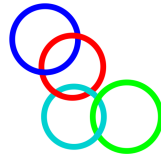


Paraşüt halqaları

Bizim indi paraşüt adlandırdığımız vasitənin erkən və mükəmməl versiyası Leonardo da Vinçinin *Codex Atlanticus* (1485) əsərində təsvir olunub. Leonardo'nun paraşütü piramida formalı ağac karkasın üzərinə sarınmış parçadan ibarət idi.

Əlaqəli halqalar

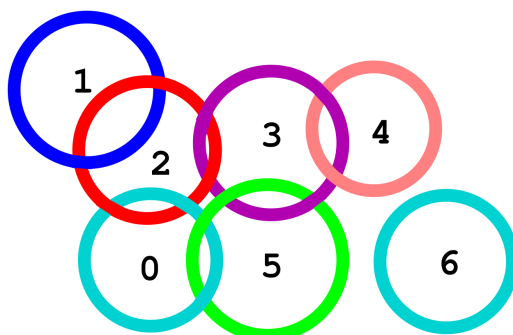
Paraşütçü Adrian Nicholas 500 ildən sonra Leonardonun layihəsini sınaqdan keçirdi. Bunun üçün, sadələşdirilmiş müasir konstruksiyada Leonardonun paraşütü insan bədənində bağlandı. Мы хотим использовать связанные кольца, где также предусмотрены крюки для ткани. Каждое кольцо - это маленькая застежка, сделанная из гибкого и прочного материала. Кольца легко соединять вместе и каждое кольцо может быть разъединено и снова соединено. Особая конфигурация связанных колец - это "цепь": последовательность одного или более связанных колец в которой каждое кольцо соединено с двумя другими кольцами, кроме первого и последнего, которые соединены только с одним другим кольцом, как показано ниже.



Other configurations are clearly possible, since a ring can be linked to three or more other rings. We say that a ring is *critical* if after opening and removing it, all remaining rings form a set of chains (or there are no other rings left). In other words, there can be nothing but chains left.

Örnək

Növbəti şəkildə 0-dan 6-dək nömrələnmiş 7 halqaya baxaq. İki kritik halqa var. Kritik halqalardan biri 2-dir: onu uzaqlaşdırdıqdan sonra qalan halqalar [1], [0, 5, 3, 4] və [6] zəncirlərini əmələ gətirir. Digər kritik halqa 3-dür: onu uzaqlaşdırdıqdan sonra qalan halqalar [1, 2, 0, 5], [4] və [6] zəncirlərini əmələ gətirir. Əgər biz hər hansı başqa halqanı uzaqlaşdırsaq, əlaqəli olmayan zəncirlər çoxluğu ala bilmərik. Məsələn, 5 halqasını uzaqlaşdırdıqdan sonra: [6] zəncirini alsaq da, əlaqəli 0, 1, 2, 3 və 4 halqaları zəncir əmələ gətirmir.



Məsələnin qoyuluşu

Proqramınız verilmiş konfigurasiyada kritik halqaların sayını hesablamalıdır.

Başlanğıcda əlaqəli olmayan müəyyən sayda halqa var. Sonra, halqalar bir-birinə bağlanır. İstənilən zamanda sizdən cari konfigurasiyadakı kritik halqaların sayını qaytarmaq tələb oluna bilər. Xüsusən, siz üç önəmli proseduru reallaşdırmalısınız.

- `Init(N)` — bu prosedur yalnız bir dəfə başlanğıcda çağırılır və ilkin olaraq 0-dan $N-1$ dək nömrələnmiş N əlaqəli olmayan halqanın olmasını bildirir.
- `Link(A, B)` — A və B nömrəli halqalar bir-birinə bağlanır. A və B -nin fərqli olmasına və birbaşa bağlı olmamalarına zəmanət verilir; bundan başqa, A və B -yə əlavə şərt qoyulmur, xüsusən, fiziki məhdudiyyətlərlə bağlı şərt yoxdur. Aydındır ki, `Link(A, B)` və `Link(B, A)` ekvivalentdir.
- `CountCritical()` — əlaqəli halqaların cari konfigurasiyadakı kritik halqaların sayını qaytarır.

Örnek

$N = 7$ halqadan ibarət şəklimizə qayıdaq və fərz edək ki, onlar başlanğıcda əlaqəli deyillər. Biz mümkün çağırışlar ardıcılığını göstərəcəyik, belə ki, sonuncu çağırışdan sonra şəkildə göstərilmiş vəziyyəti alırıq.

Çağırışlar	Qaytarılan qiymətlər
<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

Altməsələ 1 [20 bal]

- $N \leq 5\,000$.
- `CountCritical` funksiyası bütün çağırışlardan sonra yalnız bir dəfə çağırılır; `Link` funksiyası ən çoxu 5 000 dəfə çağırılır.

Altməsələ 2 [17 bal]

- $N \leq 1\,000\,000$.
- `CountCritical` funksiyası bütün çağırışlardan sonra yalnız bir dəfə çağırılır; `Link` ən çoxu 1 000 000 dəfə çağırılır.

Altməsələ 3 [18 bal]

- $N \leq 20\,000$.
- `CountCritical` funksiyası ən çoxu 100 dəfə çağırılacaq; `Link` funksiyası ən çoxu 10 000 dəfə çağırılacaq.

Altməsələ 4 [14 bal]

- $N \leq 100\,000$.
- The functions `CountCritical` and `Link` are called, in total, at most 100 000 times.

Altməsələ 5 [31 bal]

- $N \leq 1\,000\,000$.
- The functions `CountCritical` and `Link` are called, in total, at most 1 000 000 times.

Implementation details

You have to submit exactly one file, called `rings.c`, `rings.cpp` or `rings.pas`. This file implements the subprograms described above using the following signatures.

C/C++ proqramları

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

Pascal proqramları

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

These subprograms must behave as described above. Of course you are free to implement other subprograms for their internal use. Your submissions must not interact in any way with standard input/output, nor with any other file.

Yoxlayıcı modula örnək

Örnək yoxlayıcı giriş verilənlərini aşağıdakı formatda oxuyur:

- Sətir 1: N, L;
- Sətirlər 2, ..., L + 1:
 - CountCritical çağırmaq üçün -1;
 - Link üçün A, B parametrləri.

Örnək yoxlayıcı CountCritical funksiyasından bütün nəticələri çap edəcək.