
Задача А. IPvX

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

Джуманджи — интерактивная игра, поэтому произойти в ней может практически что угодно. Так и сейчас, наши герои случайно свернули не туда и попали в плен к Доктору ДХЦП — главному киберученому-злодею в Джуманджи. Однако, наши герои неинтересны Доктору, поэтому он согласен их отпустить за помощь в задаче, которая уже несколько дней ему не поддается.

У ДХЦП есть n компьютеров в его лаборатории, i -й из которых имеет IPvX-адрес c_i . Этот адрес представляет собой обычный IPv4 адрес, но с одним изменением: групп в нем может быть не 4, а целых x штук. Таким образом, каждый IPvX адрес выглядит следующим образом: $a_1.a_2.a_3.\dots.a_x$, где каждое из чисел a_i находится в диапазоне от 0 до 255 включительно. Все адреса у Доктора упорядочены довольно логичным образом: $a_1.a_2.\dots.a_x$ считается меньше $b_1.b_2.\dots.b_x$, если для некоторого $0 \leq k < x$, $a_1 = b_1, a_2 = b_2, \dots, a_k = b_k$ и $a_{k+1} < b_{k+1}$. Также Доктор использует понятия «следующий» и «предыдущий» адреса, которые определяются как минимальный возможный адрес больше текущего и максимально возможный адрес меньше текущего соответственно. Например, следующий адрес для 0.123.123.123 будет 0.123.123.124, для 0.123.123.255 — 0.123.124.0, для 0.255.255.255 — 1.0.0.0.

ДХЦП хочет добавить в свою локальную сеть еще два новых компьютера. В его сети используется следующий алгоритм выдачи IPvX адресов: первый компьютер сначала получит адрес $a = a_1.a_2.a_3.\dots.a_x$, а затем, пока этот адрес занят, будет менять его на следующий после текущего, если такой адрес есть. После того, как первый компьютер получил IPvX адрес, второй компьютер получит адрес $b = b_1.b_2.b_3.\dots.b_x$ и начнет действовать примерно по такому же алгоритму: если текущий адрес занят, он попытается занять предыдущий, если тот есть, и так далее.

Задача, которая стоит перед Доктором — чтобы после получения компьютерами адресов, адрес, полученный вторым компьютером, был следующим для адреса, полученным первым компьютером. ДХЦП понимает, что это далеко не всегда получится сделать, поэтому добавил возможность фиктивного занятия IPvX адресов — он может послать запрос в локальную сеть на занятие какого-либо адреса, после чего адрес будет считаться занятым, хоть и не принадлежит ни одному из компьютеров. Однако, Доктору очень не нравится эта операция, поэтому делать он ее хочет сделать ее как можно меньше раз. Помогите ему — скажите, какое минимальное количество раз ему нужно проделать эту операцию для данных двух адресов $a_1.a_2.a_3.\dots.a_x$ и $b_1.b_2.\dots.b_x$, чтобы после получения компьютерами IPvX адресов, эти адреса были соседними. Если же это невозможно и наши герои навсегда останутся у Доктора ДХЦП в плену, выведите «-1».

Формат входных данных

В первой строке содержатся числа n и x — количество компьютеров в сети и количество групп в IPvX адресах соответственно ($0 \leq n \leq 10^5, 1 \leq x \leq 8$).

В i -й из следующих n строк содержится адрес i -го компьютера в локальной сети c_i . Адрес представлен в формате $p_1.p_2.p_3.\dots.p_x$ ($0 \leq p_j \leq 255$).

В $n + 2$ и $n + 3$ строках содержатся адреса a и b , которые в начале получают первый и второй компьютеры соответственно, в аналогичном формате.

Гарантируется, что все адреса n компьютеров попарно различны, а также что a меньше b .

Формат выходных данных

В единственной строке выведите минимальное количество операций занятия адреса, которое Доктору нужно сделать, или «-1», если это сделать невозможно.

Система оценки

Баллы начисляются за группу, если пройдены все тесты этой и всех предыдущих групп.

Подзадача	Баллы	Ограничения
1	26	$x = 1, n \leq 256$
2	17	в a и b различаются только последние группы ($a_x < b_x$, остальные a_i и b_i равны)
3	40	в a и b различаются не более 3 последних групп
4	17	полные ограничения

Примеры

стандартный ввод	стандартный вывод
2 4 123.123.123.123 123.123.123.125 123.123.123.121 123.123.123.125	1
2 4 123.123.123.123 123.123.123.125 123.123.123.122 123.123.123.125	-1

Замечание

В первом тестовом примере достаточно занять адрес 123.123.123.124, после этого компьютеры займут адреса 123.123.123.121 и 123.123.123.122 соответственно.

Обратите внимание, что тесты из условия не подходят под тесты первой группы, однако должны быть пройдены для принятия решения на проверку.