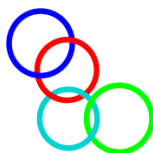


Anillos de paracaídas

Una temprana y muy sofisticada versión de lo que ahora llamamos paracaídas es descrita en el *Codex Atlanticus* de Leonardo (aprox. 1485). El paracaídas de Leonardo consistía de una tela sellada de lino mantenida abierta por una estructura de madera de forma piramidal.

Anillos enganchados

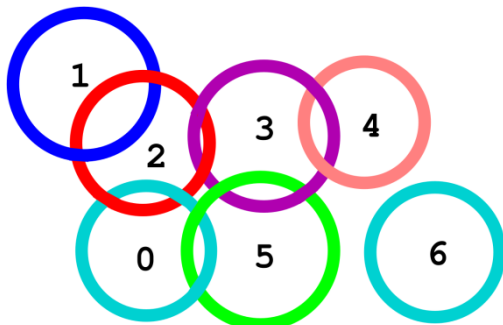
El paracaidista Adrian Nicolás probó el diseño de Leonardo más de 500 años después. Para ello, una estructura moderna liviana ataba el paracaídas de Leonardo al cuerpo humano. Deseamos usar anillos enganchados, que también provean ganchos para la tela de lino sellada. Cada anillo es un pequeño mosquetón hecho de material flexible y fuerte. Los anillos pueden ser fácilmente enganchados entre sí por cuanto cada anillo puede ser abierto y nuevamente cerrado. Una especial configuración de enganchado de anillos es la *cadena*: una secuencia de uno o más anillos enganchados en la cual cada anillo es enganchado a otros dos anillos, salvo el primero y el último, que están enganchados a sólo un otro anillo, tal como se ilustra debajo.



Otras configuraciones son claramente posibles, ya que un anillo puede ser enganchado a tres o más anillos. Decimos que un anillo es *crítico* si después de abrirlo y removerlo, todos los anillos restantes forman un conjunto de cadenas disjuntas (o no quedan otros anillos).

Ejemplo

Considere los 7 anillos en la figura siguiente, numerados de 0 a 6. Hay dos anillos críticos. Un anillo crítico es el 2: después de su remoción, los anillos restantes forman las cadenas [1], [0, 5, 3, 4] y [6]. El otro anillo crítico es el 3: después de su remoción, los anillos restantes forman las cadenas [1, 2, 0, 5], [4] y [6]. Si removemos cualquier otro anillo, no obtenemos un conjunto de cadenas disjuntas. Por ejemplo, después de remover el anillo 5: aunque tenemos que [6] es una cadena, el enganchado de los anillos 0, 1, 2, 3 y 4 no forman una cadena.



Enunciado

Tu tarea es contar el número de anillos críticos en una configuración dada que será comunicada a tu programa.

Al comienzo, hay un cierto número de anillos disjuntos. Después de eso, los anillos son enganchados entre sí. En cualquier instante se te puede pedir que retornes el número de anillos críticos en la configuración corriente. Específicamente, debes implementar tres rutinas.

- `Init(N)` — Es llamada exactamente una vez al comienzo para comunicar que hay N anillos disjuntos numerados de 0 a $N - 1$ (inclusive) en la configuración inicial.
- `Link(A, B)` — Los dos anillos numerados A y B son enganchados entre sí. Se garantiza que A y B son diferentes y no están directamente enganchados; fuera de esto, no hay condiciones adicionales sobre A y B , en particular no hay condiciones provenientes de restricciones físicas. Claramente, `Link(A, B)` y `Link(B, A)` son equivalentes.
- `CountCritical()` — Retorna el número de anillos críticos para la configuración corriente de anillos enganchados.

Ejemplo

Considera nuestra figura con $n = 7$ anillos y supone que ellos están inicialmente desenganchados. Mostramos una posible secuencia de llamadas, donde después de la última llamada obtenemos la situación expuesta en nuestra figura.

Llamada	Retorna
<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

Subtarea 1 [20 puntos]

- $N \leq 5\,000$.
- La función `CountCritical` es llamada sólo una vez, después de todas las otras llamadas; la función `Link` es llamada a lo más 5 000 veces.

Subtarea 2 [17 puntos]

- $N \leq 1\,000\,000$.
- La función `CountCritical` es llamada sólo una vez, después de todas las otras llamadas; La función `Link` es llamada a lo más 1 000 000 de veces.

Subtarea 3 [18 puntos]

- $N \leq 20\,000$.
- La función `CountCritical` es llamada a lo más 100 veces; La función `Link` es llamada a lo más 10 000 veces.

Subtarea 4 [14 puntos]

- $N \leq 100\,000$.
- Las funciones `CountCritical` y `Link` son llamadas, en total, a lo más 100 000 veces.

Subtarea 5 [31 puntos]

- $N \leq 1\,000\,000$.
- Las funciones `CountCritical` y `Link` son llamadas, en total, a lo más 1 000 000 de veces.

Detalles de implementación

Debes enviar exactamente un archivo, llamada `rings.c`, `rings.cpp` o `rings.pas`. Este archivo implementa los subprogramas descritos arriba usando los siguientes encabezamientos.

programas C/C++

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

programas Pascal

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

Estos subprogramas se deben comportar tal como está descrito arriba. Por supuesto eres libre de implementar otros subprogramas para su uso interno. Sus envíos no deben interactuar de ningún modo con “standard input/output”, ni con ningún otro archivo.

Calificador simple

El calificador simple lee su entrada en el siguiente formato:

- línea 1: N, L;
- líneas 2, ..., L + 1:
 - -1 para invocar `CountCritical`;
 - A, B parámetros para `Link`.

El calificador simple imprimirá todos los resultados de `CountCritical`.