

Zadaci

Zadatak	PJESMA	JANICA	SLOVA	TRAMVAJ
Izvršna datoteka	pjesma.exe	janica.exe	slova.exe	tramvaj.exe
Izvorni kôd	pjesma.pas pjesma.c pjesma.cpp	janica.pas janica.c janica.cpp	slova.pas slova.c slova.cpp	tramvaj.pas tramvaj.c tramvaj.cpp
Ulazna datoteka	pjesma.in	janica.in	slova.in	tramvaj.in
Izlazna datoteka	pjesma.out	janica.out	slova.out	tramvaj.out
Vremensko ograničenje (po test podatku)	10 sekundi	10 sekundi	10 sekundi	10 sekundi
Broj test podataka	10	10	10	10
Broj bodova (po test podatku)	3	4	6	7
Ukupno bodova	30	40	60	70
	200			

PJESMA

Pjesma se sastoji od jedne ili više strofa, a strofa se sastoji od **četiri** stiha. Stih se sastoji od jedne ili više riječi međusobno odvojenih s po jednim razmakom, a riječ se sastoji od barem jednog malog ili velikog slova engleske abecede (a-z, A-Z).

Zadnji slog neke riječi je niz znakova od **uključivo zadnjeg** samoglasnika pa do kraja riječi. Ako u riječi nema samoglasnika onda za zadnji slog uzimamo cijelu riječ.

Za dva stiha kažemo da se rimuju ako im se **podudaraju** zadnji slogovi (**zanemarujemo** razliku između malih i velikih slova). Rima u strofi može biti savršena, parna, ukrštena, obgrljena ili pak strofa može biti pisana slobodnim stihom.

Rima u strofi je savršena ako se svi stihovi rimuju (**a a a a**), a **ako nije** savršena onda može biti parna ako se rimuju prvi i drugi stih i treći i četvrti stih (**a a b b**), ukrštena ako se rimuju prvi i treći stih i drugi i četvrti stih (**a b a b**) i obgrljena ako se rimuju prvi i četvrti stih i drugi i treći stih (**a b b a**). Strofa je pisana slobodnim stihom ako se njeni stihovi ne mogu svrstati u niti jednu gore navedenu rime.

Napišite program koji će za svaku strofu u zadanoj pjesmi **odrediti vrstu** rime.

Ulazni podaci

U prvom retku ulazne datoteke nalazi se cijeli broj N , broj strofa u pjesmi, $1 \leq N \leq 5$.

U sljedećih $4N$ redaka nalaze se stihovi pjesme. Maksimalna duljina svakog stiha je 50 znakova.

Izlazni podaci

U N redaka izlazne datoteke treba redom za svaku strofu ispisati kakvim je stihom pisana (vrsta rime ili slodoban stih) tj. jednu od riječi '**savrsena**', '**parna**', '**ukrstena**', '**obgrljena**' ili '**slobodan stih**'.

Test primjeri

pjesma.in

```
1
mi smo Super
Edit Pjaf Ran
gari geri kuper
ej pazi na ekran
```

pjesma.out

ukrstena

pjesma.in

```
2
Tko je to
to je Zlo
nije Zlo
to je Mo
Gdje je Mu
jel na putu
Brijem da nije
vidlo bi se
```

pjesma.out

savrsena
parna

pjesma.in

```
2
mirko me
ozbiljno zanima e
dok ovo pisem
drhtim placem
ali znam da i ja
mogu racunati
na to ma znate sram me
sapce da i ja njega mamim
```

pjesma.out

parna
slobodan stih

JANICA

Zahvaljujući Janici praćenje skijaških utrka postalo je totalno 'in'. Za one koji su malo 'out' evo podsjetnik kako izgleda jedna skijaška utrka na olimpijadi. Utrka se sastoji od dvije vožnje. Prvoj vožnji pristupa N skijašica koje nose startne brojeve od 1 do N i one nastupaju upravo tim redoslijedom.

Za prvu skijašicu u prvoj vožnji (startni broj 1) saznajemo **vrijeme** u kojem je ona prošla stazu. Za svaku sljedeću skijašicu dobivamo njen **zaostatak** ili **prednost** u odnosu na **trenutno vodeću**. U drugu vožnju prolazi M najbolje plasiranih skijašica i nastupaju u obrnutom redoslijedu od plasmana nakon prve vožnje (od M-to plasirane do prvoplasirane).

Slično kao i za prvu vožnju, za prvu skijašicu u drugoj vožnji dobivamo **ukupno vrijeme** u kojem je ona prošla stazu **u obje vožnje**. Za svaku sljedeću skijašicu dobivamo njen **zaostatak** ili **prednost** u odnosu na **trenutno vodeću**.

Napišite program koji će na temelju zadanih vremena i zaostataka **odrediti** koje su skijašice osvojile medalje. Dodatno olakšanje nam je pretpostavka da ni nakon prve ni nakon druge vožnje neće biti dvije skijašice s istim vremenom, te da su sve skijašice završile utrku.

Vrijeme potrebno da neka skijašica prođe stazu u pojedinoj vožnji neće biti manje od 10 sekundi te ukupno vrijeme u obje vožnje neće biti veće od 300 sekundi. Općenito, vrijeme je decimalni broj s maksimalno dvije decimale.

