

Doi roboti mici se tarasc pe un sir compus din caractere de a A si de B. Scopul lor e sa determine daca sirul este *rafinat* sau *aspru*.

Cei doi roboti considera ca un sir de caractere este *rafinat* daca in a doua treime a sirului numarul de A-uri si numarul de B-uri este acelasi.

Ei considera toate celelalte siruri de caractere ca fiind *aspre*. Spre exemplu, sirul BBABAABBBABA este *rafinat* deoarece a doua lui treime, AABB, are doua A-uri si doua B-uri; in schimb, sirul BAABABBBBAAB este *aspru* deoarece a doua lui treime, AB BB, contine un A si trei de B. Sirul AABBAABB este *aspru* pentru ca nu are o a doua treime.

La orice moment de timp fiecare robot se afla pe o pozitie a sirului si observa anumite informatii despre mediul inconjurator. Robotul respectiv poate vedea daca se afla pe caracterul cel mai din stanga, daca este la un caracter din interiorul sirului (caracter interior) sau daca este la sfarsitul sirului (cu alte cuvinte, el poate observa daca mai sunt alte caractere la stanga sau la dreapta lui). El mai poate vedea daca este singur la pozitia respectiva sau daca si celalalt robot se afla pe aceeasi pozitie ca si el. In final, el mai poate afla pe ce tip de caracter se afla (A sau B).

Pe langa acestea, fiecare robot e capabil de a retine informatii despre calatoria lui. Din pacate, memoria fiecarui robot este limitata la doar patru biti.

In capul fiecarui robot exista cate o "lista de instructiuni", iar listele pot avea diferente substantiale.

Toate instructiunile sunt construite din trei parti, sub formatul:

Partea stanga (*conditie*) -> (*caractere speciale*) Partea dreapta (*instructiune pentru a actiona*)

Partea stanga include:

- Un caracter indicand daca un robot este la unul din capetele sirului de caractere: "L" apare "daca robotul se afla la cel mai din stanga caracter", I apare "daca robotul se afla la un caracter interior" si R "daca robotul se afla la cel mai din dreapta caracter".
- Un alt caracter, O pentru "se afla doar un robot la pozitia curenta" sau T pentru "se afla doi roboti la pozitia curenta", indicand daca robotul este singur sau daca este pe aceeasi pozitie ca si celalalt robot.
- Un al treilea caracter egal cu cel de pe pozitia robotului, A pentru "daca se afla pe A" si B pentru "daca se afla pe B".
- Alte 4 caractere, fiecare 0 sau 1, descriind bitii din memoria robotului, spre exemplu 0101 pentru "daca memoria robotului este 0101".

In acest fel, partea stanga formeaza o conditie care este o conjunctie intre aceste simple conditii, si are o valoare de "adevar" sau "falsitate", depinzand de pozitia curenta a robotului si memoria lui, stocata pe 4 biti.

Partea dreapta se descrie in modul urmator:

- Un caracter reprezentand ce actiune va lua robotul: L pentru "Muta un pas la stanga", R pentru "Muta un pas la dreapta", S pentru "Stai pe loc", Y pentru "Spune ca sirul este *rafinat*" sau N pentru "Spune ca sirul este *aspru*".
- Exceptand cazurile cand munca e terminata (actiunile Y si N) - patru alte caractere, fiecare 0 sau 1, indicand noua memorie a robotului, spre exemplu 0110.

Orice caracter dintr-o linie de instructiune pe langa cele doua caractere speciale si caracterul de actiune pot fi inlocuite de un wildcard "?".

Cand caracterul wildcard "?" apare in partea stanga a instructiunii (in conditie) inseamna ca acea conditie simpla corespunzatoare nu este verificata.

Cand caracterul wildcard "?" apare in instructiune in partea de memorie noua a robotului, (i. e. dupa caracterul de actiune) bitii corespunzatori din memoria robotului nu vor fi schimbati.

Spre exemplu, linia de instructiune

LT???01->R??10

spune: "Daca robotul ocupa cel mai din stanga caracter al sirului, al doilea robot ocupand si el aceeasi pozitie iar ultimii doi biti din memorie sunt 0 si 1, atunci (nefiind important daca se afla pe un caracter A sau B), el se va deplasa la dreapta cu un pas si va schimba ultimii biti ai memoriei robotului cu 1 si 0, fara a schimba primii doi biti din memorie.

Amandoi robotii vor porni calatoria lor de la caracterul cel mai din stanga al sirului. Initial, toti bitii din memoria fiecarui robot sunt setati la 0.

La fiecare milisecunda, fiecare robot se uita in jur, isi verifica memoria, consulta lista lui pentru instructiuni si decide daca: sa se miste la stanga cu un pas, sa se mute la dreapta cu un pas, sa stea pe loc, sau sa isi spuna parerea despre sirul de caractere. De asemenea, robotul detine memoria lui si o va modifica corespunzator instructiunii alese. Imediat ce orice robot isi spune parerea despre sirul de caractere, munca lor este terminata si ei se vor inchide.

