

Задача А. Число Фибоначчи

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

Числа Фибоначчи $F_0, F_1, F_2, \dots, F_n$ определяются следующим образом: $F_0 = 0$, $F_1 = 1$, а для любого $n > 1$ выполнено равенство $F_n = F_{n-1} + F_{n-2}$.

По заданному числу n выведите число Фибоначчи F_n .

Формат входных данных

В первой строке задано целое число n ($0 \leq n \leq 92$).

Формат выходных данных

В первой строке выведите число F_n .

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
1	1
2	1
3	2
5	5

Задача В. Три единицы

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

Требуется найти количество последовательностей заданной длины из нулей и единиц, в которых не встречается трёх единиц подряд.

Формат входных данных

В первой строке задано одно натуральное число n — длина последовательностей ($1 \leq n \leq 10^5$).

Формат выходных данных

Выведите количество последовательностей по модулю 12345.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
1	2
4	13

Задача С. Плавные числа

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

Назовем натуральное число плавным, если разность любых двух его соседних цифр не превосходит по модулю единицы. Вам необходимо определить количество N -значных плавных чисел.

Формат входных данных

В единственной строке задано одно число N ($1 \leq N \leq 20$).

Формат выходных данных

Выведите одно число — искомое количество.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
1	9

Задача D. Максимальная сумма

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

Дана последовательность чисел a_1, a_2, \dots, a_n . Разрешается выбрать из неё подпоследовательность $a_{i_1}, a_{i_2}, \dots, a_{i_k}$ ($1 \leq i_1 < i_2 < \dots < i_k \leq n$, $0 \leq k \leq n$) такую, чтобы любые два индекса i_r и i_s отличались больше, чем на 1 (таким образом, в подпоследовательности не могут попасть два числа, стоящие рядом в исходной последовательности). Как осуществить этот выбор так, чтобы сумма чисел в подпоследовательности была максимально возможной?

Формат входных данных

В первой строке ввода задано целое число n — длина последовательности ($1 \leq n \leq 10\,000$). Во второй строке заданы n целых чисел a_1, a_2, \dots, a_n через пробел — сама последовательность ($-10\,000 \leq a_i \leq 10\,000$).

Формат выходных данных

В единственной строке выведите максимально возможную сумму чисел в подпоследовательности.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 1 2	2
1 -1	0
4 -10 10 10 -10	10

Задача Е. Лестница

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

У лестницы n ступенек, пронумерованных числами $1, 2, \dots, n$ снизу вверх. На каждой ступеньке написано число. Начиная с подножия лестницы (его можно считать ступенькой с номером 0), требуется взобраться на самый верх (ступеньку с номером n). За один шаг можно подниматься на одну или на две ступеньки. После подъёма числа, записанные на посещённых ступеньках, складываются. Нужно подняться по лестнице так, чтобы сумма этих чисел была как можно больше.

Формат входных данных

В первой строке записано целое число n ($1 \leq n \leq 100$). Во второй строке заданы целые числа a_1, a_2, \dots, a_n через пробел ($-10\,000 \leq a_i \leq 10\,000$) — это числа, записанные на ступеньках.

Формат выходных данных

В первой строке выведите одно число — максимальную сумму, которую можно получить, поднявшись по данной лестнице.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 1 2	3
2 2 -1	1
3 -1 2 1	3

Задача F. Гвоздики

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

На прямой дощечке вбиты гвоздики. Любые два гвоздика можно соединить ниточкой. Требуется соединить какие-то пары гвоздиков ниточками так, чтобы к каждому гвоздику была привязана хотя бы одна ниточка, а суммарная длина всех ниточек была минимальна.

Формат входных данных

В первой строке записано число N — количество гвоздиков ($2 \leq N \leq 100$). В следующей строке записано N чисел — координаты всех гвоздиков (неотрицательные целые числа, не превосходящие 10 000).

Формат выходных данных

Выведите единственное число — минимальную суммарную длину всех ниточек.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
5 4 10 0 12 2	6

Задача G. Кирпичи

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

Дано бесконечное количество кирпичей двух цветов — красного и синего. Красные кирпичи имеют длину 2 дециметра, а синие — 3 дециметра. Сколько различных способов выложить ряд из кирпичей длины n дециметров? Способы считаются различными, если на каком-то одинаковом расстоянии от начала ряда в них лежат кирпичи разного цвета.

Формат входных данных

В первой строке задано целое число n ($0 \leq n \leq 80$).

