

Dve male Alanove mašine (u daljem tekstu roboti) šetaju se po stringu koji se sastoji od slova A i B. Njihov cilj je da odrede da li je string *lep* ili *glup*.

Dva robota smatraju da je string *lep* ukoliko je broj slova A i broj slova B u srednjoj trećini stringa jednak. Oni smatraju da su svi ostali stringovi *glupi*. Na primer, string BBABAABBBABA je lep jer njegova srednja trećina, string AABB, sadrži dva slova A i dva slova B. S druge strane, string BAABABBBBAAB je glup jer je njegova srednja trećina ABBB. String AABBAABB je glup jer nema srednju trećinu.

U svakom trenutku, svaki robot se nalazi na jednom karakteru stringa, i može da oseti svoju okolinu svojim robotskim čulima. Kako ova čula nisu jako istančana, robot može da oseti samo nekoliko stvari. Prva - da li se nalazi na levom kraju stringa, na desnom kraju stringa ili negde između. Druga - da li je sam na toj poziciji ili se tu nalazi i drugi robot. Treća - da li je karakter na kojem stoji A ili B.

Osim toga, ovi roboti imaju i specijalnu memoriju kojom se mogu služiti i koja se sastoji od neverovatnih četiri bita.

Duboko unutar hardvera svakog robota nalazi se snimljena "lista instrukcija", i te dve liste mogu biti različite za različite robote.

Svaka instrukcija sastoji se iz tri dela:

Leva strana (uslov) -> (specijalni karakteri) Desna strana (radnja)

Leva strana sadrži:

- Jedan karakter koji označava gde se nalazi robot. **L** - na levom kraju, **R** - na desnom kraju, **I** - negde unutar stringa.

- Jedan karakter koji označava broj robota na trenutnoj poziciji. **O** - jedan robot, **T** - dva robota.

- Karakter na kojem robot trenutno stoji - **A** ili **B**.

- Četiri karaktera, svaki **0** ili **1**, koji opisuju sadržaj robotove memorije, na primer, **0101**.

Na ovaj način, leva strana formira uslov, koji je logička konjunkcija (veznik *and*) ovih *prostih uslova*, i ima vrednost "tačno" or "netačno", koja zavisi od trenutne pozicije oba robota i memorije robota u skladu sa gore datim opisom.

Desna strana zadaje se na sledeći način:

- Jedan karakter koji označava koju akciju će robot izvesti: **L** - pomeri se jedno mesto levo, **R** - pomeri se jedno mesto desno, **S** - ostani u mestu, **Y** - prijavi da je string lep, **N** - prijavi da je string glup.

- Osim u slučajevima **Y** i **N**, još četiri karaktera, svaki **0** ili **1**, koji označavaju novi sadržaj robotove memorije, na primer **0110**.

Svaki karakter u liniji za instrukciju i za akciju može se zameniti džoker karakterom ?.

Ako se džoker nalazi sa leve strane instrukcije (u uslovu), onda on označava da se taj *prosti uslov* ne proverava.

Ako se džoker pak nalazi sa desne strane instrukcije, i to u delu za pisanje memorije, on označava da se tom bitu memorije ne menja vrednost.

Na primer, sledeća instrukcija

LT???01->R??10

govori: " Ako se robot nalazi na levom kraju stringa i ako se onaj drugi robot nalazi na istoj toj poziciji, i poslednja dva bita memorije su 01 (nebitno da li se robot nalazi na slovu A ili B), pomeri se za jedno mesto udesno i postavi poslednja dva bita memorije na 10 (prva dva ne diraj)".

Oba robota započinju svoju avanturu na levom kraju stringa i u svojoj memoriji sadrže 0000.

Svake pikosekunde, svaki robot posmatra svoje okruženje svojim robotskim čulima i zatim na osnovu svoje liste instrukcija bira instrukciju koju izvršava. odlučuje da se pomeri jedan korak levo, korak desno, ostane na mestu ili se oglasi o lepoti/glupoći stringa. Takođe menja svoju internu memoriju. Čim se neki robot oglasi o lepoti/glupoći stringa, njegova misija je gotova i robot se gasi. Ukoliko se **oba robota** oglase, važi se mišljenje **prvog**.

Preciznije, u svakom koraku, svaki robot prolazi kroz svoju listu instrukcija **redom** i bira **prvu** koja se **tačno poklapa** sa trenutnim stanjem stvari (dakle onu prvu bez džokera sa leve strane). Ako ne uspe da nađe ovakvu instrukciju, on počinje ponovo **od početka** liste i ponovo traži **prvu** instrukciju (sa džokerima) koja se poklapa sa trenutnom situacijom. Ako ni tad ne uspe da nađe pogodnu instrukciju, **oba robota** zatrokiraju i prevrnu se (u

ovom slučaju zadatak ostaje nerešen).

