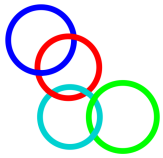
Prstenovi za padobran

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

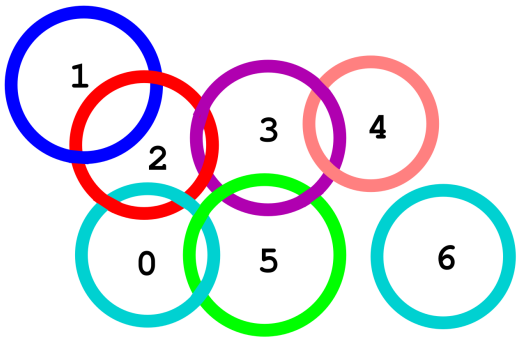
Padobranac Adrian Nikolas testirao je Leonardov dizajn više od 500 godina kasnije. Za ovo, moderna laka struktura vezala je Leonardov padobran za ljudsko tijelo. Mi želimo da koristimo povezane prstenove koji takođe obezbjeđuju kuke za zatvorenu lanenu tkaninu. Svaki prsten je mala kopča sačinjena od 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 posljednjeg koji su povezani sa samo jednim drugim prstenom, kao što je ilustrovano ispod.



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).

Primjer

Posmatrajmo sedam prstenova na sljedećoj slici, obilježenih od 0 do 6. Postoje dva kritična prstena. 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 primjer, nakon uklanjanja prstena 5: iako imamo da je [6] lanac, povezani prstenovi 0, 1, 2, 3 i 4 ne formiraju lanac.



Postavka

Vaš zadatak je da prebrojite broj kritičnih prstenova u datoj konfiguraciji koja će biti komunicirana programu.

Na početku, postoji određen broj disjunktnih prstenova. Nakon toga, prstenovi se povezuju međusobno. U bilo kojem trenutku, od Vas može biti zatraženo da vratite broj kritičnih prstenova u trenutnoj konfiguraciji. Specifično, trebate implementirati tri rutine.

- `Init`(N) — poziva se tačno jednom kako bi se komuniciralo da postoji N nepovezanih prstenova numerisanih od 0 do N - 1 (inkluzivno) u inicijalnoj konfiguraciji.
- `Link`(A, B) — dva prstena numerisana sa A i B bivaju povezana. Garantovano je da su A i B različiti i da nisu već direktno povezana; 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 prestenova.

Primjer

Razmatrajmo sliku datu ranije za n = 7 prstenova i pretpostavimo da na početku prstenovi nisu povezani. Dalje je prikazan mogući slijed poziva funkcija gdje se nakon posljednjeg poziva dobije konfiguracija prikazana na spomenutoj 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 nakon svih poziva drugih funkcija; funkcija `Link` može biti pozvana najviše 5 000 puta.

Podzadatak 2 [17 bodova]

- $N \leq 1\,000\,000$.
- Funkcija `CountCritical` se poziva samo jednom nakon svih poziva drugih funkcija; funkcija `Link` može biti pozvana najviše 1 000 000 puta.

Podzadatak 3 [18 bodova]

- $N \leq 20\,000$.
- Funkcija `CountCritical` može biti pozvana najviše 100 puta; funkcija `Link` može biti pozvana najviše 10 000 puta.

Podzadatak 4 [14 bodova]

- $N \leq 100\,000$.
- Funkcije `CountCritical` i `Link` mogu biti ukupno pozvane najviše 100 000 puta.

Podzadatak 5 [31 bodova]

- $N \leq 1\,000\,000$.
- Funkcije `CountCritical` i `Link` mogu biti ukupno pozvane najviše 1 000 000 puta.

Implementacijski detalji

Trebate poslati tačno jednu datoteku pod nazivom `rings.c`, `rings.cpp` ili `rings.pas`. U ovoj datoteci, trebate implementirati sve ranije opisane potprograme prema sljedećim opisima.

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 potprogrami se trebaju ponašati onako kako je ranije opisano. Naravno, smijete koristiti i druge, pomoćne potprograme. Vaše rješenje ne smije ni na koji način koristiti standradni ulaz i izlaz niti bilo koju drugu datoteku.

Probni test

Ulazni podaci za probni test su u sljedećem formatu:

- linija 1: N, L;
- linije 2, …, L + 1:
  - -1 za poziv `CountCritical`;
  - A, B parametri za `Link`.

Probni test će ispisati sve rezultate poziva `CountCritical`.