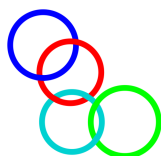


Prstenovi za padobran

Rana i prilično složena verzija onog što mi sada zovemo padobranom opisana je u Leonardovom Codex Atlanticus-u (oko 1485.). Leonardov padobran sastojao se od zatvorene lanene tkanine koju je držala otvorenom drvena struktura oblika piramide.

Povezani prstenovi

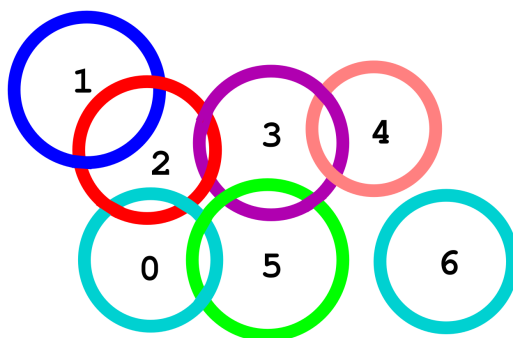
Padobranac Adrian Nikolas je testirao Leonardov dizajn više od 500 godina nakon toga. Radi toga, moderna laka struktura vezala je Leonardov padobran za ljudsko telo. Mi želimo da koristimo povezane prstenove koji takođe obezbeđuju kuke za zatvorenu lanenu tkaninu. Svaki prsten je mala kopča sačinjena od fleksibilnog i jakog materijala. Prstenovi se mogu lako povezati, jer se svaki prsten može otvoriti i zatvoriti. Posebna konfiguracija povezanih prstenova je *lanac*: niz od jednog ili više povezanih prstenova u kom je svaki prsten povezan sa neka druga dva, osim prvog i poslednjeg koji su povezani sa samo jednim drugim prstenom, kao što je ilustrovano ispod. Specijalno, samo jedan prsten je takođe lanac.



Ostale konfiguracije su očigledno moguće jer prsten može biti povezan sa tri ili više drugih prstenova. Kažemo da je prsten *kritičan* kada nakon njegovog otvaranja i uklanjanja svi preostali prstenovi formiraju skup disjunktnih lanaca (ili nema preostalih prstenova). Drugim rečima ne može da ostane ništa osim lanaca.

Primer

Posmatrajmo sedam prstenova na sledećoj slici, obeleženih od 0 do 6. Postoje tačno dva kritična prstenova. Jedan kritičan prsten je 2: nakon njegovog uklanjanja, ostali prstenovi formiraju lance [1], [0, 5, 3, 4] i [6]. Drugi kritičan prsten je 3: nakon njegovog uklanjanja, preostali prstenovi formiraju lance [1, 2, 0, 5], [4] i [6]. Ako uklonimo bilo koji drugi prsten, ne dobijamo skup disjunktnih lanaca. Na primer, nakon uklanjanja prstena 5: iako imamo da je [6] lanac, povezani prstenovi 0, 1, 2, 3 i 4 ne formiraju lanac.



Postavka problema

Vaš zadatak je da prebrojite broj kritičnih prstenova u datoj konfiguraciji koja će biti prosledjena (poslata) programu.

Na početku, postoji određen broj disjunktih prstenova. Nakon toga, prstenovi se povezuju međusobno. U bilo kojem trenutku, od Vas može biti zatraženo da vratite broj kritičnih prstenova u trenutnoj konfiguraciji. Preciznije, treba da implementirate sledeće tri funkcije.

- `Init(N)` — poziva se tačno jednom kako bi se saopštilo da postoji N nepovezanih prstenova numerisanih od 0 do $N - 1$ (uključujući i njih) u inicijalnoj konfiguraciji.
- `Link(A, B)` — dva prstena numerisana sa A i B postaju povezana. Garantovano je da su A i B različiti i da nisu već direktno povezani; pored ovoga, ne postoje dodatni uslovi za A i B , konkretno nema uslova koji proističu iz fizičkih ograničenja. Očigledno, `Link(A, B)` i `Link(B, A)` su ekvivalentni.
- `CountCritical()` — vraća broj kritičnih prstenova u trenutnoj konfiguraciji povezanih prstenova.

Primer

Posmatrajmo našu sliku sa $N = 7$ prstenova i pretpostavimo da na početku prstenovi nisu povezani. Dalje je prikazan mogući niz poziva funkcija gde se nakon poslednjeg poziva dobije konfiguracija prikazana na toj slici.

Poziv	Vraća
Init(7)	
CountCritical()	7
Link(1, 2)	
CountCritical()	7
Link(0, 5)	
CountCritical()	7
Link(2, 0)	
CountCritical()	7
Link(3, 2)	
CountCritical()	4
Link(3, 5)	
CountCritical()	3
Link(4, 3)	
CountCritical()	2

Podzadatak 1 [20 bodova]

- $N \leq 5\,000$.
- Funkcija `CountCritical` se poziva samo jednom, posle svih ostalih poziva; funkcija `Link` se poziva najviše 5 000 puta.

Podzadatak 2 [17 bodova]

- $N \leq 1\,000\,000$.
- Funkcija `CountCritical` se poziva samo jednom, posle svih ostalih poziva; funkcija `Link` se poziva najviše 1 000 000 puta.

Podzadatak 3 [18 bodova]

- $N \leq 20\,000$.
- Funkcija `CountCritical` se poziva najviše 100 puta; funkcija `Link` se poziva najviše 10 000 puta.

Podzadatak 4 [14 bodova]

- $N \leq 100\,000$.
- Funkcije `CountCritical` i `Link` se pozivaju ukupno najviše 100 000 puta.

Podzadatak 5 [31 bodova]

- $N \leq 1\,000\,000$.
- Funkcije `CountCritical` i `Link` se pozivaju ukupno najviše 1 000 000 puta.

Detalji o implementaciji

Treba da predate samo jedan fajl sa imenom `rings.c`, `rings.cpp` ili `rings.pas`. U njoj se nalaze implementacije potprograma koji su prethodno opisani sa sledećim prototipovima.

C/C++ programi

```
void Init(int N);  
void Link(int A, int B);  
int CountCritical();
```

Pascal programi

```
procedure Init(N : LongInt);  
procedure Link(A, B : LongInt);  
function CountCritical() : LongInt;
```

Ovi funkcije se moraju ponašati kao što je opisano gore. Naravno, vi ste slobodni da koristite i druge pomoćne potprograme za testiranja. Vaše rešenje ne sme ni na koji način koristiti standardni ulaz i izlaz niti bilo koju drugu datoteku.

Probni tester

Ulazni podaci za probni tester imaju sledeći format:

- red 1: N, L ;
- redovi 2, ..., $L + 1$:
 - -1 za poziv `CountCritical`;
 - A, B parametri za `Link`.

Probni tester će ispisati sve rezultate poziva `CountCritical`.