

## zadaci

---

zadatak	utrka	labudovi	nagibni
izvorni kôd	utrka.pas utrka.c utrka.cpp	labudovi.pas labudovi.c labudovi.cpp	nagibni.pas nagibni.c nagibni.cpp
ulazni podaci	standardni ulaz		
izlazni podaci	standardni izlaz		
vremensko ograničenje (Athlon 64 3000+)	1 sekunda	2 sekunde	1 sekunda
memorijsko ograničenje (heap)	32 MB	64 MB	32 MB
memorijsko ograničenje (stack)	8 MB	16 MB	8 MB
broj bodova	50	70	80
	200		

## utrka

---

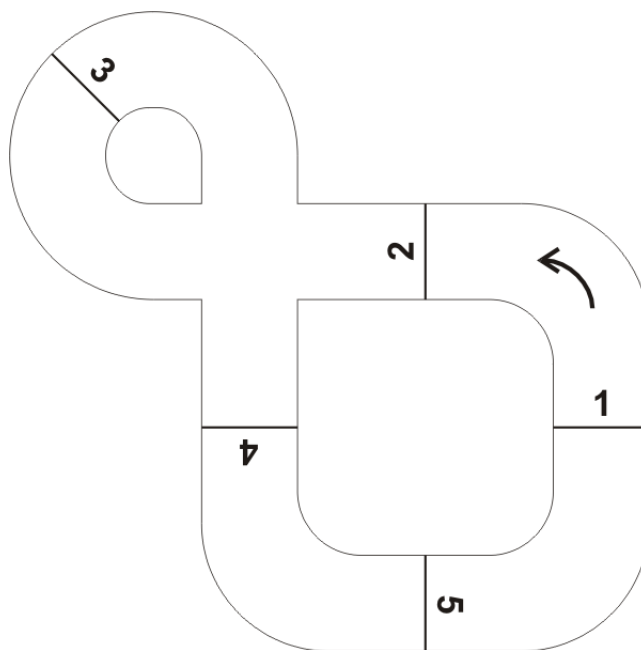
Moderne automobilske utrke postale su nezamislivo brze, a u njima sudjeluje mnogo natjecatelja. Zbog prividnog općeg kaosa koji tako nastaje, na utrci je potrebno imati posebne suce koji prate tijekom utrke, a informatički sustav na temelju dojava sudaca određuje poredak vozača u utrci.

Na kružnoj stazi na kojoj se vozi utrka nalazi se **K kontrolnih točaka** (označenih brojevima od 1 do K), a uz svaku se nalazi sudac. Svaki put kad neki vozač prođe kontrolnu točku, sudac koji se nalazi uz nju dojava centralnom računalu o kojem se vozaču radi (vozači su označeni brojevima od 1 do N). Start se nalazi **neposredno prije prve** kontrolne točke tj. vozači odmah nakon starta prolaze kroz prvu kontrolnu točku.

Na donjoj slici možete vidjeti stazu s 5 kontrolnih točaka. Vozači moraju kontrolne točke prolaziti redom kojim su označene. Ako vozač između dvije uzastopne kontrolne točke prođe kroz neke druge kontrolne točke, smatramo da ih nije ni prošao. Npr. na stazi na slici vozač može "presjeći" stazu tako da nakon kontrolne točke broj 2 skrene ulijevo. Međutim, prolasci kroz sve kontrolne točke koje bi vozač na taj način prošao (točka 4, 5, 1, 2, ...) bile bi zanemarene, sve dok vozač ne prođe kontrolnu točku broj 3, nakon čega bi njegova vožnja ponovo postala regularna, a prolasci kroz naredne kontrolne točke bi bili uzimani u obzir za konačni poredak.

Napišite program koji će izračunati **trenutni poredak** vozača, ako su nam poznate dojave sudaca uz kontrolne točke.

Ako su dva vozača regularno prošla isti broj kontrolnih točaka tj. na istom su segmentu staze koji je između dvije kontrolne točke, smatramo da je bolji onaj koji je zadnju (regularno ubrojenu) kontrolnu točku ranije prošao.



## utrka

---

### ulazni podaci

U prvom retku se nalaze tri prirodna broja  $K$ ,  $N$  i  $D$ ,  $1 \leq K \leq 100$ ,  $1 \leq N \leq 100$ ,  $1 \leq D \leq 10000$  (deset tisuća).

U svakom od sljedećih  $D$  redaka se nalaze dva prirodna broja  $X$  i  $Y$  tj. dojava da je vozač s brojem  $X$  prošao kontrolnu točku označenu brojem  $Y$ . Događaji na stazi se odvijaju redoslijedom kojim su zadani.

**Napomena:** test podaci će biti takvi da će rješenje uvijek postojati tj. svi vozači će barem jednom proći prvu kontrolnu točku.

### izlazni podaci

U prvi i jedini redak treba ispisati  $N$  prirodnih brojeva tj. redoslijed vozača u utrci nakon  $D$  dojava sudaca.