Формат выходных данных

В первой строке выведите одно число — количество способов выложить ряд из кирпичей длины n дециметров.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2	1
6	2

Задача Н. Ядра

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

Капитан Вася всегда держит на своем корабле запас пушечных ядер для борьбы с пиратами. Так как он привык во всем поддерживать порядок, он хранит ядра в виде пирамид. Каждый из слоёв одной пирамиды является заполненным ядрами равносторонним треугольником, сторона которого содержит ровно k ядер. Сторона основания пирамиды состоит из n ядер, в следующем слое сторона состоит из $n - 1$ ядра, и так далее, пока на вершину не будет положено одно ядро (которое является равносторонним треугольником со стороной 1).

Например, пирамида размера 3 состоит из трех уровней, выглядящих так (сверху вниз):

```

X

X
X X

X
X X
X X X
    
```

Ясно, что каждый из треугольников может содержать только 1, 3, 6, 10, ... ядер. Таким образом, пирамида может содержать только 1, 4, 10, 20, ... ядер.

Вася отправляется в плавание и берет с собой ровно m ядер. Какое минимальное число пирамид требуется ему сложить из них на своем корабле?

Формат входных данных

В первой строке ввода записано количество тестов $1 \leq T \leq 20$. В последующих T строках задается количество ядер в i -м тесте $1 \leq m_i \leq 300\,000$.

Формат выходных данных

Для каждого из T тестов выведите в отдельной строке минимальное количество пирамид.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
5	1
1	2
5	3
9	3
15	2
91	

Задача I. Зайчик

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

Зайчик прыгает по прямой просеке, для удобства разделённой на n клеток. Клетки пронумерованы по порядку натуральными числами от 1 до n . Некоторые клетки заболочены: если зайчик прыгнет на такую клетку, ему несдобровать. Некоторые другие клетки просеки поросли вкусной зелёной травой: прыгнув на такую клетку, зайчик сможет отдохнуть и подкрепиться.

Зайчик начинает свой путь из клетки с номером 1 и хочет попасть в клетку с номером n , по пути ни разу не провалившись в болото и скушав как можно больше вкусной зелёной травы. Конструктивные особенности зайчика таковы, что из клетки с номером k он может прыгнуть лишь в клетки с номерами $k + 1$, $k + 3$ и $k + 5$.

Выясните, какое максимальное количество клеток с травой сможет посетить зайчик на своём пути.

Формат входных данных

В первой строке задано число n — количество клеток ($2 \leq n \leq 1000$). Вторая строка состоит из n символов; i -й символ соответствует i -й клетке просеки. Символ «w» обозначает болото, символ «.» — зелёную траву, а символ «.» соответствует клетке без каких-либо особенностей. Гарантируется, что первая и последняя клетки не содержат болот и травы.

Формат выходных данных

В первой строке выведите одно число — максимальное количество клеток с травой, которые зайчик сможет посетить на своём пути. Если зайчику не удастся оказаться в клетке с номером n , выведите -1 .

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
4 ."".	2
5 ."w".	0
9 .www.www.	-1

Задача J. Кролик

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

Как и зайчик, кролик прыгает по прямой просеке, для удобства разделённой на n клеток. Клетки пронумерованы по порядку натуральными числами от 1 до n . Некоторые клетки заболочены: прыгать в них кролик боится. Некоторые другие клетки просеки поросли вкусной зелёной травой: прыгнув на такую клетку, кролик прихватит с собой пучок травы.

Кролик начинает свой путь из клетки с номером 1 и хочет попасть в клетку с номером n , по пути ни разу не провалившись в болото. Конструктивные особенности кролика таковы, что из клетки с номером k он может прыгнуть лишь в клетки с номерами $k + 2$, $k + 3$ и $k + 6$.

Поскольку в клетке с номером n кролика ждёт его крольчиха, главное для него — оказаться там как можно быстрее, то есть сделав как можно меньше прыжков. Если маршрутов с минимальным количеством прыжков несколько, кролик предпочтёт тот из них, на котором он посетит как можно больше клеток с травой. Если и таких маршрутов несколько, кролика устроит любой из них.

По данной карте просеки найдите оптимальный маршрут для кролика.

Формат входных данных

В первой строке задано число n — количество клеток ($2 \leq n \leq 1000$). Вторая строка состоит из n символов; i -й символ соответствует i -й клетке просеки. Символ «w» обозначает болото, символ «.» — зелёную траву, а символ «.» соответствует клетке без каких-либо особенностей. Гарантируется, что первая и последняя клетки не содержат болот и травы.

