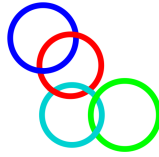


## Paraşüt halkaları

Bugün paraşüt olarak bilinen şeyin ilk ve gelişmiş versiyonu Leonardo'nun *Codex Atlanticus* (yaklaşık 1485 yılı) çalışmasında tanımlanmıştır. Leonardo'nun paraşütü piramit-şekilli ahşap bir yapı tarafından açık tutulan örtülü bir bez içermektedir.

### Bağlı halkalar

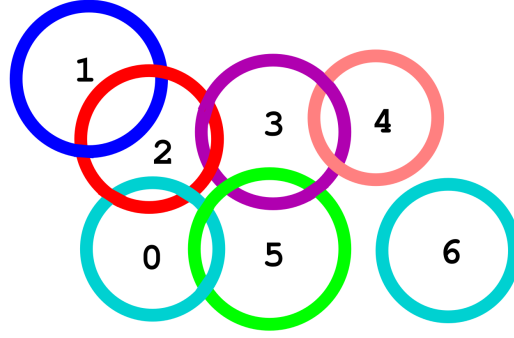
Paraşütçü Adrian Nicholas, Leonardo'nun tasarımını 500 yıl sonra test etti. Bunun için, modern hafif bir yapı Leonardo'nun paraşütünü insan bedenine bağladı. Bu iş için birbirine bağlı halkaları kullanmak istiyoruz, bu halkalar ayrıca örtülü beze askı desteği de sağlar. Her bir halka sağlam ve esnek bir materyalden yapılmıştır. Halkalar kolaylıkla birbirine bağlanabilir, çünkü her bir halka açılıp tekrar kapatılabilir. Bağlı halkaların özel bir konfigürasyonu *zincir* dir: Bir veya daha fazla sayıda halka içeren, en baş ve en sondaki hariç diğer halkaların tam olarak iki halkaya bağlı olduğu düzendir. Baş ve sondaki halkalar ise sadece bir halkaya şekilde gösterildiği gibi bağlıdır.



Tabiki başka konfigürasyonlarda mümkündür, çünkü bir halka üç veya daha fazla sayıda halkayada bağlanabilir. Bir halka *kritik* olarak adlandırılır eğer onu açıp çıkardığımızda, geri kalan halkalar birbirine bağlı olmayan bir dizi zincir kümesi oluşturuyorsa, (yada geride başka hiç bir halka kalmıyorsa).

### Örnek

Şekildeki 0 dan 6 ya kadar numaralanmış 7 halka için. Burada iki tane kritik halka vardır. Bunlardan birisi 2: onu çıkardığımızda, geri kalan halkalar şu zincirleri oluşturur [1], [0, 5, 3, 4] ve [6]. Diğeri ise 3: onu çıkardığımızda, geri kalan halkalar şu zincirleri oluşturur [1, 2, 0, 5], [4] ve [6]. Bunlar harici bir halkayı çıkardığımızısa, geriye bağlı olmayan ayrık kümeler kalmaz. Örneğin, halka 5 i çıkarırsak: [6] bir zincir olmasına karşın, bağlı halkalar 0, 1, 2, 3 ve 4 bir zincir oluşturmaz.



## Görev ifadesi

Sizin göreviniz verilen bir konfigürasyonda kritik halka sayısını bulmaktır.

Başlangıçta, belli sayıda birbirine bağlı olmayan (ayrık) halkalar vardır. Bundan sonra halkalar birbirine bağlanır. Herhangi bir anda, o anki konfigürasyonda kritik halka sayısını döndürmeniz istenebilir. Spesifik olarak üç fonksiyonu gerçekleştirmelisiniz.

- `Init(N)` — En başta bir kez çağrılır. İlk konfigürasyonda  $N$  tane (0 dan  $N - 1$  e kadar numaralı) birbirinden kopuk halka vardır anlamındadır.
- `Link(A, B)` —  $A$  ve  $B$  numaralı iki halkayı birbirine bağla anlamındadır.  $A$  ve  $B$  nin birbirinden farklı olduğu ve daha önce doğrudan bağlanmamış olduğu garanti edilmiştir; bunun haricinde,  $A$  ve  $B$  üzerinde ilave olarak fiziksel kısıt kaynaklı bir şart yoktur. `Link(A, B)` ve `Link(B, A)` eşdeğerdir.
- `CountCritical()` — Bağlı halkaların o anki konfigürasyonunda kritik halka sayısını döndür.

## Örnek

$N = 7$  halka olan şekil için bunların başlangıçta bağlı olmadığını varsayın. Aşağıda muhtemel bir çağırma sıralısı verilmiştir. Son çağırımdan sonra şekilde gösterilen durum elde edilir.

Çağırma	Dönen değer
<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

## Altgörev 1 [20 puan]

- $N \leq 5\,000$ .
- `CountCritical` fonksiyonu diğer fonksiyon çağırımları bittikten sonra yalnız bir kez çağrılır; `Link` en fazla 5 000 kez çağrılır.

## Altgörev 2 [17 puan]

- $N \leq 1\,000\,000$ .
- `CountCritical` fonksiyonu diğer fonksiyon çağırımları bittikten sonra yalnız bir kez çağrılır; `Link` en fazla 1 000 000 kez çağrılır.

## Altgörev 3 [18 puan]

- $N \leq 20\,000$ .
- `CountCritical` fonksiyonu en fazla 100 kez çağrılır; `Link` fonksiyonu en fazla 10 000 kez çağrılır.

## Altgörev 4 [14 puan]

- $N \leq 100\,000$ .
- `CountCritical` ve `Link` fonksiyonları ikisi toplamda en fazla 100 000 kez çağrılır.

## Altgörev 5 [31 puan]

- $N \leq 1\,000\,000$ .
- `CountCritical` ve `Link` fonksiyonları ikisi toplamda en fazla 1 000 000 kez çağrılır.

## Gerçekleştirim detayları

Tam olarak tek bir dosya göndermelisiniz, dosyanın adı `rings.c`, `rings.cpp` yada `rings.pas` olmalı. Bu dosya yukarıda anlatılmış olan fonksiyonları aşağıdaki fonksiyon başlığı yapısında gerçekleştirmelidir.

### C/C++ programı

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

## Pascal programı

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

Bu fonksiyonlar yukarıda anlatıldığı gibi davranmalıdır. Bu fonksiyonların kullanacağı başka fonksiyonları isterseniz kendiniz ekleyebilirsiniz. Kodunuz standart girdi/çıkı yada başka bir dosyadan okuma yazma yapmamalıdır.

## Örnek notlandırıcı

Örnek notlandırıcı girdiyi aşağıdaki formatta okumaktadır:

- Satır 1: N, L;
- satırlar 2, ..., L + 1:
  - -1 ( CountCritical yi çağırarak için);
  - A, B parametreleri ( Link için).

Örnek notlandırıcı CountCritical dan dönen tüm sonuçları basmaktadır.