

Задача А. Стикеры

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

У Ильдара дома на большой стене висит прямоугольная доска со сторонами a и b . Нижняя сторона этой доски параллельна полу. Каждый день, просыпаясь, он наклеивал на стену по одному одинаковому стикеру, для того чтобы написать на нем расписание очередного дня, при этом стикер частично накрывал доску. Ильдар наклеивал каждый стикер так, чтобы его нижняя сторона была параллельна полу.

Спустя k дней у Ильдара закончились стикеры, и он пошел в магазин за новыми. Там он понял, что не знает, какого размера стикеры были бы лучше. Он запомнил, что за эти k дней он полностью покрыл доску стикерами, то есть каждая точка доски оказалась покрыта одним из наклеенных стикеров (или находилась на его границе). Это свойство ему очень понравилось, поэтому Ильдар решил купить стикеры такого целого размера s , что за следующие k дней он снова сможет покрыть доску стикерами. Будем считать, что пространства стены достаточно, чтобы наклеить стикер в любую позицию относительно доски.

Конечно, запутавшись в вычислениях, Ильдар попросил вас о помощи. Напишите программу, которая по числам a , b , k определит, стикеры какого минимального целого размера s он должен купить, чтобы за следующие k дней он смог покрыть ими доску.

Формат входных данных

В единственной строке записаны 3 целых числа a , b , k , разделенных пробелами — размеры доски и количество дней, соответственно ($1 \leq a, b, k \leq 10^{18}$; $a \cdot b \leq 10^{18}$).

Формат выходных данных

Выведите минимальное целое число s , такое что за k дней Ильдар сможет покрыть стикерами со стороной s всю доску.

Примеры

стандартный ввод	стандартный вывод
4 4 4	2
2 5 1	5
7 4 5	4

Замечание

В первом тесте, если Ильдар купит стикеры со стороной 2, то поскольку квадрат 4×4 можно разбить на 4 квадрата 2×2 , то четырьмя стикерами он сможет покрыть доску. Очевидно, что стикеров со стороной 1 ему не хватит, поскольку за четыре дня он может покрыть часть доски, не превосходящую 4 по площади.

Во втором тесте, если Ильдар купит стикеры со стороной 5, то прямоугольник 2×5 он легко сможет покрыть за один день. Если же взять стикеры с меньшей стороной, то за один день никак не получится покрыть прямоугольник со стороной 5.

Задача В. Большие экраны

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	2 секунды
Ограничение по памяти:	256 мегабайт

В наше время у довольно многих людей плохое зрение, но им тоже может понадобится что-нибудь считать. Именно поэтому наша компания «Краборачие», используя самые передовые технологии, скоро выпустит специальные калькуляторы для слабовидящих людей!

Публичное заявление компании
«Краборачие»

Вам, как ведущему программисту этой компании, поручили реализовать самую важную часть — вывод числа на экран калькулятора.

Все выводимые числа неотрицательные, строго меньше 10^{300} . Цифры должны быть символьными прямоугольниками 7×5 , такими как показано ниже.

```
....X  XXXXX  XXXXX  X...X  XXXXX  XXXXX  XXXXX  XXXXX  XXXXX  XXXXX
....X  ....X  ....X  X...X  X....  X....  ....X  X...X  X...X  X...X
....X  ....X  ....X  X...X  X....  X....  ....X  X...X  X...X  X...X
....X  XXXXX  XXXXX  XXXXX  XXXXX  XXXXX  ....X  XXXXX  XXXXX  X...X
....X  X....  ....X  ....X  ....X  X...X  ....X  X...X  ....X  X...X
....X  X....  ....X  ....X  ....X  X...X  ....X  X...X  ....X  X...X
....X  XXXXX  XXXXX  ....X  XXXXX  XXXXX  ....X  XXXXX  XXXXX  XXXXX
```

Соседние цифры следует разделять столбцами 7×1 , заполненными точками.

Формат входных данных

В единственной строке входных данных находится целое число n , которое надо вывести на экран калькулятора ($0 \leq n < 10^{300}$).

Формат выходных данных

Выведите число n так, как показано в примере.

