

Два маленьких робота повзають по рядку, що складається з символів **A** і **B**. Їх мета - визначити, чи рядок гарний чи поганий.

Два робота вважають, що рядок буде добрим, якщо кількість **As** і кількість **Bs** в його середній третині рівна. Вони вважають усі інші рядки поганими. Наприклад, рядок **ВВАВААВВВВАВА** гарний, оскільки його середня третина, **ААВВ**, містить два **As** та два **Bs**; навпаки, рядок **ВААВАВВВВААВ** поганий, тому що його середня третина, **АВВВ**, містить одну **A** та три **Bs**. Рядок **ААВВААВВ** також поганий, хоча він не має середньої третини.

Кожного моменту, кожен робот займає один символ рядка і знає наступне про своє оточення. Він може бачити, чи він стоїть на лівому символі, внутрішньому символі або правому символі рядка (іншими словами, він може бачити, чи є більше символів ліворуч і праворуч). Він може бачити, чи є він єдиним у своєму положенні, або інший робот займає однакову позицію разом з ним. Нарешті, він може бачити, який символ він займає, **A** або **B**.

Крім того, кожен робот здатний зберігати спогади про його подорожі. На жаль, пам'ять кожного робота обмежена всього чотирма бітами.

У кожного робота є "перелік інструкцій", і ці два списки можуть сильно відрізнятись.

Кожна інструкція складається з трьох частин:

Ліва сторона (*стан*) -> (*спеціальні символи*) Праворуч (*інструкція для дії*)

Ліва частина включає:

- Один символ, який вказує, чи є робот на одному з кінців рядка: L - «Якщо це самий лівий символ», I - «Якщо він має внутрішній символ» або R - «Якщо він знаходиться на символі правого краю»
- один символ, O - "якщо є один робот у цій позиції" або T - "якщо у цьому місці є два робота", вказуючи, чи є робот сам, або інший робот займає ту ж позицію разом з ним
- Символ, на якому робот стоїть, A - «якщо він знаходиться на A», або B - «якщо він знаходиться на B»
- 4 символи, кожен 0 або 1, що описують вміст пам'яті робота, наприклад, 0101 - "якщо пам'ять містить 0101"

Таким чином, ліва частина формує умову, яка є сукупністю цих простих умов і має значення "true" або "false" залежно від поточної позиції робота та історії, що зберігається в її чотирьох бітах пам'яті.

Права сторона виглядає наступним чином:

○ Один символ, що вказує, які дії повинен зробити робот: L - «Переміститись на один крок ліворуч», R - «Переміститись один крок вправо», S - «Залишатися на місці», Y - «Повідомити про те, що рядок гарний», або N - "Звіт про те, що рядок є поганим"

○ За винятком випадків, коли дія виконується (дії Y або N) - ще чотири символи, кожен 0 або 1, що вказує на змінений вміст пам'яті робота, наприклад, 0110.

Будь-який символ у рядку інструкції, окрім двох спеціальних символів, і символ дії може бути заміщений символом підстановки ?.

Коли підстановка символу ? відбувається в лівій частині інструкції (умова), це означає, що відповідний простий стан не перевіряється.

Коли підстановка символу ? відбувається в пам'яті, перезаписана частиною рядка команд (тобто після символу дії), це вказує на те, що відповідні біти пам'яті робота залишатимуться незмінними.

Наприклад, рядок інструкції

LT ??? 01-> R ?? 10

каже: "Якщо робот займає найлівіший символ рядка, інший робот займає ту ж позицію разом з ним, а останні два біти його пам'яті встановлюються на 0 і 1, потім (незалежно від того, чи він стоїть на A або B), він повинен перемістити один крок вправо і перевернути останні два біти своєї пам'яті до 1 і 0, не змінюючи перші два біти".

Обидва робота починають подорож у лівому символі рядка. Спочатку всі біти в кожній дії пам'яті встановлюються в 0.

Кожну мілісекунду кожен робот дивиться навколо, шукає його пам'ять, консультує його список для

керівництва, і вирішує або переміститися на один крок ліворуч, переміститися на один крок праворуч, залишитися на місці або висловити свою думку про рядок. Крім того, робот може здійснити будь-які зміни чи наміри в пам'яті. Як тільки будь-який робот висловлює думку про рядок, їхнє завдання зроблено, і вони вимикаються.

Детально, кожен робот перевіряє послідовно список інструкцій для точної відповідності його поточному стану лівою частиною інструкції. Коли він вперше зустрічається з одним, він виконує знайдену інструкцію. В іншому випадку робот відновлює пошук з початку списку і виконує першу інструкцію, що містить підстановку, яка відповідає його положенню та вмісту пам'яті. Якщо знову не знайдено відповідності, роботи вимикаються, не розпізнаючи різновид рядка (і їх завдання залишається невирішеним).

