

Два маленьких робота перемещаются по строке из символов 'A' и 'B'. Они должны определить, *хорошая* эта строка или *плохая*.

Роботы считают строку *хорошей* если символы 'A' и 'B' встречаются в средней трети строки одинаковое количество раз. Все остальные строки считаются *плохим*. Например, строка "BBABAABVBAVA" *хорошая*, потому что ее средняя треть "AABV" содержит 2 символа 'A' и два символа 'B'; а строка "BAABVBBVBAAB" *плохая*, потому что ее средняя треть "ABVV", содержит 1 символ 'A' и 3 символа 'B'. Строка "AABVBAABV" тоже *плохая*, потому что ее длина не делится на 3 и у нее нет средней трети.

В каждый момент времени, каждый робот стоит на одном из символов строки и получает следующие сведения о своем окружении. Он может видеть, является ли символ, на котором он стоит, самым левым в строке, самым правым в строке, или ни тем, ни другим. Также робот видит, стоит ли другой робот на той же позиции в строке или нет. Наконец, робот видит, на каком символе он стоит – 'A' или 'B'.

Кроме того, каждый робот имеет 4 бита оперативной памяти.

Программа для робота представляет собой список инструкций. Списки инструкций первого и второго роботов могут отличаться.

Каждая инструкция состоит из трех частей:

Левая часть (условие) -> (специальные символы) Правая часть (инструкции)

Левая часть включает:

- Символ, означающий стоит ли робот в одном из концов строки: 'L' (от Leftmost), если робот стоит на левом конце строки, 'R' (от Rightmost), если на правом конце, 'I' (от Inner), если робот не стоит в одном из концов строки.
- Символ, означающий положение другого робота: 'O' (от One), если другой робот стоит на другой позиции, 'T' (от Two), если роботы стоят на одной позиции.
- Символ, означающий символ строки, на котором стоит робот: 'A' или 'B'.
- 4 символа '0' или '1': содержимое оперативной памяти робота.

Таким образом, левая часть - это условие, которая является конъюнкцией (логическим И) простых условий: положения робота относительно концов строки, положения другого робота, символа, на котором стоит робот и содержимого памяти робота.

Правая часть включает:

- Символ, означающий действие, которое робот должен совершить: 'L' (от Left), если робот должен перейти на один символ влево, 'R' (от Right), если робот должен перейти на один символ вправо, 'S' (от Stay), если робот должен остаться на месте, 'Y' (от Yes), если робот считает строку *хорошей*, 'N' (от No), если робот считает строку *плохой*.
- Если предыдущий символ не равен 'Y' или 'N', то 4 символа '0' или '1', означающих новое содержимое памяти.

Каждый символ в строке инструкций (кроме двух специальных символов '->' и символа действия) может быть заменен знаком вопроса '?'. Если какой-то символ в левой части заменен знаком вопроса '?', то соответствующее условие не проверяется. Если один из символов в новом содержимом памяти в правой части заменен знаком вопроса '?', то соответствующий бит не меняется.

Например, строка инструкций "LT???01->R??10" означает следующее: если робот стоит в левом конце строки, другой робот стоит на одной позиции с ним, и два младших бита памяти равны '0' и '1' соответственно, тогда (независимо от символа под роботом) он двигается на один символ вправо, не меняет два старших бита и меняет два младших бита на '1' и '0' соответственно.

Изначально, оба робота находятся в левом конце строки. Все биты в памяти роботов равны '0'.

Каждую миллисекунду роботы одновременно выполняют одно из действий из своего списка инструкций. Робот читает список инструкций последовательно, сверху вниз. Если робот находит условие **без знаков вопроса '?'**, которое в точности совпадает с его текущим состоянием, то он немедленно выполняет действия, соответствующие этому условию. Если точного совпадения нет, то робот **снова читает список инструкций** сверху вниз, находит первое условие (возможно со знаками вопроса '?'), которое совпадает с текущим состоянием и выполняет соответствующее этому условию действие. Если совпадений нет, то роботы выключаются, не решив задачу. Когда какой-либо из роботов отвечает 'Y' или 'N', ответ

задает результат работы роботов, и оба робота выключаются. Если оба робота одновременно дают различные ответы, то результатом работы становится ответ **первого робота**.