Формат выходных данных

В первой строке выведите через пробел два числа — минимальное количество прыжков k , за которое кролик может попасть из первой клетки просеки в последнюю, и максимальное количество клеток с травой t , которые он при этом сможет посетить. Во второй строке выведите через пробел $k + 1$ число — номера клеток, которые должен посетить кролик, в порядке их посещения.

Если кролику не удастся оказаться в клетке с номером n , выведите одно число -1 .

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
4 ."".	1 0 1 4
5 .w".	2 1 1 3 5
9 .www.www.	-1

Задача K. Длина подпоследовательности

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

Вам требуется написать программу, которая по заданной последовательности находит длину её максимальной невозрастающей подпоследовательности (то есть такой последовательности чисел $a_{i_1}, a_{i_2}, \dots, a_{i_k}$ ($i_1 < i_2 < \dots < i_k$), что $a_{i_1} \geq a_{i_2} \geq \dots \geq a_{i_k}$ и не существует последовательности с теми же свойствами длины $k + 1$).

Формат входных данных

В первой строке задано число n — количество элементов последовательности ($1 \leq n \leq 239\,017$). В последующих строках идут сами числа последовательности a_i , отделённые друг от друга произвольным количеством пробелов и переводов строки (все числа не превосходят по модулю $2^{31} - 2$).

Формат выходных данных

Вам необходимо выдать число k — длину максимальной невозрастающей подпоследовательности.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
5 5 8 10 4 1	3

Задача Л. Невозрастающая подпоследовательность

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

Вам требуется написать программу, которая по заданной последовательности находит максимальную невозрастающую её подпоследовательность (то есть такую последовательность чисел $a_{i_1}, a_{i_2}, \dots, a_{i_k}$ ($i_1 < i_2 < \dots < i_k$), что $a_{i_1} \geq a_{i_2} \geq \dots \geq a_{i_k}$ и не существует последовательности с теми же свойствами длины $k + 1$).

Формат входных данных

В первой строке задано число n — количество элементов последовательности ($1 \leq n \leq 239\,017$). В последующих строках идут сами числа последовательности a_i , отделённые друг от друга произвольным количеством пробелов и переводов строки (все числа не превосходят по модулю $2^{31} - 2$).

Формат выходных данных

Вам необходимо выдать в первой строке число k — длину максимальной невозрастающей подпоследовательности. В последующих строках должны быть выведены (по одному числу в каждой строке) все номера элементов исходной последовательности i_j , образующих искомую подпоследовательность. Номера выводятся в порядке возрастания. Если оптимальных решений несколько, разрешается выводить любое.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
5	3
5 8	2
10 4 1	4
	5

Задача М. Последовательность

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

Задана последовательность из n чисел a_1, a_2, \dots, a_n . *Погноследовательностью* длины k этой последовательности называется набор индексов i_1, i_2, \dots, i_k , удовлетворяющий неравенствам $1 \leq i_1 < i_2 < \dots < i_k \leq n$. Подпоследовательность называется *возрастающей*, если выполняются неравенства $a_{i_1} < a_{i_2} < \dots < a_{i_k}$.

Широко известна задача поиска максимальной по длине возрастающей подпоследовательности. В этот раз предлагается решить другую задачу: найти максимальную по длине последовательность данной последовательности, которая не является возрастающей.

Формат входных данных

В первой строке задано целое число n — число элементов последовательности ($1 \leq n \leq 100$). В второй строке записаны n чисел a_i — элементы последовательности ($0 \leq a_i \leq 1000$).

Формат выходных данных

В первой строке выведите целое число k — длину максимальной по длине последовательности, не являющейся возрастающей, или число 0, если такой не существует. В случае, если искомая подпоследовательность существует, во второй строке выведите k чисел i_j — набор индексов подпоследовательности.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3	3
3 2 1	1 2 3
2	0
1 2	

Задача N. Спуск с горы

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

В одном из горнолыжных курортов Италии проводятся соревнования по горнолыжному спуску. Каждому спортсмену предстоит скатиться с горы на лыжах. На любом этапе спуска участник получает определённое число очков. После прохождения трассы очки суммируются. Участник, набирающий наибольшее количество очков, выигрывает. Гора представляет собой треугольник, в качестве элементов которого выступают целые числа — очки за прохождение этапа. На каждом уровне спортсмену предоставляется выбор — двигаться вниз влево или вниз вправо. Начало спуска — в самой высокой точке горы, конец — в любой из самых низких.

```
  1
 4 3
5 6 7
8 9 0 9
```

Требуется найти максимальное количество очков, которое может набрать спортсмен.