Пример

стандартный ввод	стандартный вывод
123X.XXXXX.XXXXXX....X....XX....X....XX.XXXXX.XXXXXX.X.....XX.X.....XX.XXXXX.XXXXX

Задача С. Тест на терпение

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Мастер Оби-Ван решил дать много одинаковых задачек Энакину для тренировки терпения. А именно, он дал ему n троек чисел a_i, b_i, c_i и попросил найти для каждой тройки два числа x_i и y_i таких, что побитовое И этих двух чисел равно a_i , побитовое ИЛИ этих чисел равно b_i , а их разность $x_i - y_i$ равна c_i . У Энакина нет времени на решение задач, поэтому он просит вас сделать это задание вместо него. Учтите, что Оби-Ван торопился на совет джедаев, поэтому среди троек могут быть такие, для которых решение не существует.

Формат входных данных

В первой строчке дано целое число n — количество троек ($1 \leq n \leq 10\,000$).

В следующих n строчках даны описания троек — три целых числа a_i, b_i, c_i ($0 \leq a, b, c \leq 10^9$).

Формат выходных данных

Если ответ для i -й тройки существует, то выведите в одной строчке два целых числа x_i, y_i . Если существует несколько правильных пар чисел, являющихся ответом, разрешается вывести любую.

Если не существует, то выведите в одной строчке -1 .

Пример

стандартный ввод	стандартный вывод
3	12 10
8 14 2	6 3
2 7 3	-1
8 13 4	

Замечание

Побитовое И двух неотрицательных целых чисел a и b определяется так. Сначала два числа записываются в двоичной системе счисления одно над другим так, чтобы последняя цифра второго числа оказалась точно под последней цифрой первого числа. Далее, если записи чисел оказались неравной длины, то более короткое из двух чисел дополняется слева нулями, пока цифр в числах не станет поровну. Далее под этими двумя числами записывается третье двоичное число по следующим правилам: если на i -м месте в первом и втором числе стоят 1, то в третьем числе на i -м месте записывается 1; в противном случае там записывается 0. Построенное таким образом третье число и является двоичной записью побитового И данных двух чисел.

Например, рассмотрим числа 17 и 71. В двоичном виде они записываются как 10001 и 1000111. Дополним первое число двумя нулями слева, чтобы в нём стало семь цифр, как и во втором числе: 0010001. Напишем под парой цифр 1, если они обе равны 1, и 0 иначе:

```
0010001
1000111
0000001
```

Если стереть лидирующие нули, получится 1 — двоичная запись числа 1. Таким образом, $17 \text{ AND } 71$ — побитовое И чисел 17 и 71 — равно 1.

Похожим образом определяется побитовое ИЛИ. Мы так же выписываем числа, дополняем нулями, но под двумя цифрами ставим 1 в случае, если хотя бы одна из цифр равна 1, а иначе ставим 0. В частности, для тех же чисел 17 и 71 имеем:

```
0010001
1000111
1010111
```

Полученная двоичная запись представляет число $2^6 + 2^4 + 2^2 + 2^1 + 2^0 = 87$. Таким образом, $17 \text{ OR } 71$ — побитовое ИЛИ чисел 17 и 71 — равно 87.

В языке Pascal побитовые И и ИЛИ вычисляются с помощью команд $a \text{ and } b$ и $a \text{ or } b$, а в C++, Java, Python — с помощью $a \& b$ и $a | b$.

Задача D. Обмен

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Недавно Штирлиц узнал, что Мюллер едет в страну Труляляндию, ввиду чего ему будет необходимо обменять валюту. Штирлиц просто не мог не использовать этот шанс, чтобы проследить за Мюллером. К сожалению, Труляляндия известна своей сложной валютной системой. В ней есть n типов банкнот номиналами $a_1 < a_2 < \dots < a_n$ тугриков. Исторически сложилось, что при всех $k > 1$ номинал a_k делится на a_{k-1} . Достоверные источники сообщили, что Мюллер хочет обменять деньги так, чтобы получить b тугриков. Помогите Штирлицу понять, какое минимальное количество банкнот может получить Мюллер, чтобы у него была сумма ровно в b тугриков.