Potrebno je da osmislite dve liste instrukcija, po jednu za svakog robota, tako da mogu tačno da utvrde da li je dati string **koji se sastoji od bar dva karaktera** lep ili glup.

Ovo je **output only** zadatak. Potrebno je da pošaljete jedan tekstualni fajl robots.txt koji sadrži, u sledećem redosledu:

- Jednu liniju sa tekстом: Robot 1
- Listu instrukcija za prvog robota (svaka instrukcija u svom redu)
- Jednu liniju sa tekстом: Robot 2
- Listu instrukcija za drugog robota (svaka instrukcija u svom redu)

Fajl robots.txt može sadržati i komentare, koje robot preskače. Takve linije treba da počinju karakterom % (procent).

Ograničenja

Ukupan broj instrukcija koje roboti smeju da izvrše pre donošenja odluke ne sme biti veći od 1000 puta dužina stringa.

Svaki test primer sadrži 16 do 20 stringova, svaki dužine bar 12 i najviše 3600 karaktera, svaki karakter je A ili B.

Jedan primer sadrži samo stringove dužine 12.

Još dva primera sadrže samo stringove dužine do 36.

Ocenjivanje

Svaki test primer se boduje maksimalnim brojem poena samo ukoliko roboti tačno ocene svaki string iz tog primera i ako i broj instrukcija izvršenih za svaki string ne pređe 1000 puta dužina tog stringa. U suprotnom dobijate 0 poena.

Primer

Radi jednostavnost, posmatrajmo mnogo lakši zadatak: Neka roboti smatraju string lepim ako ako je string dužine 3, gde je treći karakter A. Odnosno, samo AAA, ABA, BAA i BBA su lepi stringovi. Ovo se može izvesti vrlo jednostavnom strategijom - jedan robot je dovoljan. Drugom možemo reći da prosto stoji u mesto sve vreme. "Radilica" može da se pomeri dva koraka u desno i da proveriti da li je do poslednjeg slova i da li stoji na slovu A.

robots.txt	Objašnjenje
Robot 1	Slede instrukcije za robota 1
??????->S????	Šta god da se dešava oko tebe ostani u mestu
Robot 2	Slede instrukcije za drugog robota
L??0000->R0001	Ako si na levom kraju sa memorijom 0000, idi desno i postavi memoriju na 0001
I??0001->R0010	Ako si na unutrašnjem karakteru sa memorijom 0001 – idi desno i zapiši 0010 u svoju pametnu robotsku glavicu.
R??0001->N	Ako si na desnom kraju a memorija je 0001, string se sastoji od samo dva karaktera, odgovori da je string <i>glup</i>
R?A0010->Y	Ako si na desnom kraju na slovu A i sa memorijom 0010, string je <i>lep</i>
R?B0010->N	Ako si na desnom kraju na slovu B i sa memorijom 0010, string je <i>lep</i>
I??0010->N	Ako još nisi došao do desnog kraja, string je predugačak, odgovori da je string <i>glup</i>

Interpreter

Radi testiranja na vašem računaru, dat vam je interpreter za fajlove robots.txt, pod imenom robots.cpp. Skinite ga u folder gde se nalazi fajl robots.txt. Koristeći interpreter, možete testirati svoje rešenje na stringovima sa ulaza. Ovaj interpreter može da radi i u konzolnom režimu, ako se njegova kompajlirana verzija stavi u isti folder gde se nalazi fajl robots.txt. Takođe postoji i *trace* režim rada, gde se izvršenje kretanja robota vrši korak po korak, prikazujući stanje robota i čekajući da lupite ENTER na standardni ulaz.

Ako se pokrene bez parametara, interpreter prihvata stringove sa standardnog ulaza, nad kojima zatim izvršava listu instrukcija zadatu u fajlu robots.txt, bez *trace*-ovanja. Ako se zada jedan parametar:

- Ako je to string slova A ili slova B, ovaj parametar se proverava pomoću instrukcija datih u robots.txt bez *trace*-ovanja;

- Ako se string ne sastoji samo od slova A ili B, onda interpreter čeka da se unese string na standardni ulaz i uključuje se *trace* režim.

Ako su data dva parametra, prvi od njih se smatra za string koji treba da se proveriti, a prisustvo drugog uključuje *trace* režim.

Na primer (u konzolnom režimu):

Pokretanjem: robots ABAB	proverava string ABAB bez <i>trace</i> -a;
Pokretanjem: robots ABAB 1	proverava string ABAB sa <i>trace</i> -om;
Pokretanjem: robots 1	čeka se na konzolni ulaz, koji se proverava bez <i>trace</i> -ovanja.

Svakako, dozvoljeno je da koristite i menjate ovaj interpreter po sopstvenom nahođenju.