Ви повинні створити два списки інструкцій, по одному для кожного робота, таким чином, щоб вони завжди правильно визначали, чи заданий рядок довжиною принаймні двох символів, що складається з букв А та В, гарний чи поганий.

Ви повинні створити та надіслати єдиний текстовий файл robots.txt, що містить (у такому порядку):

о Один рядок, що читає Робот 1

о кілька інструкцій, що складають список інструкцій першого робота (одна інструкція в рядку)

о Один рядок, що читає Робот 2

о кілька інструкцій, що складають список інструкцій другого робота (одна команда в рядку).

Файл robots.txt може містити рядки коментарів, які просто пропускаються роботом. Такі лінії починаються з символу% (відсоток).

#### Обмеження

Загальна кількість кроків, які можуть виконувати роботи перед оголошенням типу рядка, не повинна перевищувати 1000 разів довжини рядка.

Кожен тест містить 16 або 20 рядків, кожен з яких має довжину щонайменше 12 і не більше 3600 символів; кожен символ А або В.

Один тест містить лише рядки довжини 12.

Ще два тести містять лише рядки довжиною не більше 36.

#### Оцінка

Кожен тест оцінюється в повні бали, якщо два робота визначатимуть кожен рядок в ньому правильно у кілька кроків, які не перевищують 1000 разів довжини рядка та 0 балів інакше.

#### Приклад

Для додаткового розуміння давайте розглянемо набагато легше завдання: нехай роботи вважають "відмінними" лише рядки довжини 3, з третього символу А. І. е. тільки ААА, АВА, ВАА та ВВА вважаються "гарними". Це можна отримати, використовуючи дуже просту стратегію. По суті, для цього завдання достатньо одного робота, тому ми можемо навчити іншого, щоб залишитися на місці. "Робочий" може рухатись двома кроками вправо і перевіряти, чи він досяг правого кінця, і якщо він стоїть на літері А.

robots.txt	Пояснення
Robot 1	Для робота 1
??????->S????	Незалежно від стану - залишитися на місці
Robot 2	Для робота 2
L??0000->R0001	Якщо лівий кінець закінчиться пам'яттю 0000 - йдіть праворуч і замініть пам'ять 0001
I??0001->R0010	Якщо на внутрішньому символі і з пам'яттю 0001 - йдіть праворуч, пам'ятаючи 0010
R??0001->N	Якщо на самому правильному кінці з пам'яттю 0001 - рядок містить лише два символи, то кажіть, що це "поганим"
R?A0010->Y	Якщо в правій частині кінця, на А і з пам'яттю 0010 - рядок "гарним"
R?B0010->N	Якщо в правій частині кінця, на В і з пам'яттю 0010 - рядок є "поганим"



I??0010->N	Якщо він ще не досяг правого кінця, рядок занадто довгий, оголосити що він "поганий"
------------	--

## Interpreter

Для локального тестування вам надається інтерпретатор файлу robots.txt під назвою robots.cpp. Завантажте його в папку, де ви створюєте файл robots.txt. Використовуючи інтерпретатора, ви можете перевірити своє рішення на рядки, які ви вводите. Це також може бути запущено в режимі консолі, якщо його компілятивна версія розміщена у тій самій папці, де знаходиться robots.txt. Крім того, існує трасування режиму інтерпретатора, коли завдання виконується крок за кроком, показуючи стан роботів і чекаючи ENTER з стандартного вводу.

Запустити без параметрів інтерпретатор приймає рядок зі стандартного вводу, на якому виконується список команд, створений вами у файлі robots.txt, без відстеження. Якщо вказано один параметр:

- Якщо це рядок A і B, цей параметр перевіряється, використовуючи інструкції в robots.txt без трасування;
- Якщо це не визнано як дійсний рядок A та B, інтерпретатор чекає іншого рядка зі стандартного входу та відслідковує трасування.

Якщо параметри є двома, перший з них вважається рядком, який потрібно перевірити, а наявність другого перемикається у режимі трасування.

Наприклад (у схованому режимі):

Запуск роботів ABAB перевіряє рядок ABAB без трасування;

Запуск роботів ABAB 1 перевіряє рядок ABAB з трасуванням;

Запуск роботів 1 чекає на введення консолі, який потрібно перевірити за допомогою трасування.

Звичайно, ви можете використовувати та змінити інтерпретатора, як ви вважаєте за потрібне.