Формат входных данных

В первой строке записано число n — количество различных номиналов в Труляляндии ($1 \leq n \leq 30$).

Далее идут n целых чисел a_1, a_2, \dots, a_n — номиналы банкнот ($1 \leq a_i \leq 10^9$).

В следующей строке идет число b — количество тугриков, которое надо выдать ($0 \leq b \leq 10^{18}$).

Формат выходных данных

Выведите «Impossible» без кавычек, если это невозможно. Иначе выведите n целых чисел, где i -е число обозначает, сколько следует выдать банкнот i -го типа.

Примеры

стандартный ввод	стандартный вывод
3 1 2 10 33	1 1 3
1 2 3	Impossible

Задача Е. Картошка

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

В младшей группе детского сада «Иои» двум озорникам Джону и Полу не спится во время тихого часа. Чтобы развлечься, они решили сыграть друг с другом в картошку. Правила этой легендарной игры таковы: игроки по очереди выкрикивают слово «картошка», и первый, кого услышит воспитатель, проиграет, так как получит нагоняй.

Как опытные игроки, Джон и Пол великолепно контролируют громкость, с которой произносят слово «картошка», и могут выкрикивать его с любой положительной целой *громкостью*, по-другому называемой *силой крика*, находящейся в их голосовом диапазоне (у каждого из игроков этот диапазон свой, однако минимальная сила крика у обоих равна единице). Благодаря этому за месяцы практики они выяснили, что обычный воспитатель в детском саду имеет некоторый шумовой порог h , и если ребёнок выкрикнет слово с силой, меньшей h , то воспитатель ничего не заметит, но если крик будет силы h или больше, то воспитатель услышит шум и явится к его источнику. Однако сегодня во время тихого часа дежурит Марк, который нестандартно воспринимает звуки: сила криков, произведённых за игру, *накапливается*. Это значит, что если в течение игры участники кричали с громкостями a_1, a_2, \dots , то вместе с криком, с которым сумма всех громкостей впервые окажется не меньше шумового порога Марка h , игра сразу закончится. Другими словами, если

$$a_1 + a_2 + \dots + a_{n-1} < h \leq a_1 + a_2 + \dots + a_{n-1} + a_n,$$

то игра закончится ровно с n -м криком, и проиграет тот, кто его издаст.

Итак, ребята приступают к игре и, начиная с Джона, по очереди кричат слово «картошка». Джон не очень хорош в стратегической части, поэтому он заблаговременно сообщил вам диапазоны громкости его и Пола, а также шумовой порог Марка. Определите, может ли Джон действовать так, чтобы гарантированно победить, и если да, то посоветуйте ему, с какой силой крикнуть в первый раз.

Формат входных данных

В единственной строке входного файла находится три целых числа h, R_J, R_P — шумовой порог Марка, наибольшая сила крика Джона, наибольшая сила крика Пола ($1 \leq h, R_J, R_P \leq 10^{18}$). Это значит, что на каждом своём ходу Джон кричит с одной из громкостей $1, 2, \dots, R_J - 1, R_J$, а Пол — с одной из громкостей $1, 2, \dots, R_P - 1, R_P$.

Формат выходных данных

Если Джон может победить, выведите в первой строке слово «John» без кавычек, а во второй строке — одно целое число W_J — силу, с которой он должен крикнуть, чтобы у него до конца игры существовала выигрышная стратегия ($1 \leq W_J \leq R_J$). Если есть несколько подходящих чисел, вы можете вывести любое из них.

Если Джон не может победить, выведите в первой строке слово «Paul» без кавычек.

Примеры

стандартный ввод	стандартный вывод
6 7 8	John 5
6 4 4	Paul

Замечание

В первом тесте Джон может крикнуть с силой, которая на единицу меньше шумового порога Марка; тогда, как бы ни крикнул Пол, Марк это заметит.

