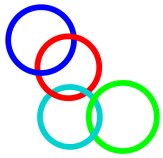
Anillos de Paracaídas

Una antigua y poco sofisticada version de lo que actualmente llamamos paracaidas es descrita en el *Codex Atlanticus* (ca. 1485) de Leonardo Da Vinci. El paracaidas de Leonardo consistia en una tela de lino sellada (costurada) la cual se mantenia abierta por una estructura de madera en forma de piramide.

Anillos enlazados

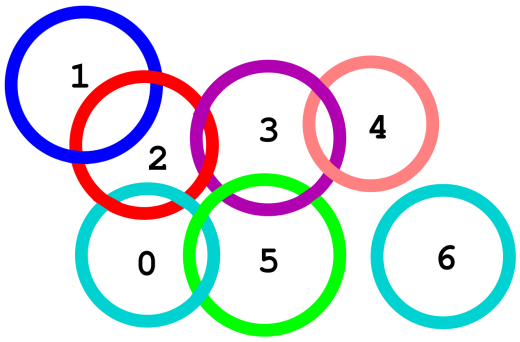
El paracaidista Adrian Nicholas probo el diseño de Leonardo 500 años despues. Para esto, una moderna y liviana estructura unia el paracaidas de Leonardo con el cuerpo humano. Nosotros queremos usar anillos enlazados, que tambien se puedan enganchar a la tela de lino sellada (costurada). Cada anillo esta hecho de un material fuerte y flexible. Los anillos pueden ser facilmente enlazados entre ellos considerando que cada anillo se puede abrir y volver a cerrar. Una configuracion especial de anillos enlazados es la *cadena*. Una *cadena* es una secuencia de uno o mas anillos enlazados en la cual cada anillo esta solamente conectado (a lo mucho) con sus dos vecinos, tal como muestra el dibujo de abajo. Esta secuencia debe tener un inicio y un fin (anillos que estén conectados con a lo mucho un anillo cada uno). Especificamente, un solo anillo es tambien una cadena.



Otras configuraciones son tambien posibles, dado que un anillo puede ser enlazado con tres o mas anillos. Decimos que un anillo es *critico* si despues de abrirlo y retirarlo, todos los restantes anillos forman un conjunto de cadenas (o no quedan más anillos). En otras palabras, solo pueden quedar cadenas.

Ejemplo

Considere los 7 anillos en la siguiente figura, numerados del 0 al 6. Alli tenemos 2 anillos criticos. Un anillos critico es el 2: despues de removerlo, los restantes anillos forman las cadenas [1], [0, 5, 3, 4] y [6]. El otro anillo critico es el 3: despues de removerlo, los restantes anillos forman las cadenas [1, 2, 0, 5], [4] y [6]. Si removemos otro anillo no obtendremos un conjunto de cadenas disjuntas. Por ejemplo, despues de remover el anillo 5: aunque tengamos a [6] como una cadena, los anillos enlazados 0, 1, 2, 3 y 4 no forman una cadena.



Enunciado

Tu tarea es contar el numero de anillos criticos en una determinada configuracion que le sera comunicada a tu programa.

Al principio, habrá un número determinado de anillos disjuntos. Despues de eso se entrelazaran todos los anillos. En cualquier momento se te preguntara el numero de anillos criticos en la configuracion actual. Especificamente, tu tienes que implementar 3 rutinas.

- `Init`(N) — sera llamada exactamente una vez al inicio para comunicar que existen N anillos disjuntos numerados de 0 a N - 1 (inclusive) en la configuracion inicial.
- `Link`(A, B) — los dos anillos A y B se entrelazan. Esta garantizado que A y B son diferentes y que no estan conectados; ademas de esto no existe condiciones adicionales para A y B, en particular no existen condiciones relacionadas a restricciones fisicas. Claramente, `Link`(A, B) y `Link`(B, A) son equivalentes.
- `CountCritical`() — devuelve el numero de anillos criticos para la configuracion actual de los anillos enlazados.

Ejemplo

Considere nuestra figura con N = 7 anillos y suponga que ellos estan inicialmente sin enlazarse. Mostramos una posible secuencian de llamadas, donde despues de la ultima podemos obtener la situacion descrita en la figura.

Call	Returns
`Init`(7)	
`CountCritical`()	7
`Link`(1, 2)	
`CountCritical`()	7
`Link`(0, 5)	
`CountCritical`()	7
`Link`(2, 0)	
`CountCritical`()	7
`Link`(3, 2)	
`CountCritical`()	4
`Link`(3, 5)	
`CountCritical`()	3
`Link`(4, 3)	
`CountCritical`()	2

Subtask 1 [20 puntos]

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

Subtask 2 [17 puntos]

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

Subtask 3 [18 puntos]

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

Subtask 4 [14 puntos]

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

Subtask 5 [31 puntos]

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

Detalles de implementación

Debes enviar solamente un archivo, llamado `rings.c`, `rings.cpp` o `rings.pas`. Este archivo implementa los subprogramas (subrutinas) descritos arriba usando los siguientes encabezados.

Programas en C/C++

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

Programas en Pascal

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

Estos subprogramas (subrutinas) deben comportarse como se describen en el enunciado. Por supuesto eres libre de implementar otros suprogramas de uso interno. no deben interactuar de ninguna forma con la entrada/salida standar, o con algun otro archivo.

Calificador ejemplo

El calificador ejemplo lee la entrada y la salida con el siguiente formato:

- linea 1: N, L;
- lineas 2, …, L + 1:
  - -1 para llamar `CountCritical`;
  - A, B los parametros para `Link`.

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