

Автор: Николай Белухов

Две роботчета пълзят по низ, съставен от символите *A* и *B*. Целта им е да установят дали низът е *красив* или *грозен*.

Според роботското разбиране, *красив* е оня низ, в който броят на буквите *A* и броят на буквите *B* в средната третина от низа е един и същ. Всеки друг низ роботчетата смятат за *грозен*. Например, низът *ВВАВААВВВАВА* е красив, защото средната му третина *ААВВ* е съставена от две букви *A* и две букви *B*; напротив, низът *ВААВАВВВВААВ* е грозен, защото средната му третина *АВВВ* съдържа едно *A* и три *B*-та. Грозен е и низът *ААВВААВВ* – той няма средна третина.

Във всеки момент всеки робот заема един символ в низа и възприема някои елементи от обкръжението си. Той знае дали се намира на най-левия символ, на вътрешен символ, или на най-десния символ (с други думи, той вижда дали има още символи наляво и надясно). Той вижда и дали е сам на текущата позиция, или и другият робот е на същата позиция като него. И последно – той вижда символа, на който е „стъпил“: дали е *A* или *B*.

Отделно от това, всяко от двете роботчета запомня някои неща от пътешествието си. За съжаление, паметта на всеки от тях е само 4 бита.

В главата на всяко роботче има „списък с инструкции“. Списъците с инструкции на двете роботчета могат да бъдат съществено различни.

Една инструкция се състои от три части:

Лява част (*условие*) ->(специални символи) Дясна част (*указание за действие*)

Лявата част има вида:

- Един символ, означаващ положение на робота: *L* за „Ако е върху най-левия символ“, *I* за „Ако е върху вътрешен символ“, или *R* за „Ако е върху най-десния символ“
- Един символ, *O* за „ако е сам“ или *T* за „ако са двама“, който определя дали роботът е сам на текущата позиция или другият робот заема същата позиция
- Един символ, определящ това, върху което е „стъпил“ роботът: *A* за „ако е върху *A*“ или *B* за „ако е върху *B*“
- 4 символа, всеки от които *0* или *1*, описващи съдържанието на паметта на робота, напр. *0101* за „ако в паметта има записано *0101*“

По този начин лявата част задава условие, конюнкция от такива *прости условия*, което има стойност „вярно“ или „невярно“ според текущото положение на робота и историята, която се пази в четирите му бита памет.

Дясната част има вида:

- Един символ, определящ какво действие да предприеме робота: *L* за „Премини една стъпка наляво!“, *R* за „Премини една стъпка надясно!“, *S* за „Остани на място!“, *Y* за „Обяви, че стрингът е красив!“ или *N* за „Обяви, че стрингът е грозен!“.
- Освен в случаите, когато работата приключва (действие *Y* или *N*) – още 4 символа, всеки от които *0* или *1*, задаващи новото съдържание на паметта, напр. *0110*

Всеки символ в инструкцията, с изключение на специалната двойка -> и символа, определящ последващото действие на робота, може да бъде заменян с глобалния символ ?.

Когато глобалният символ ? се намира в лявата част (условие), това означава, че съответното просто условие не се проверява.

Когато глобалният символ ? се намира в частта на инструкцията, която променя състоянието на паметта (т.е., след символа за действие), той показва, че съответният бит остава непроменен.

Например, инструкцията:

LT???01->R??10

се разчита така: „Ако роботът заема най-левия символ в низа и е заедно с другия робот на една и съща позиция, и двата последни бита в паметта му са 0 и 1, то, без значение върху коя буква е „стъпил“ (А или В) и все едно какви са първите два бита в паметта му, трябва да премине един символ надясно и да инвертира последните си два бита съответно в 1 и 0, без да променя първите два бита .

Двете роботчета започват движението си от най-левия край на низа. В началото всички битове на паметта им са нули.

На всяка милисекунда всяко от роботчетата се оглежда, консултира се със запомненото в скромната си памет, проверява списъка с инструкции и решава дали да се придвижи един символ наляво, дали един символ надясно, дали да остане на мястото си или да си каже мнението за низа. Допълнително, всяко роботче може да запомня впечатленията или намеренията си в паметта. Щом някое от роботчетата съобщи какъв според него е този низ, работата им е свършена и те се изключват. Ако и двата робота направят съобщения, то е валидно мнението на първия.

