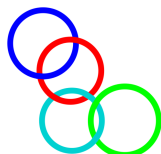


## 降落傘環

在 Leonardo 於 1485 發表的 "Codex Atlanticus" 文章中描述了一種早期而複雜的降落傘。Leonardo 的降落傘由一種布料縫製而成並以一個木制的金字塔型架支撐著使其打開。

## 互相連扣的環

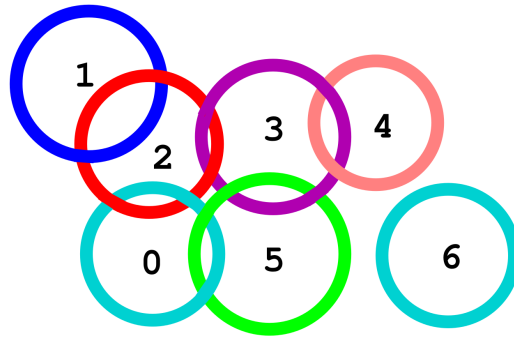
跳傘員 Adrain Nicholas 在 Leonardo 的降落傘發明後五百年曾經測試了該設計。在該測試中，人們用了一個現代的輕量結構將 Leonardo 的降落傘使綁在人體上。我們想要使用互相連扣圓環來為縫製的布料提供了鉤子。每一個圓環是用一些富彈性且高強度的金屬製成的小扣。圓環可以很簡單地扣連在一起，而且每一個圓環都可以從新打開或關閉。扣連的圓環可以構成一種特殊的型態叫做"鍊"。而所謂一個"鍊"就是指一連串的圓環互相扣連在一起所組成，其中每一個圓環只和其個隣の圓環(最多兩個)互相扣連在一起(如下圖所示)。而且在鍊一必存在一個開頭及一個結尾(它們每個只扣連到最多另外一個環)。要注意的是若單獨只有一個環時是特殊情況，它單獨亦算是一個鍊。



這裡當然是會有其他不同的扣連狀態的可能，因為一個環亦可以扣到其他三個或更多的環。我們說一個環是"關鍵環"如果我們將它打開並移除的話，其他餘下的環組成一些鍊的集合(又或者是根本沒有其他餘下的環)。換言之，餘下的只能夠是鍊而不是其他東西。

### 例子

試看看下圖的 7 個環，它們分別以 0 至 6 編號之。在這個情況下它共有兩個關鍵環。其中一個關鍵環是 2 號環：若它被除去的話，餘下的環就會組成三條分開的鍊 [1]，[0, 5, 3, 4] 及 [6]。另外一個關鍵環是 3 號環，若它被除去的話，餘下的環就會變成 [1, 2, 0, 5]，[4] 及 [6] 三條鍊。若我們除去其他的環的話，則餘下的環不會組成全是鍊的集合。舉例來說，若 5 號環被除去的話，雖然 [6] 是一個鍊，但扣連在一起的 0 號，1 號，2 號，3 號及 4 號環並不構成一個鍊。



## 說明

你的任務是要在給你的環的組態下找出一共有多少個關鍵環。

開始時，有分散而不互相扣連的環。接著，這些環將被一一扣連起來。在任意的時刻，你的程式必須回傳目前關鍵環的數目。特別為此，你必須實現三個子程式。

- `Init(N)` — 這個函數是在程式開初時被呼叫一次，它的作用是通知你的程式在開始時共有  $N$  個不相連的環，這些環是以  $0$  至  $N - 1$  編號。
- `Link(A, B)` —  $A$  及  $B$  是兩個扣連在一起的環。系統保證  $A$  及  $B$  是不同的而且未被直接扣連在一起：但除此之外對於  $A$  和  $B$  則沒有其他的條件限制，特別是物理上的條件限制。很明的, `Link(A, B)` 和 `Link(B, A)` 是相同的。
- `CountCritical()` — 返回目前狀態中關鍵環的數目。

### 例子

參考圖中  $N = 7$  個環並假設它們一開始尚未扣連在一起。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	返回
<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]

- $N \leq 1\,000\,000$ .
- 函數 `CountCritical` 在其他函數呼叫後被呼叫只 1 次; 函數 `Link` 最多被呼叫  $1\,000\,000$  次。

## 子任務 3 [18 分]

- $N \leq 20\,000$ .
- 函數 `CountCritical` 被呼叫最多 100 次; 函數 `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]

- $N \leq 1\,000\,000$ .
- 函數 `CountCritical` 及 `Link` 最多被呼叫  $1\,000\,000$  次。

## 編程細節

你必須只提交一程式檔案 `scrivener.c`、`scrivener.cpp` 或 `scrivener.pas`。這檔案必須以上述的相關函數編寫並符合下面的說明。

### C/C++ programs

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

## Pascal programs

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

這些子程式一定要根据上述的特點來編寫。當然你亦可自由地編寫其他在計算過程中需要的子程式。你所編寫的程式不能與標準輸入輸出有任何的直接互動。亦不能使用任何其他的檔案。

### 樣例 grader

樣例使用的 grader 將會需要讀取以下格式的輸入資料。

- 第 1 行:  $N, L$ ;
- 第 2 , ... ,  $L + 1$ :
  - -1 代表呼叫程式 `CountCritical`;
  - $A, B$  代表以它們為參數呼叫程式 `Link`.

樣例的 grader 將輸出由 `CountCritical` 生成的答案。