

Zadaci

Zadatak	KOLA	PARKING	STOKA	LIVADA
Izvorni kôd	kola.pas kola.c kola.cpp	parking.pas parking.c parking.cpp	stoka.pas stoka.c stoka.cpp	livada.pas livada.c livada.cpp
Ulazna datoteka	kola.in	parking.in	stoka.in	livada.in
Izlazna datoteka	kola.out	parking.out	stoka.out	livada.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)	6	7	8	9
Ukupno bodova	60	70	80	90
	300			

KOLA

U sobi se nalazi M **punih** boca kole označenih brojevima od 1 do M . Ispred sobe je red u kojem se nalazi N ljudi redom označenih brojevima od 1 do N koji čekaju da popiju svoju čašu kole.

Svaka osoba uđe u sobu i popije **jedan decilitar** kole. Postoje dvije vrste ljudi: **rastrošni** tj. oni koji uvijek piju iz **najpunije** boce i **štedljivi** tj. oni koji uvijek piju iz boce u kojoj ima **najmanje** kole.

Poznate su količine kole u svim bocama **nakon** što su se svi ljudi izredali.

Napišite program koji će za svaku osobu odrediti **iz koje boce** je pila.

Ulazni podaci

U prvom retku ulazne datoteke nalaze se tri cijela broja N , M i K ; broj ljudi, broj boca i količina kole u svakoj boci na početku (u decilitrima), $1 \leq N \leq 1000$, $1 \leq M \leq 100$, $1 \leq K \leq 20$. Ljudi su označeni brojevima od 1 do N , a boce brojevima od 1 do M .

U sljedećem retku zapisan je niz znakova duljine N koji se sastoji samo od znakova 'R' i 'S'. Ako se na i -tom mjestu u nizu nalazi slovo 'R' onda je i -ta osoba rastrošna, a ako se nalazi slovo 'S' onda je ta osoba štedljiva.

U sljedećem retku nalazi se M cijelih brojeva koji nam redom za svaku bocu kažu koliko je na kraju decilitara kole ostalo u njoj. Svaki taj broj je manji ili jednak od K .

Izlazni podaci

U prvom i jedinom retku izlazne datoteke treba za svaku osobu od 1 do N ispisati broj boce iz koje je pila. Brojeve međusobno odvojiti razmakom.

Napomena: ulazni podaci će biti takvi da će rješenje, ne nužno jedinstveno, uvijek postojati.

Test primjeri

kola.in

4 3 3
SSSS
3 0 2

kola.out

2 2 2 3

kola.in

4 3 3
SSRS
3 0 2

kola.out

2 2 3 2

kola.in

5 5 2
SSRRS
1 2 0 2 0

kola.out

3 3 5 1 5

PARKING

Podzemna garaža se sastoji od N parkirališnih mjesta (**parkinga**). Parkinzi su označeni brojevima od 1 do N , a svaki je potpuno **odvojena** prostorija u koju stane **točno jedan** auto. Neki parkinzi su direktno povezani prolazima.

Ako je neki auto parkiran na nekom parkingu, onda se kroz taj parking **ne može** prolaziti. Izlaz iz garaže nalazi se kod parkinga broj 1 koji ne služi za parkiranje nego samo za prolaz automobila prema izlazu.

Između svaka dva parkinga postoji **točno jedan** put, a prilikom vožnje od nekog parkinga prema izlazu svaki parking kroz koji prođemo se nalazi **kat više** u garaži.

Kozmo K. parkirao je svoj auto na parkingu broj P i želi izaći iz garaže. Međutim, na nekima od ostalih parkinga u garaži parkirani su neki drugi auti pa da bi izašao iz garaže on mora prvo **osloboditi prolaz** između svog auta i izlaza. Na sreću, svi automobili su u leri i on ih može **gurati nizbrdo** tj. prema slobodnim parkinzima koji se nalaze na katovima niže. Jedno **pomicanje** auta definiramo kao guranje auta nizbrdo s jednog kata na kat niže.

Napišite program koji će odrediti koliki je **minimalni** broj pomicanja automobila koji Kozmo i njegovo veselo društvo moraju napraviti da bi oslobodili prolaz za svoj auto.

Kozmo neće micati svoj automobil sve dok ne oslobodi prolaz do izlaza.

Ulazni podaci

U prvom retku ulazne datoteke nalaze se cijeli brojevi N , P i K , $2 \leq N \leq 5000$, $2 \leq P \leq N$, $0 \leq K \leq N-2$. Broj K je broj automobila u garaži (ne računajući Kozmov).

U svakom od sljedećih N redaka nalaze se podaci o parkinzima. U $(i+1)$ -om retku nalaze se podaci o **svim** parkinzima **jedan kat niže** s kojima je i -ti parking direktno povezan:

$T A_1 A_2 \dots A_T$

Broj T predstavlja broj tih parkinga, a A_1, A_2, \dots, A_T su njihovi brojevi.