Задача F. Сбор свидетельских показаний

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Айдар Гульниллин — заместитель главного детектива Мимополиса. После ограбления музея, посвящённого семейству курообразных, детектив дал Айдару задание — допросить свидетелей. Лениво потянувшись, тот достал список всех, кто находился в музее во время инцидента, и принялся по очереди звонить им и задавать вопросы насчёт ограбления (внимательно слушать не требовалось, так как все разговоры с телефона главного детектива записываются).

Слова свидетелей были довольно скучными. Они включали много лишних подробностей, в том числе описание некоторых экспонатов и пересказ передачи, увиденной вчера по телевизору. Ухудшалась ситуация тем, что каждый рассказывал не только то, что знал и видел сам, но и вещи, которые слышал от знакомых. И тут-то Айдара осенило — это же плюс ситуации, а не минус! Если он узнает у свидетеля всё, что видел в музее его друг, то этому другу, в принципе, можно даже не звонить!

Айдар отлично знает психологию людей: каждый из свидетелей расскажет всё, что он видел, всем своим знакомым, попросив каждого из них не передавать дальше эту информацию. Таким образом, чтобы узнать свидетельские показания любого человека, нужно просто позвонить ему или любому из его знакомых и допросить. Другими словами, если заместитель детектива позвонит любому из свидетелей, то тот расскажет всё, что видел лично он, и то, что видел лично каждый из его знакомых (несмотря на их просьбы никому этого не сообщать — всё-таки речь идёт о звонке от заместителя главного детектива).

Айдар уже не собирается отступать от своего намерения звонить свидетелям именно в том порядке, в котором они указаны в списке. Теперь ему интересно: после какого по счёту звонка у него впервые будут собраны показания от всех людей из списка? Поскольку от этого зависит, во сколько Айдар сможет отчитаться о сделанной работе и пойти домой, тот пошёл на рискованный шаг: открыл очень жирную папку, помеченную словами «Максимальная секретность», нашёл там досье на всех свидетелей и сообщил вам, кто из них с кем знаком. Ответьте на вопрос Айдара, используя эти сведения.

Формат входных данных

В первой строке находятся два целых числа n и m , разделённых пробелом — количество свидетелей и количество пар знакомых между собой людей ($1 \leq n \leq 100\,000$; $0 \leq m \leq 100\,000$). В каждой из следующих m строк находится по два целых числа a_i и b_i , разделённых пробелом — номера i -й пары знакомых свидетелей ($1 \leq a_i, b_i \leq n$). Гарантируется, что никакая пара свидетелей не фигурирует в списке дважды, даже в обратном порядке — другими словами, не найдутся никакие два индекса $i \neq j$, что $a_i = a_j$ и $b_i = b_j$, и не найдутся никакие два индекса $i \neq j$, что $a_i = b_j$ и $b_i = a_j$. Гарантируется, что для всех i $a_i \neq b_i$: никто не считается своим собственным знакомым.

Формат выходных данных

Выведите одно целое число x — наименьшее количество звонков, за которое Айдар соберёт информацию всех n свидетелей ($0 \leq x \leq n$).

Примеры

стандартный ввод	стандартный вывод
4 3 4 3 3 2 2 1	3
3 3 1 2 1 3 2 3	1
3 2 1 3 1 2	1
3 1 1 2	3
3 1 3 1	2
6 0	6

Замечание

В первом тесте четвёртый свидетель знаком только с третьим, поэтому Айдару придётся допросить или третьего, или четвёртого свидетеля, а с ними и первых двух, поскольку допрос ведётся в порядке, в котором люди указаны в списке. При этом, если Айдар допросит первых трёх свидетелей, то он получит от них всю их информацию, а также получит информацию о четвёртом свидетеле от третьего. Поэтому ответ — 3.

Во втором и третьем тесте первый свидетель знаком со всеми остальными, поэтому, допросив его, Айдар сразу получит все необходимые показания.

В четвёртом тесте третий свидетель ни с кем не знаком, так что его необходимо допросить лично. Значит, потребуется звонить всем троим.

В пятом тесте надо допросить первых двух свидетелей, так как второй ни с кем не знаком, а вот третьего — не надо, поскольку его показания уже сообщил первый.