In detaliu, fiecare dintre roboti verifica linie cu linie lista de instructiuni pentru o **potrivire exacta** intre starea curenta cu partea stanga a instructiunii. Cand un robot intalneste **prima** asemenea instructiune, el o executa. In caz contrar, robotul reporneste cautarea de la inceputul listei si executa **prima** instructiune continand un wildcard care corespunde cu pozitia si memoria lui. Daca iar nu gaseste o potrivire, robotii se opresc fara a recunoaste tipul sirului de caractere (si munca lor ramane neterminata).

Misiunea ta este sa creezi doua liste de instructiuni, una pentru fiecare robot astfel incat ei vor reusi de fiecare data sa determine in mod corect daca sirul de caractere este de cel putin doua caractere, compus numai din litere de A si B si daca sirul este *rafinat* sau *aspru*.

Aceasta problema este output only. Tu trebuie sa creezi si sa trimiti un singur fisier robots.txt continand (in aceasta ordine):

- O linie "Robot 1"
- Un numar de linii formand lista de instructiuni pentru primul robot (cate o instructiune pe linie)
- O linie "Robot 2"
- Un numar de linii formand lista de instructiuni pentru al doilea robot (cate o instructiune pe linie).

Fisierul robots.txt poate contine linii de comentariu iar robotii pur si simplu vor trece peste acestea. Astfel de linii vor incepe cu caracterul "%" (percent).

Limite

Numarul comun de pasi pe care cei doi roboti trebuie sa-l faca inainte de a anunta tipul sirului nu trebuie sa depaseasca de 1000 de ori lungimea sirului.

Fiecare test contine fie 16, fie 20 de siruri de caractere, fiecare avand lungimea cel putin 12 si cel mult 3600.

Un test contine doar siruri de caractere de lungime 12.

Doua alte teste contin doar siruri de caractere de lungime cel mult 36.

Testare

Orice test va primi scorul maxim daca cei doi roboti vor cataloga corespunzator sirul de caractere intr-un numar de pasi care sa nu depaseasca de 1000 de ori lungimea initiala a sirului de caractere. In caz contrar se vor acorda 0 puncte pentru testul respectiv.

Exemple

Drept lamurire suplimentara, haideti sa consideram o problema mai simpla: presupunand ca robotii ar considera rafinat doar sirurile de caractere de lungime 3 unde al treilea caracter este A (ex: AAA, ABA, BAA si BBA sunt rafinate). Aceasta problema se poate rezolva utilizand urmatoarea strategie triviala: In primul rand, un singur robot este suficient in acest caz, deci il putem instrui pe celalalt sa stea pe loc. Primul se poate muta 2 pozitii la dreapta si verifica atat ca atins capatul, cat si ca se afla pe un caracter de tipul A.

robots.txt	Explicatie
Robot 1	Pentru robotul 1
??????->S????	Oricare ar fi starea – stai pe loc
Robot 2	Pentru robotul 2
L?0000->R0001	Data te aflii la capatul stang cu memorie 0000 – du-te la dreapta si tine minte 0001
I?0001->R0010	Daca te aflii pe un caracter interior cu memoria 0001 – du-te la dreapta si tine minte 0010
R?0001->N	Data te aflii la capatul drept cu memorie 0001 – sirul de caractere contine doar doua caractere, asa ca raspunde ca sirul este “aspru”
R?A0010->Y	Data te aflii la capatul drept, pe un A si cu memorie 0010 – sirul de caractere este “rafinat”
R?B0010->N	Data te aflii la capatul drept, pe un B si cu memorie 0010 – sirul de caractere este “aspru”
I?0010->N	Daca capatul drept nu a fost inca atins, atunci sirul este prea lung, deci il vom considera “aspru”

Interpretor

Pentru testare locala, vei primi un interpretor pentru fisierul robots.txt numit robots.cpp. Descarca-l in folderul unde este creat fisierul robots.txt. Folosind interpretorul poti testa solutia pe siruri produse de tine local. Interpretorul poate fi rulat si in modul consola daca versiunea compilata a acestuia este plasata in acelasi folder cu fisierul robots.txt. De asemenea, acesta poate rula si in modul “trace” aratand pas cu pas starea robotilor si asteptand tasta ENTER de la intrarea standard.

Rulat fara parametrii, interpretorul accepta un sir de caractere de la intrarea standard, string pentru care va executa listele de instructiuni create de tine in fisierul robots.txt, fara a intra in modul “trace”.

Rulat cu un singur parametru:

- Fie parametrul este recunoscut ca un sir de caractere compus din caractere de A si B, caz in care sirul va fi verificat fara a porni modul “trace”.
- In caz contrar, interpretorul va porni modul “trace” si va astepta sirul de caractere la intrarea standard.

Rulat cu doi parametrii, primul termen este considerat sirul de caractere, iar prezenta celui de-al doilea parametru activeaza automat modul “trace”.

De exemplu (in modul consola):

Ruland robots ABAB verifica sirul ABAB fara “tracing”;
Ruland robots ABAB 1 verifica sirul ABAB cu “tracing”;
Ruland robots 1 asteapta sirul ce va fi verificat la standard input, activand modul tracing.

Desigur ca puteti folosi si modifica sursa interpretorului dupa cum doriti.