Формат входных данных

В первой строке содержится целое число n — количество этапов ($1 \leq n \leq 100$). Далее следуют n строк, каждая из которых характеризует один уровень. В i -й из этих строк содержится ровно i целых чисел: a_1, a_2, \dots, a_i — количество очков в каждой из позиций ($-100 \leq a_k \leq 100$ для всех $1 \leq k \leq i$).

Формат выходных данных

Выведите одно целое число: максимальное количество очков, которое может набрать спортсмен.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
4 1 4 3 5 6 7 8 9 0 9	20

Задача O. Поле

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

Отряду нужно пересечь прямоугольное поле размера $m \times n$ квадратов, двигаясь из левого верхнего угла в правый нижний и перемещаясь между соседними квадратами только в двух направлениях — вправо и вниз. Поле не очень ровное, но у отряда есть карта, на которой отмечена высота каждого квадрата. Опасность перехода с квадрата высоты h_1 на соседний квадрат высоты h_2 оценивается числом $|h_2 - h_1|$; опасность всех переходов в пути суммируется. Выясните, какова минимальная опасность пути из квадрата $(1, 1)$ в квадрат (m, n) .

Формат входных данных

В первой строке заданы два числа m и n через пробел ($1 \leq m, n \leq 100$). В следующих n строках записано по m чисел в каждой; i -е число j -й из этих строк соответствует высоте квадрата (i, j) . Все высоты — целые числа в диапазоне от 1 до 100, включительно.

Формат выходных данных

Выведите одно число — минимальную опасность пути из квадрата $(1, 1)$ в квадрат (m, n) .

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 2 1 1 1 1	0
4 2 1 2 3 5 3 8 4 7	6
2 3 1 2 2 3 3 1	4

Задача P. Грибы

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

Маша решила навестить свою бабушку. Она взяла с собой две корзинки — одну с пирожками, а другую — пустую, для грибов, которые она хочет собрать по пути.

Для того, чтобы попасть к бабушке, Маше необходимо пройти через лес, который представляет собой прямоугольник размером $m \times n$, в некоторых клетках которого растут деревья, а в некоторых — грибы. Маша выходит из клетки $(1, 1)$ и идёт к бабушке в деревню, расположенную в клетке (m, n) . Каждым своим ходом Маша может пойти вправо или вниз (то есть увеличить одну и только одну из своих координат на 1), если в клетке, в которой она после этого окажется, не стоит дерево. Если в обеих клетках и справа, и снизу, находятся деревья, то Маша считается заблудившейся.

Вам необходимо по данному лесу выяснить, может ли Маша дойти до бабушки, не заблудившись, и если может, то посчитать максимальное количество грибов, которое она может при этом собрать.

Формат входных данных

В первой строке находятся четыре числа m, n, g, t ($2 \leq m, n \leq 100$, $0 \leq g, t \leq g + t \leq m \cdot n - 2$). В следующих g строках расположены по два числа в каждой — x и y -координаты i -го гриба. За ними следуют t строк с описаниями деревьев в аналогичном формате. Ни в какой клетке не может расти больше одного гриба, гриб и дерево одновременно, или больше одного дерева. Кроме того, в клетках $(1, 1)$ и (m, n) ничего не растёт.

Формат выходных данных

Если Маша может дойти до бабушки, то в первой строке необходимо выдать максимальное количество грибов, которое она сможет при этом собрать, а в последующих $m + n - 1$ строках нужно выдать координаты клеток, последовательно посещаемых Машей, в формате $x_i y_i$, для пути, на котором достигается максимальное количество грибов. Если таких путей несколько, то разрешается выдавать любой из этих путей.

В противном случае нужно вывести единственное число -1 .

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
4 4 3 2 1 4 2 3 4 3 2 2 3 4	2 1 1 1 2 1 3 2 3 3 3 4 3 4 4
2 2 0 2 1 2 2 1	-1

Задача Q. Редакционное расстояние

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

В информатике *редакционным расстоянием* между двумя строками называется минимальное количество добавлений, удалений и замен символов, при помощи которых можно из одной строки получить другую. К примеру, редакционное расстояние между строками «ab» и «ab» равно нулю, так как строки равны между собой безо всяких изменений; расстояние между строками «short» и «ports» равно трём: в слове «short» нужно удалить из начала букву «s», заменить «h» на «p» и добавить в конец букву «s». Редакционное расстояние также называют *расстоянием Левенштейна*.

Найдите редакционное расстояние между двумя заданными строками.

Формат входных данных

В первой строчке задана одна строка, во второй — другая. Длины обеих строк — от 1 до 100 символов, включительно.