Ulazni podaci

U prvom retku ulazne datoteke nalaze se dva cijela broja N i M, međusobno odvojena jednim razmakom, $3 \leq M \leq N \leq 100$.

U sljedećem retku nalazi se vrijeme prve skijašice iz prve vožnje, a u sljedećih N-1 redaka su zaostaci svake sljedeće skijašice. Analogno, u sljedećih M redaka nalaze se podaci za drugu vožnju.

Izlazni podaci

U prvi redak datoteke treba zapisati startni broj skijašice koja je osvojila zlato, u drugi redak startni broj skijašice koja je osvojila srebro i u treći redak startni broj skijašice koja je osvojila broncu.

Test primjeri

janica.in

3 3
25.13
+1.14
+2.18
45.08
+2.14
+3.11

janica.out

3
2
1

janica.in

4 3
29.18
+2.18
+0.05
+1.13
54.22
+1.23
+1.11

janica.out

4
1
3

janica.in

5 4
27.29
-1.02
+1.83
-0.43
+0.03
56.98
+1.83
-0.43
+0.03

janica.out

5
4
1

SLOVA

Mirko igra jednu jako zanimljivu igru koja se igra na pravokutnoj ploči koja se sastoji od manjih polja poredanih u R redaka i S stupaca. U svakom polju na ploči zapisano je jedno veliko slovo engleske abecede (A-Z).

Na početku igre Mirko ima figuricu koja se nalazi na gornjem lijevom polju (u prvom retku i prvom stupcu). U svakom potezu Mirko može pomaknuti figuricu na jedno od **četiri susjedna** polja na ploči (gore, dolje, lijevo ili desno - **ako postoje**) ali tako da niti u jednom trenutku ne stane na polje na kojem piše slovo na kojem je figurica **ranije već bila**.

Napišite program koji će izračunati **maksimalni** broj polja (**uključujući** i početno polje) koje Mirkova figurica može posjetiti.

Ulazni podaci

U prvom retku ulazne datoteke nalaze se dva cijela broja R i S, međusobno odvojena jednim razmakom, $1 \leq R, S \leq 20$.

U (i+1)-om retku zapisan je niz znakova duljine S koji predstavlja slova na poljima i-tog retka ploče.

Izlazni podaci

U prvi i jedini redak izlazne datoteke treba zapisati traženi maksimalni broj polja.

Test primjeri

slova.in

2 4
CAAB
ADCB

slova.out

3

slova.in

3 6
HFDFFB
AJHGDH
DGAGEH

slova.out

6

slova.in

5 5
IEFCJ
FHFKC
FFALF
HFGCF
HMCCH

slova.out

10

TRAMVAJ

Kada je mladi Julijus došao na studij u Zagreb i prvi put u životu ugledao tramvaj, odmah se prebacio s matematike na promet (tramvajski, naravno) i odlučio honorarno voziti noćne linije, a danju studirati. I tako, polako, došao je do vozačkog ispita. Kao uzoran i nadobudan student-tramvajac, sve je odlično svladao osim rukovanja sa skretnicama.

Mreža tramvajskih linija u gradu se sastoji od stanica i tračnica koje međusobno povezuje **neke** od stanica. Skretnice se nalaze **samo u stanicama** i to u svakoj stanici **točno jedna** skretnica. Tramvaj prilikom izlaska iz stanice može krenuti **samo** u onom smjeru u kojem je **okrenuta** skretnica. Ako želi krenuti prema nekoj drugoj stanici, prvo **mora** okrenuti skretnicu.

I sada, na dan vozačkog ispita, došao mu je instruktor i rekao da proizvoljnim putem dođe do stanice B. Nalazeći se u stanici A i posjedujući uređaj - posljednji krik tehnologije pomoću kojeg Julijus zna u kojem je smjeru okrenuta skretnica u svakoj od stanica, odlučio je odabrati put na kojem će **minimalni** broj puta biti prisiljen okrenuti skretnice.

Napišite program koji će izračunati **minimalni** broj promjena skretnica na putu od A do B.

Ulazni podaci

U prvom retku ulazne datoteke nalaze se cijeli brojevi N, A i B, međusobno odvojeni s po jednim razmakom, $2 \leq N \leq 100$, $1 \leq A, B \leq N$. Broj N označava broj stanica u tramvajskoj mreži.

U svakom od sljedećih N redaka nalazi se niz brojeva međusobno odvojenih s po jednim razmakom. U i-tom retku prvi broj K_i ($0 \leq K_i \leq N-1$) označava broj stanica u koje se može doći direktno iz i-te stanice. Sljedećih K_i brojeva su redni brojevi tih stanica. Skretnica u i-toj stanici na početku je **okrenuta** prema **prvoj** navedenoj stanici.

Izlazni podaci

U prvi i jedini redak izlazne datoteke treba ispisati minimalni broj skretnica koje je potrebno promijeniti. Ako put između stanica A i B ne postoji, treba ispisati '-1'.

Test primjeri

tramvaj.in	tramvaj.in	tramvaj.in
3 2 1	3 1 3	4 4 2
2 2 3	1 2	1 2
2 3 1	2 1 3	1 1
2 1 2	1 2	1 4
		1 3
tramvaj.out	tramvaj.out	tramvaj.out
0	1	-1