Задача G. Обеденное время

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Как известно, динозавры — большие животные, а значит, им требуется много еды для выживания. Современные же реалии таковы, что чтобы позволить себе пропитание, сначала надо его заработать. Вот динозавр Гера и сидит на работе в ожидании, когда уже наконец-то можно будет поесть.

Строгие правила в его офисе позволяют уходить на обед только в целое число часов, например, в 03:00 или 16:00. К счастью, его начальник еще не смотрел на часы сегодня, поэтому если их быстро перевести на другое время, то он не заподозрит обмана и отпустит Геру на обед, как только минутная стрелка покажет на ноль минут.

Проблема заключается в том, что часы уже довольно старые, и с некоторыми шестеренками есть проблемы, поэтому часы можно перевести не более чем на a минут назад и не более чем на b минут вперед. Также нельзя переводить часы на следующий или на предыдущий день, потому что это будет слишком заметно.

Вам дано, какое время сейчас показывают часы. Подскажите Гере, на какое время их следует перевести, чтобы отправиться за едой как можно раньше.

Формат входных данных

В первой строке ввода записано текущее время на часах в формате HH:MM (текущее число часов и минут двузначными числами через двоеточие без пробелов). Допустимые значения времени лежат от 00:00 до 23:59 включительно.

Во второй строке через пробел указаны числа a и b — максимальные возможные границы перевода времени в минутах ($0 \leq a, b < 720$)

Формат выходных данных

В том же формате, как и во входных данных, выведите время, на которое следует перевести часы, чтобы они показали целое число часов как можно раньше. Если таких способов перевести часы несколько, выведите любой.

Примеры

стандартный ввод	стандартный вывод
11:30 30 29	11:00
00:01 0 58	00:59

Задача Н. Часовщик

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

В мастерской «У Сени» делают лучшие настенные часы! У очередного покупателя нестандартный запрос: он хочет, чтобы часы были как можно тоньше. У Сени есть несколько стрелок разной толщины, про каждую известно, может ли она использоваться в качестве часовой, минутной или секундной стрелки часов (одна и та же стрелка может быть нескольких типов).

Чтобы собрать часы, надо выбрать три стрелки, первая из которых может использоваться как часовая, вторая — как минутная, третья — как секундная. При этом их суммарная толщина должна быть минимальна. Требуется вывести наименьшую возможную суммарную толщину трёх стрелок или сообщить, что часы собрать невозможно.

Формат входных данных

В первой строке дано число n — число стрелок ($0 \leq n \leq 5 \cdot 10^5$).

Далее идут n строк — описание стрелок.

На i -й строчке находится число w_i — толщина i -й стрелки ($1 \leq w_i \leq 10^9$). Далее в этой же строке через пробел даны три числа: h_i, m_i, s_i — может ли использоваться ли соответствующая стрелка в качестве часовой, минутной или секундной соответственно ($0 \leq h_i, m_i, s_i \leq 1$). Если число равно 1, стрелку можно вставить на соответствующее место, иначе - нельзя.

Формат выходных данных

Если невозможно сделать часы из имеющихся стрелок, выведите «Impossible» без кавычек, иначе выведите суммарную толщину стрелок, которые Сения должен выбрать.

Примеры

стандартный ввод	стандартный вывод
4 3 1 0 0 1 1 1 0 2 0 1 0 1 0 0 1	4
2 3 1 1 1 2 1 1 1	Impossible

Задача I. Треугольные числа

Имя входного файла: стандартный ввод
Имя выходного файла: стандартный вывод
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Первоклассник Сеня проспал урок по геометрии. Разумеется, в целях воспитания и прививания Сене любви к учебе, его учительница поставила ему в дневник кол.

Поскольку это было сделано лишь с намерением показать Сене, что плохо пропускать уроки, через какое-то время учительница сжалась и предложила Сене исправить оценку. Чтобы избавиться от кола в дневнике, Сеня должен выполнить одно несложное задание на пропущенную тему.

Задание Сеня получил следующее: ему надо найти n **различных** натуральных чисел a_1, a_2, \dots, a_n , любые три из которых могут быть сторонами некоторого треугольника на плоскости. Чтобы учительнице было не так сложно проверять его ответ, все числа должны быть не более 10^9 .