Формат выходных данных

Выведите единственное число — редакционное расстояние между двумя заданными строками.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
ab ab	0
short ports	3

Задача R. Наибольшая общая подпоследовательность

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

Даны две последовательности. Найдите длину их наибольшей общей подпоследовательности (подпоследовательность — это то, что можно получить из данной последовательности вычёркиванием некоторых элементов).

Формат входных данных

В первой строке записано число N — длина первой последовательности ($1 \leq N \leq 1000$).

Во второй строке записаны члены первой последовательности (через пробел) — целые числа, не превосходящие 10 000 по модулю.

В третьей строке записано число M — длина второй последовательности ($1 \leq M \leq 1000$).

В четвёртой строке записаны члены второй последовательности (через пробел) — целые числа, не превосходящие 10 000 по модулю.

Формат выходных данных

Выведите единственное целое число: длину наибольшей общей подпоследовательности, или же число 0, если такой не существует.

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 1 2 3 4 2 1 3 5	2

Задача S. Три последовательности

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

Даны три последовательности целых чисел. Ваша задача — найти их наибольшую общую подпоследовательность.

Формат входных данных

Входные данные содержат описание трёх последовательностей. Каждая последовательность задаётся в двух строках. Первая из них содержит длину последовательности n ($1 \leq n \leq 100$), а вторая — её элементы (32-битные целые числа).

Формат выходных данных

Первая строка должна содержать длину максимальной общей подпоследовательности. Саму подпоследовательность необходимо вывести во второй строке. Если таких подпоследовательностей несколько, можно вывести любую из них.

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
3 1 2 3 3 2 1 3 3 1 3 5	2 1 3
3 1 2 3 3 4 5 6 3 1 3 5	0

Задача Т. Шаблоны

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

Многие операционные системы используют шаблоны для ссылки на группы объектов: файлов, пользователей и прочего. Ваша задача — реализовать простейший алгоритм проверки шаблонов для имён файлов.

В этой задаче алфавит состоит из маленьких букв английского алфавита и точки («.»). Шаблоны могут содержать произвольные символы алфавита, а также два специальных символа: «?» и «*». Знак вопроса («?») соответствует ровно одному произвольному символу. Звёздочка «*» соответствует подстроке произвольной длины (возможно, нулевой). Символы алфавита, встречающиеся в шаблоне, отображаются на ровно один такой же символ в проверяемой строчке. Строка считается подходящей под шаблон, если символы шаблона можно последовательно отобразить на символы строки таким образом, как описано выше. Например, строчки «ab», «aab» и «beda.» подходят под шаблон «*a?», а строчки «bebe», «a» и «ba» — нет.

Формат входных данных

Первая заданная строка определяет шаблон P . Вторая строка S состоит только из символов алфавита. Её необходимо проверить на соответствие шаблону. Длины обеих строк не превосходят 10 000. Строки могут быть пустыми — будьте внимательны!

Формат выходных данных

Если данная строка подходит под шаблон, выведите «YES». Иначе выведите «NO».

Примеры

<i>стандартный ввод</i>	<i>стандартный вывод</i>
k?t*n kitten	YES
k?t?n kitten	NO

Задача У. Два шаблона

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

Многие операционные системы используют шаблоны для ссылки на группы объектов: файлов, пользователей и прочего. Ваша задача — найти строку минимально возможной длины, которая подходит под два заданных шаблона.

Алфавит в этой задаче состоит из маленьких букв латинского алфавита и точки («.»). Шаблоны могут содержать любые символы алфавита, а также специальные символы «?» и «*». Под «?» подходит любой символ алфавита, а под «*» — любая строка символов алфавита (возможно, пустая). Под символы алфавита, встречающиеся в шаблоне, подходят только такие же символы алфавита. Строка считается подходящей под шаблон, если символы шаблона можно последовательно отобразить в строку вышеуказанным способом. Например, строки «ab», «aab» и «beda.» подходят под шаблон «*a?», а строки «bebe», «a» и «ba» — нет.

Формат входных данных

Входные данные состоят из одного или нескольких тестовых случаев. В первой строке записано одно целое число — количество тестовых случаев.

Каждый тестовый случай состоит из двух строк, содержащих шаблоны P_1 и P_2 . Длина любого из шаблонов не превосходит 100 символов.

Формат выходных данных

Для каждого из тестовых случаев ответ задаётся одной строкой:

- Если строка, подходящая под оба шаблона, существует, выведите такую строку минимально возможной длины (если таких несколько, разрешается выводить любую).
- В противном случае выведите строку «NO».

Пример

<i>стандартный ввод</i>	<i>стандартный вывод</i>
2 *k*tt*n* *i*e* haha hihi	kitten NO