturn.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Одним погожим деньком на плацу проходили учения. Если точнее, то шла отработка тактически важных строевых приемов: команд «направо», «налево», «кругом» и «шаг вперед». Стояла пятиградусная жара, и солдаты скучали, в отличии от работника спецслужб потенциального врага, прибывшего на плац с целью оценки боевой готовности войск.

Разведчика звали Смит, и трудился он в поте лица, делая снимки настолько часто, что буквально через три часа у него закончилась пленка. Проклиная себя за безалаберность, он покинул место проведения учений и вернулся лишь через 28 минут 12 секунд, захватив на этот раз с собой все снаряжение.

Внимательно изучив обстановку, Смит понял, что за прошедшее время лишь один солдат сменил свое месторасположение. Поскольку ему не хочется признаваться в провале, он решил проверить пару интриг и списать в конечном счете нехватку снимков на пожар в одной из африканских деревень. Смит — парень изворотливый, и такого рода вещи для него не составляют труда. Единственное, что осталось узнать — двигались ли за время его отсутствия другие солдаты, ведь они могли просто вернуться на свое место в ходе сложного тактического маневра. Кроме того, начальство может спросить, сколько команд было отдано в тот интервал времени, что не отражен на пленке.

Теперь перед Смитом стоит задача: узнать, за какое минимальное число команд солдат мог переместиться из одной позиции в другую. Для наглядности, представим плац прямоугольным полем размера $N \cdot M$, а солдата на нем — фигурой, занимающей три подряд идущие смежные клетки. Далее проиллюстрировано выполнение команд.

Команда «налево»

```
.....  
..... \..  
..\-/. ...|..  
..... /..  
.....
```

Команда «направо»

```
.....  
..... /..  
..\-/. ...|..  
..... \..  
.....
```

Команда «кругом»

```
.....  
..\-/. .../-\  
.....  
.....
```

Команда «шаг вперед»

```
..... \-/  
..\-/. .....  
.....  
.....
```

Вам, как человеку проверенному, поручено войти в доверие к Смицу, решив для него эту задачу.

Формат входного файла

В первой строке входного файла содержатся два целых числа N и M ($1 \leq N, M \leq 100$). Далее следуют N строк по M символов каждая — описание исходного положения солдата на плаце. Формат описания аналогичен примерам выше. Символом «*» задаются препятствия — клетки, занимать которые солдат в процессе своего перемещения не может — так Смит обозначил других солдат и противопехотные мины.

Далее в аналогичном формате следует описание конечное положение солдата. Гарантируется, что все препятствия остались на своих местах.

Формат выходного файла

В выходной файл выведите минимальное количество команд, которое необходимо отдать солдату, чтобы он переместился из начального положения в конечное. Если же такое перемещение невозможно, выведите в выходной файл число «-1».

Примеры

leftturn.in	leftturn.out
4 7/-\\	3
4 7 ***** .../-\\ *****	-1

Задача Н. Перестановка букв

Имя входного файла: `lterets.in`
Имя выходного файла: `lterets.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

По результатам исследования одонго английского университета, не имеет значения, в каком порядке расположены буквы в слове. Говорят, чтобы преобразить и расположить буквы были на месте. Остальные буквы могут располагаться в любом беспорядке, все равно текст читается без проблем. При этом это является то, что мы не читаем каждую букву по отдельности, а все слово целиком.

Ваша задача — подтвердить исследования английских ученых для произвольного текста.

Каждое слово *s* в тексте, если это возможно, необходимо преобразовать так, чтобы первая и последняя буква остались на своих местах, а остальные буквы стояли вперемешку, но так, чтобы преобразованное слово не совпадало с исходным. Если же слово указанным образом преобразовать невозможно, то его следует оставить без изменений.

Напишите программу, выполняющую указанное преобразование.

Формат входного файла

Во входном файле задан текст, состоящий из заглавных и строчных латинских букв, знаков препинания: «!?:;-.» и пробелов. Словом считается любая последовательность подряд идущих букв, ограниченная пробелами, знаками препинания, переводом строки, началом или концом текста. Длина текста не превышает 10 килобайт, длина каждого слова не превышает 20 символов. Последняя строка файла завершается переводом строки.

Формат выходного файла

В выходной файл выведите тот же самый текст, каждое слово в котором преобразовано согласно условию. В примере, для пояснения, пробелы заменены подчеркиваниями. Ваша программа должна выводить пробелы.

Примеры

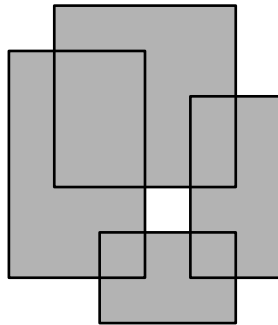
<code>lterets.in</code>	<code>lterets.out</code>
<code>Hello_World!</code>	<code>Hlleo_Wolrd!</code>
<code>Po__rezultatam...</code>	<code>Po__rzeluatatm...</code>
<code>QaQ_QaaaaQ</code>	<code>QaQ_QaaaaQ</code>

Задача I. Приготовление сэндвича

Имя входного файла: sandwich.in
Имя выходного файла: sandwich.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Паша очень любит готовить сэндвичи. Свой фирменный сэндвич «Купе» он готовит из четырех главных ингредиентов: верхний кусок хлеба, ломтик сыра, ломтик ветчины и нижний кусок хлеба. Все ингредиенты имеют прямоугольную форму. Однажды он решил быстро приготовить этот сэндвич. Чтобы это сделать, он подбросил ингредиенты в воздух в надежде, что они упадут и сами сложатся в нужном порядке. Однако ингредиенты упали на стол беспорядочным образом.

Паша также является большим поклонником игры «Крестики-нолики», поэтому скатерть на его столе выглядит как клетчатая плоскость. Быстро введя на своем столе координатную сетку, Паша вычислил в какие клетки какой ингредиент упал, причем выяснилось, что каждый ингредиент оказался прямоугольником с углами в точках с целыми координатами и сторонами, параллельными осям координат.



Теперь Паше стало интересно, какую суммарную площадь на столе занял его сэндвич. Ваша задача в этом, чтобы написать программу, вычисляющую эту площадь.

Формат входного файла

Входной файл содержит по четыре строки, в каждой по четыре целых числа — координаты левого нижнего и правого верхнего угла очередного ингредиента.

Координатная сетка введена так, что первая координата возрастает слева направо, а вторая снизу вверх.

Все числа во входном файле по модулю не превышают 1000.

Формат выходного файла

В выходной файл выведите единственное целое число — ответ на задачу.

Примеры

sandwich.in	sandwich.out
2 0 5 2 0 1 3 6 1 3 5 7 4 1 6 5	35
0 0 2 1 0 0 1 2 1 0 2 2 0 1 2 2	4