U sljedećem tj. zadnjem retku ulazne datoteke nalazi se K cijelih brojeva, popis parkinga na kojima su parkirani automobili (ne računajući Kozmov).

PARKING

Izlazni podaci

U prvi i jedini redak izlazne datoteke treba zapisati traženi broj pomicanja iz teksta zadatka. Ako rješenje ne postoji, ispisati 'ne postoji'.

Test primjeri

parking.in

```
6 3 2
1 4
1 6
0
1 5
2 2 3
0
4 5
```

parking.out

```
4
```

parking.in

```
6 4 2
2 5 6
0
0
0
2 4 2
1 3
6 5
```

parking.out

```
1
```

parking.in

```
8 5 3
1 3
1 7
2 2 4
2 5 6
0
0
1 8
0
2 3 7
```

parking.out

```
2
```

STOKA

Ukrcavamo životinje u vlak. One su redom označene brojevima od 1 do N i čekaju u redu jedna iza druge. Vlak se sastoji od K vagona, a svaki vagon prima maksimalno M životinja.

Životinje se ukrcavaju tako da u vagon koji punimo ukrcamo **proizvoljni** broj životinja **s početka reda**.

Poznato je i da se neki **parovi** životinja međusobno ne vole i da ako se nalaze u istom vagonu jedna od njih će pojesti ovu drugu. Ovu drugu životinju jedino može spasiti neka treća životinja (njen zaštitnik od prve životinje) ako se nalazi **u istom** vagonu.

Napišite program koji će raspodijeliti svih N životinja po vagonima tako da broj životinja koje će preživjeti put bude **što je moguće veći**.

Životinje počinju jesti jedna drugu tek nakon što se svi ukrcaju, tj. tek nakon što vlak krene.

Ulazni podaci

U prvom retku ulazne datoteke nalaze se tri cijela broja N, K i M, $1 \leq N, K \leq 1000$, $1 \leq M \leq 20$.

U drugom retku nalazi se cijeli broj D tj. broj međusobnih neprijateljstava među životinjama.

U svakom od sljedećih D redaka nalaze se tri cijela broja A, B i C. To znači da, ako se nalaze u istom vagonu, životinja broj A će pojesti životinju broj B osim ako životinja broj C nije prisutna u istom vagonu. A, B i C su međusobno različiti.

Ako se životinja broj A negdje pojavljuje kao potencijalni ubojica, onda se ta životinja nigdje drugdje ne pojavljuje kao žrtva.

Ako je životinja broj A potencijalni ubojica životinje broj B, onda postoji točno jedna životinja broj C koja je zaštitnik životinje broj B od A.

Izlazni podaci

U prvi i jedini redak izlazne datoteke treba ispisati broj preživjelih životinja iz teksta zadatka.

Test primjeri

stoka.in

5 2 3
1
5 4 1

stoka.out

4

stoka.in

8 3 3
6
4 6 2
5 6 4
7 8 6
5 3 6
7 6 5
2 3 4

stoka.out

5

stoka.in

5 2 3
2
1 2 3
1 3 2

stoka.out

5

LIVADA

Na livadi se nalaze cvjetovi.

Pčele moraju skupiti pelud sa svih cvjetova tj. posjetiti svaki cvijet. Pčelica Maja je zadužena za izradu **rasporeda** skupljanja peluda. Raspored se sastoji od popisa cvjetova za svaku pčelu.

Svaka pčela leti od cvijeta do cvijeta sa svog popisa prozvoljnim redoslijedom, a svaki cvijet može posjetiti prozvoljan broj puta. **Težina redoslijeda** obilaženja cvjetova je **najveća** udaljenost između dva uzastopna cvijeta na tom putu. Za zadani popis cvjetova, pčela će uvijek odabrati onaj redoslijed obilaženja cvjetova koji ima **minimalnu** težinu i tu minimalnu težinu zovemo **težina popisa**.

Težina rasporeda je težina najtežeg popisa u rasporedu, a pčelica Maja želi odrediti raspored s **minimalnom** težinom.

Napišite program koji će pomoći pčelici Maji.

Ulazni podaci

U prvom retku ulazne datoteke nalaze se dva cijela broja C i P, $1 \leq C \leq 2000$, $1 \leq P \leq C$, broj cvjetova i broj pčela.

U sljedećih C redaka nalaze se koordinate cvjetova, dva cijela broja X i Y, $1 \leq X, Y \leq 10000$.

Izlazni podaci

U prvi i jedini redak izlazne datoteke treba ispisati težinu traženog rasporeda iz teksta zadatka, zaokruženu na dvije decimale.

Zbog nepreciznosti aritmetike realnih brojeva, dopušteno odstupanje od točnog rješenja iznosi 0,01.

Test primjeri

livada.in

3 2
1 1
2 3
3 2

livada.out

1.41

livada.in

5 3
1 1
1 4
1 5
5 1
5 5

livada.out

3.00

livada.in

7 4
1 1
3 9
9 4
2 2
6 4
5 5
6 9

livada.out

3.00