К сожалению, Сеня пропустил тот самый урок про треугольники, поэтому он просит помощи у знакомых. Помогите Сене найти такие n натуральных чисел.

Формат входных данных

В единственной строке ввода задано целое положительное число n — требуемое количество чисел ($3 \leq n \leq 10^5$).

Формат выходных данных

Выведите n различных натуральных чисел, удовлетворяющих условию, разделенных пробелами.

Примеры

стандартный ввод	стандартный вывод
3	10 15 23
5	10 4 9 7 8

Задача J. Объявления

Имя входного файла:	стандартный ввод
Имя выходного файла:	стандартный вывод
Ограничение по времени:	1 секунда
Ограничение по памяти:	256 мегабайт

Миша очень любит ездить на электричках. Особенно он любит слушать объявление перед отправлением, когда машинист оглашает, на каких станциях остановится данная электричка. Миша заметил, что эти объявления звучат не всегда одинаково. За многие годы поездок в электричках он запомнил, какие существуют способы объявить остановки.

У каждой электрички есть маршрут, вдоль которого расположены станции. На каждой из этих станций электричка может как остановиться, так и проследовать станцию без остановки. Машинист может выбрать один из трех способов, чтобы объявить, на каких именно станциях остановится электричка (разумеется, объявления делаются на английском, чтобы туристы могли их понимать):

1. «`Train goes with all stops`» — можно сделать такое объявление, если электричка останавливается на **всех** станциях вдоль маршрута.
2. «`Train goes with stops {...}`» — можно сделать такое объявление, при этом вместо многоточия нужно перечислить все станции, на которых остановится электричка.
3. «`Train goes with all stops except {...}`» — можно сделать такое объявление, при этом вместо многоточия нужно перечислить все станции, на которых электричка **не останавливается**.

Сегодня Миша сел в электричку, и ему стало интересно, какой способ объявления остановок нужно выбрать машинисту, чтобы произнести объявление как можно быстрее. Будем считать, что скорость произнесения объявления пропорциональна суммарному количеству букв в нем. Обратите внимание, что знаки препинания, а также паузы между словами не учитываются.

Миша заранее посмотрел список всех станций вдоль маршрута электрички, а также узнал, на каких из станций она будет останавливаться. Помогите Мише определить, какое объявление нужно сделать машинисту, чтобы произнести его как можно быстрее.

Формат входных данных

Первая строка входных данных содержит единственное целое число n — общее количество станций вдоль маршрута электрички ($1 \leq n \leq 10^4$).

Каждая из следующих n строк содержит название очередной станции вдоль маршрута электрички. Название каждой станции является последовательностью из заглавных и строчных латинских букв. Суммарная длина всех названий станций не превосходит 10^4 .

Следующая строка входных данных содержит единственное целое число m — количество станций, на которых останавливается электричка ($1 \leq m \leq n$).

Каждая из следующих m строк содержит название очередной станции, на которой останавливается электричка. Гарантируется, что каждое из этих названий встречается среди названий всех станций вдоль маршрута.

Формат выходных данных

Выведите единственное целое число — минимальное количество букв в объявлении об остановках электрички.

Примеры

стандартный ввод	стандартный вывод
6 Murino Devyatkino Lavriki Kapitolovo Kuzmolovo Toksovo 4 Devyatkino Toksovo Kuzmolovo Murino	44
8 Fili Kuntsevo Poselok Setun Nemchinovka Trehgorka Bakovka Odintsovo 3 Kuntsevo Setun Odintsovo	40
3 Mercury Venus Earth 3 Mercury Venus Earth	21

Замечание

В первом тесте из примера самое короткое объявление выглядит так: «Train goes with all stops except Lavriki, Kapitolovo». Объявление содержит 44 буквы.

В втором тесте из примера самое короткое объявление выглядит так: «Train goes with stops Kuntsevo, Setun, Odintsovo». Объявление содержит 40 букв.

В третьем тесте из примера самое короткое объявление выглядит так: «Train goes with all stops». Объявление содержит 21 букву.