По-конкретно, всяко роботче последователно проверява списъка от инструкции за **точно съответствие** на текущото му състояние с лявата част на някоя инструкция. При **първо срещане** на такава, той изпълнява намерената инструкция. В противен случай роботът проверява списъка отначало и изпълнява **първата** инструкция, съдържаща глобален символ, която съответства на положението и състоянието на паметта му. Ако и такава съответствие не се намери, роботите се изключват, без да са установили вида на низа (и задачата им остава нерешена).

Вие трябва да създадете списъци с инструкции за двете роботчета, такива че за всеки низ с дължина **поне два символа**, съставен от буквите А и В, те винаги правилно да определят дали е красив или грозен.

Това е задача от тип **output only**. Вие трябва да създадете и предадете към системата текстов файл robots.txt, който съдържа (в този ред):

- един ред със съдържание Robot 1
- няколко реда с инструкции, съставляващи списъка с инструкции на първия робот (по една инструкция на ред)
- един ред със съдържание Robot 2
- няколко реда с инструкции, съставляващи списъка с инструкции на втория робот (по една инструкция на ред).

Файлът robots.txt може да съдържа коментарни редове, които просто се пропускат от робота. Такива редове имат за първи символ % (знак за процент).

Ограничения

Общият брой на стъпките, които роботите изпълняват, преди да бъде определен видът на низа, не трябва да надхвърля 1000 пъти неговата дължина.

Всеки тест съдържа или 16, или 20 низа, всеки от които е дълъг поне 12 и не повече от 3600 символа, всеки от които А или В.

Един тест съдържа само низове с дължина 12.

Други два теста съдържат низове, не по-дълги от 36 символа.

Оценяване

Всеки тест получава предвидените за него точки, ако роботите определят правилно вида на всеки от съдържащите се в него низове за общ брой стъпки, който не надхвърля 1000 пъти неговата дължина.

Пример

За допълнително изясняване, да разгледаме много по-простата задача: роботите да разпознават като „красив“ само низ, който съдържа точно три символа, третият от които да бъде буквата А. Т. е., красиви да са само ААА, АВА, ВАА и ВВА. Това може да се постигне с много проста стратегия. Всъщност, един робот е достатъчен за тази задача, така че другият можем да си го оставим на място. „Работният“ можем да го придвижим две стъпки надясно, като проверим дали е стигнал най-десния символ в низа и дали той е точно А.

robots.txt	Обяснение
Robot 1	За робот 1
??????->S????	Каквото и да е състоянието – стой си там
Robot 2	За робот 2
L??0000->R0001	Ако си в левия край с памет 0000 – иди надясно с памет 0001
I??0001->R0010	Ако не си в край и паметта ти е 0001 – иди надясно с памет 0010
R??0001->N	Ако си в десния край с памет 0001 – низът е бил само от два символа – обяви го за „грозен“
R?A0010->Y	Ако си в десния край върху символ А с памет 0010 – низът е „красив“
R?B0010->N	Ако си в десния край върху символ В с памет 0010 – низът е „грозен“
I??0010->N	Ако не си стигнал десния край, низът е много дълъг – обяви го за „грозен“

Интерпретатор

За локално тестване ви се предоставя интерпретатор на създадения файл robots.txt, наречен robots.cpp. Запишете го в папката, в която създавате файла robots.txt. Чрез него можете да тествате написаното решение върху въведени от вас низове. Той може да се стартира и в конзолен режим, ако и компилираната му версия се намира в папката, където е robots.txt. Интерпретаторът има и режим на трасиране, който изпълнява задачата стъпка по стъпка, като показва състоянието на роботите и изчаква ENTER от стандартния вход.

Стартиран без параметри, интерпретаторът очаква от стандартния вход низ, за който да изпълни описаните от вас списъци с инструкции в robots.txt без трасиране. При зададен един параметър:

- ако той е низ от А и В, този параметър се проверява по указанията в robots.txt без трасиране;
- ако той не се разпознае като коректен низ за проверка, очаква се друг низ от стандартния вход и се включва трасиране.

Ако параметрите са два, интерпретаторът приема, че първият е низ за проверка, а наличието на втори включва трасиране.

Например (в конзолен режим):

Стартиране на robots АВАВ

включва проверка на низа АВАВ без трасиране;

Стартиране на robots АВАВ 1

включва проверка на низа АВАВ с трасиране;

Стартиране на robots 1

очаква конзолен вход, който да провери с трасиране.

Разбира се, можете да използвате и промените интерпретатора както ви е удобно.