Ваша задача - написать два списка инструкций, которые позволят роботам всегда корректно определять, является ли данная строка *хорошей* или *плохой*.

Это **output only** задача. Вы должны отправить текстовый файл robots.txt содержащий:

- Строку "Robot 1"
- Несколько строк инструкций для первого робота, по одной на каждой строке.
- Строку "Robot 2"
- Несколько строк инструкций для второго робота, по одной на каждой строке.

Файл robots.txt может содержать закомментированные строки, которые игнорируются при обработке. Такие строки начинаются с символа процента '%'.
Constraints

Обозначим длину строки как $|s|$. Количество инструкций, которое могут выполнить роботы не может превышать $1000 \times |s|$.

Каждый тест состоит из 16 или 20 строк, длиной от 12 до 3600 символов 'A' и 'B'.

Один тест содержит только строки длины 12.

Еще два теста содержат строки длины не более 36.

Grading

Тесты оцениваются независимо. Баллы за тест начисляются только если роботы не превысили ограничение на количество инструкций и корректно определили тип для каждой строки.

Example

Для пояснения, рассмотрим более простую задачу: пусть роботы считают *хорошими* только строки длины 3, у которых последний символ равен 'A': "AAA", "ABA", "BAA", "BBA". Тогда стратегия проста и для этой задачи достаточно одного робота.

robots.txt	Explanation
Robot 1	
??????->S????	Независимо от состояния, остаться на месте
Robot 2	
L??0000->R0001	Если робот стоит в левом конце строки с памятью "0000" - перейти вправо и изменить память на "0001"
I??0001->R0010	Если робот не стоит в конце строки с памятью "0001" - перейти вправо и изменить память на "0010"
R??0001->N	Если робот стоит в правом конце строки с памятью "0001" - ответить что строка <i>плохая</i> , т.к. состоит из двух символов
R?A0010->Y	Если робот стоит в правом конце строки с памятью "0010" на символе 'A' - ответить что строка <i>хорошая</i>
R?B0010->N	Если робот стоит в правом конце строки с памятью "0010" на символе 'B' - ответить что строка <i>плохая</i> , т.к. последний символ - 'B'
I??0010->N	Если робот не стоит в концах строки с памятью "0010" - ответить что строка <i>плохая</i> , т.к. состоит из более чем трех символов

Interpreter

Для локального тестирования, у вас есть интерпретатор файла robots.txt. Он называется robots.cpp. Скачайте его и откомпилируйте. Используя интерпретатор, вы можете тестировать решение на строках, которые вы вводите. Его можно запускать в консоли если исполняемый файл находится в одной папке с файлом robots.txt. Кроме того, интерпретатор можно запускать в отладочном режиме, когда ходы роботов показываются в пошаговом режиме: после каждого шага показывается состояние роботов и ожидается нажатие ENTER.



Если интерпретатор запущен без аргументов, он читает строку из стандартного ввода и выполняет инструкции из файла robots.txt без отладки.

Если интерпретатор запущен с одним аргументом:

- Если аргумент является строкой из символов 'А' и 'В', то он распознается как строка, на которой запускаются инструкции из файла robots.txt.
- Если аргумент не распознается как корректная строка из символов 'А' и 'В', то интерпретатор читает строку из стандартного ввода и выполняет инструкции из файла robots.txt в отладочном режиме.

Если интерпретатор запущен с двумя аргументами, первый из них распознается как строка для проверки, а наличие второго означает запуск в отладочном режиме.

Например, (в консольном режиме):

Running robots ABAB	проверить строку ABAB без отладки
Running robots ABAB 1	проверить строку ABAB с отладкой
Running robots 1	читать строку из stdin и проверить ее с отладкой

Вы можете использовать и модифицировать интерпретатор как угодно.