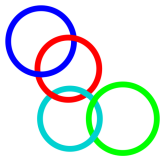
Fallschirmringe

Eine frühe und ziemlich hochentwickelte Version von dem, was wir jetzt einen Fallschirm nennen ist in Leonardo's *Codex Atlanticus* (ca. 1485) beschrieben. Leonardo's Fallschirm bestand aus beschichteten Leinentüchern, die durch eine pyramidenartige Holzstruktur offengehalten werden.

Verbundene Ringe

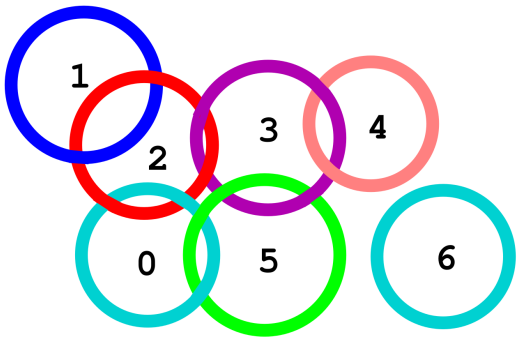
Der Fallschirmspringer Adrian Nicholas testete Leonardo's Entwurf mehr als 500 Jahre später. Für diesen Zweck wurde Leonardo's Fallschirm mit einer modernen leichten Vorrichtung an den menschlichen Körper befestigt. Wir wollen verbundene Ringe verwenden, die auch Haken für die beschichteten Tücher haben. Jeder Ring besteht aus flexiblem und festem Material. Ringe können einfach verbunden werden, da jeder Ring geöffnet und wieder geschlossen werden kann. Folgende spezielle Konfiguration von verbundenen Ringen wird *Kette* genannt. Eine Kette ist eine Folge von Ringen, in der jeder Ring nur mit seinen (höchstens zwei) Nachbarn verbunden ist, wie man der Abbildung unten entnehmen kann. Diese Folge muss einen Beginn und ein Ende haben (also Ringe, die jeweils höchstens mit einem anderen Ring verbunden sind). Im Speziellen ist ein einzelner Ring auch eine Kette.



Andere Konfigurationen sind durchaus möglich, da ein Ring mit drei oder mehreren anderen Ringen verbunden werden kann. Wir bezeichnen einen Ring als *kritisch*, wenn nach dem Öffnen und Entfernen des Ringes alle verbleibenden Ringe eine Menge von Ketten bilden (oder es bleiben keine anderen Ringe über). Mit anderen Worten: Es bleiben nur Ketten übrig.

Beispiel

Betrachte die 7 Ringe, die von 0 bis 6 nummeriert sind, in der nächsten Figur. Es gibt zwei kritische Ringe. Ein kritischer Ring ist 2: Nach seiner Entfernung bilden die verbleibenden Ringe die Ketten [1], [0, 5, 3, 4] und [6]. Der andere kritische Ring ist 3: Nach seiner Entfernung bilden die verbleibenden Ringe die Ketten [1, 2, 0, 5], [4] und [6]. Wenn wir einen anderen Ring entfernen, erhalten wir keine Menge von nicht zusammenhängenden Ketten. Zum Beispiel nach dem Entfernen vom Ring 5: Obwohl [6] eine Kette ist, bilden die verbundenen Ringe 0, 1, 2, 3 und 4 keine Kette.



Problemstellung

Deine Aufgabe ist es, die Anzahl der kritischen Ringe einer gegebenen Konfiguration zu ermitteln, die deinem Programm übermittelt wird.

Am Anfang gibt es eine bestimme Anzahl von nicht zusammenhängenden Ringen. Danach werden Ringe miteinander verbunden. Jederzeit kannst du nach der Anzahl der kritischen Ringe der aktuellen Konfiguration gefragt werden. Im Speziellen musst du drei Routinen implementieren.

- `Init`(N) — Wird genau einmal zu Beginn aufgerufen um mitzuteilen, dass es N nicht zusammenhängende Ringe in der Anfangskonfiguration gibt. Diese sind von 0 bis N-1 (inklusive) nummeriert.
- `Link`(A, B) — Die zwei Ringe A und B werden miteinander verbunden. Es ist garantiert, dass A und B verschieden sind und momentan noch nicht direkt verbunden sind; abgesehen davon gibt es keine weitere Bedingungen für A und B. Im Speziellen treten keine mechanischen Einschränkungen auf. Logischerweise sind `Link`(A, B) und `Link`(B, A) gleichwertig.
- `CountCritical`() — Soll die Anzahl der kritischen Ringe für die aktuelle Konfiguration der verbundenen Ringe zurückgeben.

Beispiel

Betrachte unsere Figur mit N = 7 Ringen und nimm an, dass alle zu Beginn nicht verbunden sind. Wir zeigen eine mögliche Folge von Aufrufen, wo wir nach dem letzten Aufruf die in der Figur abgebildete Situation erhalten.

Aufruf	Rückgaben
`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 Punkte]

- $N \leq 5\,000$.
- Die Funktion `CountCritical` wird nur einmal nach allen anderen Aufrufen aufgerufen; Die Funktion `Link` wird höchstens 5 000 mal aufgerufen.

Subtask 2 [17 Punkte]

- $N \leq 1\,000\,000$.
- Die Funktion `CountCritical` wird nur einmal nach allen anderen Aufrufen aufgerufen; Die Funktion `Link` wird höchstens 1 000 000 mal aufgerufen.

Subtask 3 [18 Punkte]

- $N \leq 20\,000$.
- Die Funktion `CountCritical` wird höchstens 100 mal aufgerufen; Die Funktion `Link` wird höchstens 10 000 mal aufgerufen.

Subtask 4 [14 Punkte]

- $N \leq 100\,000$.
- Die Funktionen `CountCritical` und `Link` werden insgesamt höchstens 100 000 mal aufgerufen.

Subtask 5 [31 Punkte]

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

- Die Funktionen `CountCritical` und `Link` werden insgesamt höchstens 1 000 000 mal aufgerufen.

Implementationsdetails

Du musst genau eine Datei names `rings.c`, `rings.cpp` oder `rings.pas` einreichen. In dieser Datei sind die oben beschriebenen Unterprogramme mit der folgenden Signatur zu implementieren.

C/C++ Programme

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

Pascal Programme

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

Diese Unterprogramme müssen sich wie oben verhalten. Natürlich kannst du andere Unterprogramme für interne Verwendung implementieren. Dein Programm darf weder Standard-Input oder -Output noch eine andere Datei benützen.

Beispiel-Grader

Der Beispiel-Grader liest die Eingabe im folgenden Format:

- Zeile 1: N, L;
- Zeile 2, …, L + 1:
  - -1 um `CountCritical` aufzurufen;
  - A, B Parameter zu `Link`.

Der Beispiel-Grader gibt alle Ergebnisse von `CountCritical` aus.