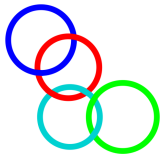
Obroči padala

Zgodnja in precej dovršena različica tega, kar danes imenujemo padalo, je Leonardo opisal v *Codex atlanticus* (ok. 1485). Leonardovo padalo sestavlja lesena piramidna konstrukcija, z zgornje strani obdana z zašitimi lanenimi plahtami.

Povezani obroči

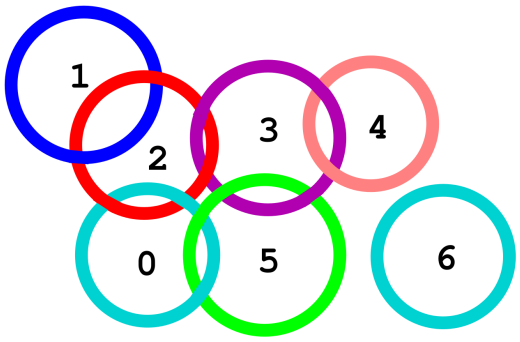
Padalec Adrian Nicholas je testiral Leonardov načrt več kot 500 let kasneje. Za to je uporabil lahko moderno ogrodje, ki je Leonardovo padalo privezalo na človeško telo. Mi želimo uporabiti povezane obroče, ki imajo tudi kavlje za pripenjanje lanene plahte. Obroči so narejeni iz upogljive in zelo močne snovi ter se zelo preprosto povezujejo, saj je vsak obroč moč odpreti in zapreti (kot karabin). Posebno konfiguracijo povezanih obročev imenujemo *veriga*. Veriga je niz obročev, kjer je vsak obroč povezan z (največ) dvema sosednjima obročema, kakor je prikazano na spodnji sliki. Tak niz mora imeti začetek in konec (obroč je povezan z največ enim drugim obročem). Po tej definiciji je en obroč ravno tako veriga.



Seveda so možne tudi drugačne konfiguracije, saj je posamezen obroč lahko povezan tudi s tremi ali več drugimi obroči. Za obroč pravimo, da je *kritičen*, če po odpiranju in njegovi odstranitvi, ostali obroči tvorijo množico verig (ali pa ne ostane več noben obroč). Z drugimi besedami, od konfiguracije lahko ostanejo le še verige.

Primer

Vzemi v obzir 7 obročev na spodnji sliki, oštevilčenih od 0 do 6. Obstajata dva kritična obroča. Eden izmed kritičnih obročev je 2: po njegovi odstranitvi, preostali obroči tvorijo verige [1], [0, 5, 3, 4] in [6]. Drugi kritični obroč je 3: po njegovi odstranitvi, preostali obroči tvorijo verige [1, 2, 0, 5], [4] in [6]. Če odstanimo katerikoli drug obroč, ne dobimo disjunktne množice verig. Na primer če odstranimo obroč 5, dobimo verigo [6], vendar povezani obroči 0, 1, 2, 3 in 4 ne tvorijo verige.



Naloga

Tvoja naloga je prešteti število kritičnih obročev v dani konfiguraciji, ki bo predana tvojemu programu.

Na začetku je nekaj nepovezanih obročev. Po tem se obroči povezujejo. Tvoj program je lahko v poljubnem trenutku vprašan po številu kritičnih obročev trenutne konfiguracije. Natančneje, implementirati moraš tri funkcije.

- `Init`(N) — pokliče se natanko enkrat na začetku, da pove, da je v začetni konfiguraciji N nepovezanih obročev, oštevilčenih z 0 do N - 1 (vključujoče).
- `Link`(A, B) — obroča, oštevilčena z A in B se povežeta. Zagotovljeno je, da sta A in B različna in še nista neposredno povezana; poleg tega ni za A in B nobenih dodatnih pogojev, posebej ni nobenih pogojev zaradi morebitnih fizičnih omejitev. Očitno sta klica `Link`(A, B) in `Link`(B, A) ekvivalentna.
- `CountCritical`() — vrni število kritičnih obročev za trenutno konfiguracijo povezanih obročev.

Primer

Oglej si zgornjo sliko z N = 7 obroči in predpostavi, da so na začetku nepovezani. Prikazano je možno zaporedje klicev, pri čemer po zadnjem klicu dobimo stanje, kot je na sliki.

Klic	Vrne
`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

1. podnaloga [20 točk]

- $N \leq 5\,000$.
- Funkcija `CountCritical` bo poklicana natanko enkrat, po vseh drugih klicih; funkcija `Link` bo poklicana največ 5 000 krat.

2. podnaloga [17 točk]

- $N \leq 1\,000\,000$.
- Funkcija `CountCritical` bo poklicana natanko enkrat, po vseh drugih klicih; funkcija `Link` bo poklicana največ 1 000 000 krat.

3. podnaloga [18 točk]

- $N \leq 20\,000$.
- Funkcija `CountCritical` bo poklicana največ 100 krat; funkcija `Link` bo poklicana največ 10 000 krat.

4. podnaloga [14 točk]

- $N \leq 100\,000$.
- Funkciji `CountCritical` in `Link` bosta vsega skupaj poklicani največ 100 000 krat.

5. podnaloga [31 točk]

- $N \leq 1\,000\,000$.
- Funkciji `CountCritical` in `Link` bosta vsega skupaj poklicani največ 1 000 000 krat.

Podrobnosti implementacije

Oddati moraš natanko eno datoteko, z imenom `rings.c`, `rings.cpp` ali `rings.pas`. V tej datoteki naj se nahajajo implementacije zgoraj opisanih funkcij s sledečimi podpisi.

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;
```

Te funkcije se morajo vesti, kot je opisano zgoraj. Seveda imaš proste roke pri implementaciji drugih čudovitih funkcij za lastno rabo. Tvoje oddaje nikakor ne smejo uporabljati standardnega vhoda/izhoda ali kakršnekoli druge datoteke.

Primer ocenjevalca

Ocenjevalec vhodne podatke prebere v tem formatu:

- vrstica 1: N, L;
- vrstice 2, …, L + 1:
  - -1 za klic `CountCritical`;
  - A, B argumenta `Link`.

Ocenjevalec bo izpisal vse rezultate funkcije `CountCritical`.