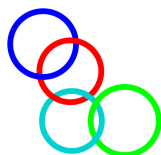


降落傘圓環

在李安納度的文獻 *Codex Atlanticus* (ca. 1485) 中描述了一種早期而頗精密的降落傘。李安納度的降落傘是一個由布料縫製而成的金字塔型木頭結構。

鏈接的圓環

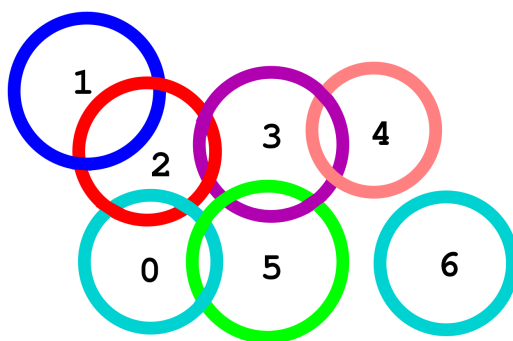
跳傘員 Adrain Nicholas 在超過 500 年後測試了李安納度的設計。在這個測試中，一個現代的輕量結構將李安納度的降落傘繫在人體上。我們要使用鏈接的圓環，這些圓環也為縫製的布料提供鉤子。每個圓環是由富彈性與紮實的物料製成的鉤環。因每一個圓環可以打開或關閉，所以圓環可以很輕易地鏈接在一起。鏈接的圓環構成一種特殊的型態叫做鍊(chain)。所謂的鍊指的是一序列的圓環，其中每個圓環只連接到〔最多兩個〕其鄰接的圓環，如下圖所示。此序列必須有一個開始及一個結尾〔此等圓環只連接到最多一個其他的圓環〕。特別的是，單一的環也是鍊。



因為一個圓環可以鏈接到三個或更多的圓環，明顯地鍊可以有其他型態。如果我們將一個圓環打開並移去這個圓環，而其他剩下的圓環會形成一組鍊〔或沒有任何的圓環留下〕，則我們說這個圓環為 **關鍵的**。換句話說，剩下的只有鍊。

範例

參考下圖中的 7 個圓環，其編號由 0 到 6。當中有兩個關鍵的圓環。其中一個關鍵的圓環是 2 號圓環：移除此圓環後，剩下的圓環形成三條鍊 [1]，[0,5,3,4] 以及 [6]。另外一個關鍵的圓環是 3 號圓環：移除此圓環後，剩下的圓環形成三條鍊 [1,2,0,5]，[4]，以及 [6]。如果我們移除其他圓環，我們不能得到無交集的一組鍊。舉例來說，移去 5 號圓環後：雖然可以得到 [6] 這條鍊，但是鏈接的圓環 0,1,2,3 及 4 並沒有形成一條鍊。



陳述

一特定的圓環型態將傳達給你的程式，你的任務是數算其關鍵的圓環的數目。

開始時，有一些無交集的圓環。接著，這些圓環將被鏈接在一起。在任意的時刻，你的程式必須回傳目前關鍵的圓環的數目。特別為此，你必須實現三個子程式。

- `Init(N)` — 這個子程式一開始只被呼叫一次，以傳達有 N 個無交集的圓環，其編號從 0 到 $N - 1$ (包含在內) 作為初始的圓環型態。
- `Link(A, B)` — 將編號 A 以及編號 B 的圓環鏈接在一起。已保證 A 與 B 不相同，而且兩個圓環尚未鏈接在一起。除此之外， A 與 B 並沒有額外的沒有其他的條件，尤其是沒有任何物理限制需要考量。顯而易見，`Link(A, B)` 與 `Link(B, A)` 是一樣的。
- `CountCritical()` — 回傳目前鏈接的圓環型態中關鍵的圓環的數目。

範例

參考圖中 $N = 7$ 個圓環並假設它們一開始尚未鏈接在一起。在此列出一個可能的呼叫序列，在最後的一個呼叫後，我們得到了圖中所示的狀況。

呼叫	回傳
<code>Init(7)</code>	
<code>CountCritical()</code>	7
<code>Link(1, 2)</code>	
<code>CountCritical()</code>	7
<code>Link(0, 5)</code>	
<code>CountCritical()</code>	7
<code>Link(2, 0)</code>	
<code>CountCritical()</code>	7
<code>Link(3, 2)</code>	
<code>CountCritical()</code>	4
<code>Link(3, 5)</code>	
<code>CountCritical()</code>	3
<code>Link(4, 3)</code>	
<code>CountCritical()</code>	2

子任務 1 [20 分]

- $N \leq 5\,000$.

子 程 式 `CountCritical` 只會被呼叫一次，而且是在其他子程式被呼叫之後。子程式 `Link` 最多被呼叫 $5\,000$ 次。

子任務 2 [17 分]

- $N \leq 1\,000\,000$ 。

子程式 `CountCritical` 只會被呼叫一次，而且是在其他子程式被呼叫之後。子程式 `Link` 最多被呼叫 $1\,000\,000$ 次。

子任務 3 [18 分]

- $N \leq 20\,000$ 。
- 子程式 `CountCritical` 最多被呼叫 100 次；子程式 `Link` 最多被呼叫 10 000 次。

子任務 4 [14 分]

- $N \leq 100\,000$ 。
- 子程式 `CountCritical` 和 `Link` 最多共被呼叫 100 000 次。

子任務 5 [31 分]

- $N \leq 1\,000\,000$ 。
- 子程式 `CountCritical` 和 `Link` 最多共被呼叫 $1\,000\,000$ 次。

實現細節

你必須上傳一個檔案叫做 `rings.c` , `rings.cpp` 或 `rings.pas`。這個檔案實現上述的子程式並使用下面敘述的函式原型。

C/C++ 程式

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

Pascal 程式

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

你實現的子程式必須表現上述的行為。當然，你可以實現其他內部使用的子程式。你上傳的程式不能與標準輸入 / 輸出進行互動，也不能與其他的檔案進行互動。

範例評分系統

範例評分系統從讀取符合以下格式的輸入：

- 行 1：N, L；
- 行 2, ..., L+1：
 - -1 會呼叫 `CountCritical`；
 - A, B 是 `Link` 的參數。

範例評分系統會將 `CountCritical` 的結果列印出來。