### test primjeri

#### ulaz

3 3 6  
3 1  
1 1  
2 1  
3 2  
3 3  
2 2

#### izlaz

3 2 1

#### ulaz

2 2 5  
2 1  
2 1  
1 1  
2 1  
1 2

#### izlaz

1 2

#### ulaz

4 4 22  
3 1  
2 1  
4 2  
4 3  
4 4  
4 1  
1 1  
4 2  
1 2  
1 3  
3 2  
3 3  
2 2  
2 3  
3 4  
2 4  
3 1  
1 4  
2 1  
4 3  
2 2  
3 2

#### izlaz

2 3 1 4

## labudovi

Dolaskom proljeća počela su se topiti do jučer zaleđena jezera, pa tako i naše omiljeno jezero na kojem živi **jedan par** ledom razdvojenih labudova.

Jezero je pravokutnog oblika i sastoji se od polja poredanih u R redaka i S stupaca. Neka polja mogu biti pokrivena ledom. Jezero se topi tako da se **za jedan dan** otopi svako polje leda koje je **u dodiru s vodom**. Smatramo da je polje leda u dodiru s vodom ako je to polje neposredan susjed (gore, dolje, lijevo ili desno, ali ne i dijagonalno) jednom ili više polja vode.

Na slici se nalazi jezero iz 3. test primjera i njegov izgled kroz prva tri dana:

...XXXXXX..XX.XXX	....XXXX.....XX	.....XX.....
...XXXXXXXXXX.XXX	....XXXX..X....	.....X.....
...XXXXXXXXXXXXXX..	....XXX..XXXX....	.....X.....X....
..XXXXX..XXXXXX..	...XXX....XXXX...	....X.....XX....
.XXXXXX..XXXXXX..	..XXXX....XXXX...	...XX.....XX....
XXXXXXXXX..XXXX...	..XXXX.....XX....	....X.....
..XXXXX..XXX....	....XX.....X....	.....
....XXXXX.XXX....	....XX.....X....	.....
na početku	nakon prvog dana	nakon drugog dana

Labudovi se mogu kretati samo po vodi i to u 4 dozvoljena smjera - gore, dolje, lijevo i desno, ali ne i dijagonalno.

Napišite program koji će izračunati **nakon koliko dana** će se labudovi moći susresti.

### ulazni podaci

U prvom retku se nalaze dva prirodna broja R i S,  $1 \leq R, S \leq 1500$ .

U sljedećih R redaka se nalazi opis jezera na početku: '.' (točka) označava vodu, 'X' označava polje pokriveno ledom, a 'L' polja na kojima se nalaze labudovi (ta polja nisu pokrivena ledom).

### izlazni podaci

U prvi i jedini redak treba ispisati traženi broj dana iz teksta zadatka.

### test primjeri

<b>ulaz</b>	<b>ulaz</b>	<b>ulaz</b>
10 2	4 11	8 17
.L	..XXX...X..	...XXXXXX..XX.XXX
..	.X.XXX...L.	...XXXXXXXXXX.XXX
XX	....XXX..X.	...XXXXXXXXXXXXXX..
XX	X.L..XXX...	..XXXXX.LXXXXXX..
XX		.XXXXXX..XXXXXX..
XX	<b>izlaz</b>	XXXXXXXXX..XXXX...
XX	2	..XXXXX..XXX....
XX		....XXXXX.XXXL...
..		<b>izlaz</b>
.L		2
<b>izlaz</b>		
3		

## nagibni

---

Željeznička mreža u jednoj zemlji sastoji se od određenog broja pojedinačnih pruga. Svaka od njih povezuje neka dva grada, poznata nam je duljina svake pruge, a između svaka dva grada postoji **jedinstveni** put željeznicom.

Nedavno je vlada odlučila da se stari i spori vlakovi prodaju siromašnim zemljama, a da se od ludih Japanaca kupe najmoderniji i najbrži nagibni vlakovi.

Linija na kojoj vozi neki nagibni vlak se sastoji od niza gradova (barem dva) tako da između svaka dva susjedna grada na liniji postoji pruga, a **duljinu linije** definiramo kao **zbroj duljina** svih pruga na toj liniji. Dvije linije ne smiju prolaziti kroz isti grad jer su vlakovi vrlo brzi pa postoji velika opasnost od sudara.

Napišite program koji će odrediti neki raspored linija nagibnih vlakova, takav da kroz svaki grad prolazi **točno jedna** linija, a da **zbroj duljina svih linija** bude **što je moguće veći**.

### ulazni podaci

U prvom retku se nalazi prirodni broj  $N$ ,  $1 \leq N \leq 2000$  (dvije tisuće), broj gradova. Gradovi su označeni brojevima od 1 do  $N$ .

U sljedećih  $N-1$  redaka se nalaze podaci o prugama. U svakom od tih redaka se nalaze tri prirodna broja  $A$ ,  $B$  i  $C$  - to znači da između grada  $A$  i grada  $B$  postoji pruga duljine  $C$ ,  $1 \leq C \leq 1000000$  (milijun).

**Napomena:** test podaci će biti takvi da će rješenje uvijek postojati.

### izlazni podaci

U prvi i jedini redak treba ispisati traženi maksimalni zbroj iz teksta zadatka.

### test primjeri

ulaz	ulaz	ulaz
5	6	9
1 2 10	1 2 1	1 2 1
1 3 10	1 3 3	2 4 2
1 4 10	1 4 7	3 4 1
4 5 1	2 5 4	4 5 5
	2 6 6	6 5 1
<b>izlaz</b>	<b>izlaz</b>	5 7 2
21		7 8 1
		8 9 2
		<b>izlaz</b>
		11