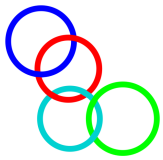
Падобрански прстени

Една рана и прилично софистицирана верзија на она што денес го нарекуваме падобран е опишана во Леонардовата *Codex Atlanticus* (год. 1485). Падобранот на Леонардо се состоел од ленено платно кое што било држено отворено од страна на една дрвена структура во форма на пирамида.

Поврзани прстени

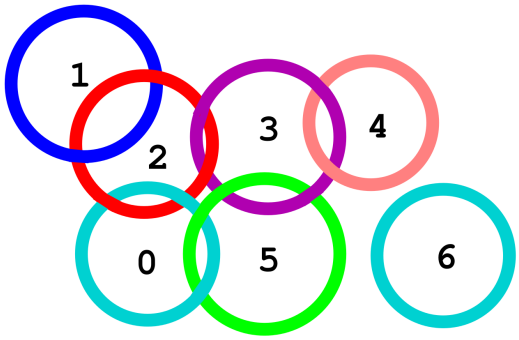
Падобранецот Кико го тестирал дизајнот на Леонардо нешто повеќе од 500 години подоцна. Притоа, една модерна и лесна структура го врзувала Леонардовиот падобран за човечкото тело. Сакаме да користиме поврзани прстени, кои исто така обезбедуваат куки за лененото платно. Секој прстен е направен од цврст и флексибилен материјал. Прстените лесно може да се поврзуваат, бидејќи секој прстен може да се отвори и повторно да се затвори. Специјална конфигурација на поврзани прстени е *синџир*. *Синџир* е секвенца од прстени во која секој прстен е поврзан само со неговите (најмногу два) соседи, како што е илустрирано подолу. Оваа секвенца мора да има почеток и крај (тоа се прстени кои што се поврзани најмногу со по еден друг прстен). Специфично, единечен прстен е исто така синџир.



Јасно е дека се возможни и други конфигурации, со оглед на тоа што еден прстен може да биде поврзан со три или повеќе други прстени. Ќе велиме дека еден прстен е *критичен* ако по неговото отворање и отстранување, сите преостанати прстени формираат множество од синџири (или повеќе нема преостанати прстени). Со други зборови, не може да остане ништо друго освен синџири.

Пример

Да ги разгледаме седумте прстени на следната слика, нумерирани со целите броеви од 0 до 6. Постојат два критични прстена. Еден критичен прстен е 2: по неговото отстранување, преостанатите прстени ги формираат синџирите [1], [0, 5, 3, 4] и [6]. Другиот критичен прстен е 3: по неговото отстранување преостанатите прстени ги формираат синџирите [1, 2, 0, 5], [4] и [6]. Ако отстраниме кој било од другите прстени, нема да добиеме множество од дисјунктни синџири. На пример, по отстранување на прстенот 5: иако имаме дека [6] е синџир, поврзаните прстени 0, 1, 2, 3 и 4 не формираат синџир.



Задача

Ваша задача е да пресметате колку критични прстени постојат во дадена конфигурација која ќе биде зададена како влез на вашата програма.

На почетокот, постојат одреден број на дисјунктни прстени. После тоа, прстените се поврзуваат. Во даден произволен момент, од вас може да се побара да го вратите бројот на критични прстени во моменталната конфигурација. Конкретно, треба да имплементирате три рутини (функции).

- `Init`(N) — оваа функција се повикува точно еднаш - на почетокот, за да се наведе дека постојат N дисјунктни прстени нумерирани со броевите од 0 до N - 1 во почетната конфигурација.
- `Link`(A, B) — се поврзуваат двата прстени нумерирани со A и B. Се гарантира дека A и B се различни и дека не се претходно поврзани директно; покрај ова, нема други услови во врска со A и B, конкретно - нема услови што потекнуваат од физички ограничувања. Јасно, `Link`(A, B) и `Link`(B, A) се еквивалентни.
- `CountCritical`() — врати го бројот на критични прстени за моменталната конфигурација на поврзани прстени.

Пример

Разгледајте ја сликата со N = 7 прстени и претпоставете дека на почетокот истите не се поврзани. Даден е приказ на можна секвенца од повици, каде што по последниот повик ја добиваме ситуацијата прикажана на сликата.

Повик	Враќа
`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

Подзадача 1 [20 поени]

- $N \leq 5\,000$.
- Функцијата `CountCritical` се повикува само еднаш, по сите други повици; функцијата `Link` се повикува најмногу 5 000 пати.

Подзадача 2 [17 поени]

- $N \leq 1\,000\,000$.
- Функцијата `CountCritical` се повикува само еднаш, по сите други повици; функцијата `Link` се повикува најмногу 1 000 000 пати.

Подзадача 3 [18 поени]

- $N \leq 20\,000$.
- Функцијата `CountCritical` се повикува најмногу 100 пати; функцијата `Link` се повикува најмногу 10 000 пати.

Подзадача 4 [14 поени]

- $N \leq 100\,000$.
- Функциите `CountCritical` и `Link` се повикуваат, вкупно, најмногу 100 000 пати.

Подзадача 5 [31 поен]

- $N \leq 1\,000\,000$.

- Функциите `CountCritical` и `Link` се повикуваат, вкупно, најмногу 1 000 000 пати.

Имплементациски детали

Треба да предадете точно еден документ, со име `rings.c`, `rings.cpp` или `rings.pas`. Овој документ ги содржи имплементациите на потпрограмите опишани погоре користејќи ги следниве потписи.

C/C++ програми

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

Pascal програми

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

Овие потпрограми мора да се однесуваат како што е опишано погоре. Се разбира, дозволено е да имплементирате и други потпрограми за нивна внатрешна употреба. Вашите предадени решенија не смеат да имаат никаков тип на интеракција ниту со стандардниот влез/излез, ниту пак со која било друга датотека.

Пример-оценувач

Пример-оценувачот чита влез во следниот формат:

- линија 1: N, L;
- линии 2, …, L + 1:
  - -1 за да се повика `CountCritical`;
  - A, B параметри на `Link`.

Пример-оценувачот ќе ги испечати сите резултати од функцијата `CountCritical`.