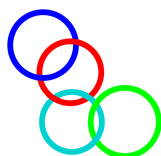


Parachute ringen

Leonardo heeft omstreeks 1485 een beschrijving gegeven van het eerste ontwerp voor een parachute. Die bestond uit een piramidevormige houten raamwerk bespannen met een linnen doek.

Geschakelde ringen

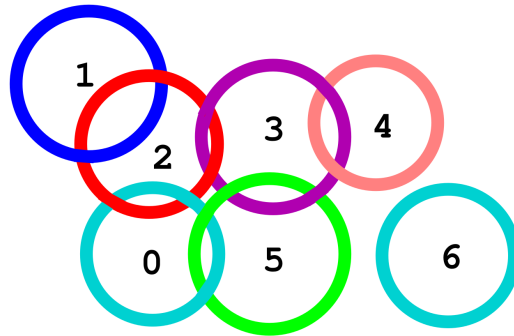
Adrian Nicholas heeft, meer dan 500 later, Leonardo's versie uitgetoetst. Daartoe werd met behulp van een moderne lichtgewicht constructie zo'n parachute vastgemaakt aan een mens. We willen gebruik maken van geschakelde ringen die tevens van haken voorzien zijn voor het bevestigen van het zeil. Elke ring is gemaakt van flexibel en sterk materiaal. De ringen kunnen makkelijk gesloten en geopend worden. Als je de ringen schakelt kan je er een *chain* mee maken: dat is een reeks van één of meer geschakelde ringen waarbij iedere ring aan zijn (maximaal twee) buuren geschakeld is. De chain moet een begin en een einde hebben (dat wil zeggen dat deze maximaal met één andere ring geschakeld moet zijn). In het bijzonder vormt één ring ook een chain.



Ook andere situaties zijn mogelijk omdat een ring ook geschakeld kan worden met drie of meer ringen. We zeggen dat een ring *kritiek* is als na het openen en het verwijderen ervan, alle overgebleven ringen een verzameling vormen van losgemaakte chains (of dat er geen enkele ring meer over is). Met andere woorden er kunnen alleen maar chains overblijven.

Voorbeeld

Beschouw de 7 ringen in de volgende figuur, genummerd van 0 tot en met 6. Er zijn twee kritische ringen. De eerste kritische ring is nummer 2: nadat hij is weggehaald, vormen de overige ringen de chains [1], [0, 5, 3, 4] en [6]. De andere kritische ring is nummer 3: nadat deze is weggehaald vormen de volgende ringen de chains [1, 2, 0, 5], [4] en [6]. Als we een andere ring weghalen, krijgen we niet een verzameling van losgemaakte chains. Bijvoorbeeld: na het weghalen van ring 5 kan je zeggen dat [6] een chain is, maar de geschakelde ringen 0, 1, 2, 3 en 4 vormen geen chain.



Opdracht

Je programma moet het aantal kritische ringen tellen in verkregen situatie die ontstaan is doordat de grader met je programma erover heeft gecommuniceerd.

In het begin is er een zeker aantal losgemaakte ringen. Na een tijdje zijn alle ringen geschakeld. Op ieder moment kan je gevraagd worden het aantal kritische ringen te retourneren in een bepaalde situatie. Je moet drie functies implementeren.

- `Init(N)` — deze moet precies eenmaal aangeroepen worden om te weten te komen dat er N losgemaakte ringen, genummerd van 0 tot en met $N - 1$ in de oorspronkelijke situatie.
- `Link(A, B)` — de twee ringen A en B worden geschakeld aan elkaar. A en B zijn verschillende ringen en zijn niet direct aan elkaar geschakeld. Er zijn geen aanvullende voorwaarden voor A en B , in het bijzonder zijn er geen beperkingen met het in het echt schakelen van de ringen. Het moet duidelijk zijn dat `Link(A, B)` en `Link(B, A)` hetzelfde doen.
- `CountCritical()` — retourneert het aantal kritische ringen voor de huidige situatie van geschakelde ringen.

Voorbeeld

Bekijk de figuur met $N = 7$ ringen en veronderstel dat deze oorspronkelijk niet geschakeld zijn. We laten een mogelijke reeks van aanroepen zien, waarna we de situatie hebben gekregen die zichtbaar is in het plaatje.

Aanroep	Resultaat
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

Subtask 1 [20 punten]

- $N \leq 5\,000$.
- De functie `CountCritical` wordt eenmaal aangeroepen en als laatste; de functie `Link` wordt maximaal 5 000 keer aangeroepen.

Subtask 2 [17 punten]

- $N \leq 1\,000\,000$.
- De functie `CountCritical` wordt eenmaal aangeroepen en als laatste; de functie `Link` wordt maximaal 1 000 000 keer aangeroepen.

Subtask 3 [18 punten]

- $N \leq 20\,000$.
- De functie `CountCritical` wordt maximaal 100 keer aangeroepen; de functie `Link` wordt maximaal 10 000 keer aangeroepen.

Subtask 4 [14 punten]

- $N \leq 100\,000$.
- De functies `CountCritical` en `Link` worden, samen, maximaal 100 000 keer aangeroepen.

Subtask 5 [31 punten]

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

- De functies `CountCritical` en `Link` worden, samen, maximaal 1 000 000 keer aangeroepen.

Implementation details

Je moet precies één programmabestand inzenden, genaamd `rings.c`, `rings.cpp` of `rings.pas`. Dit bestand moet de deelprogramma's hierboven implementeren met de volgende kenmerken.

C/C++ programma's

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

Pascal programma's

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

Deze deelprogramma's moeten werken zoals hierboven beschreven. Natuurlijk staat het je vrij andere deelprogramma's te maken voor intern gebruik. Je inzendingen mogen op geen enkele manier ingrijpen op standard input en standard output, of met andere bestanden.

Voorbeeld grader

De voorbeeld grader leest de invoer in het volgende format:

- regel 1: `N, L`;
- regels 2, ..., `L + 1`:
 - `-1` om `CountCritical` aan te roepen;
 - `A, B` om de parameters van `Link` aan te duiden.

De voorbeeld grader zal alle resultaten van `CountCritical` afdrukken.