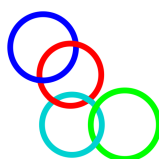


Anillos de paracaídas

Una versión antigua de lo que hoy llamamos paracaídas se encuentra descrita en el libro *Codex Atlanticus* de Leonardo (alrededor de 1485). Dicho paracaídas consistía de una tela sellada de lino que se mantenía abierta por una estructura piramidal de madera.

Anillos entrelazados

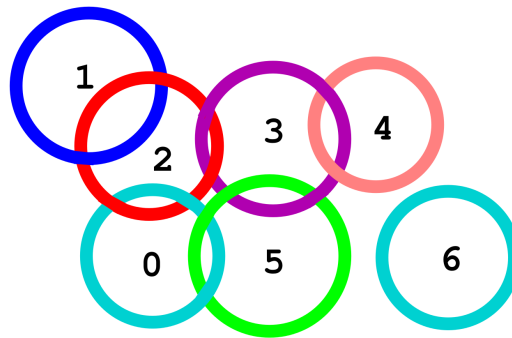
El paracaidista Adrian Nicholas probó el diseño de Leonardo 500 años después. Para esto, una estructura hecha con materiales ligeros modernos ató el paracaídas de Leonardo a la persona que lo probó. Queremos usar anillos entrelazados, los cuales proporcionan enganches para la tela de lino sellada. Cada anillo es una argolla hecha de un material fuerte y flexible. Los anillos fácilmente se pueden entrelazar pues cada anillo se puede abrir y volver a cerrar. Una configuración especial de anillos es una *cadena*. Una *cadena* es una secuencia de anillos donde cada anillo sólo está conectado con sus (a lo mucho) dos vecinos, como muestra la imagen. Esta secuencia debe tener un principio y un final (anillos que estén conectados con a lo mucho un anillo cada uno). Específicamente, un solo anillo también es una cadena.



Otras configuraciones también son posibles, dado que un anillo se puede entrelazar con tres o más anillos. Se dice que un anillo es *crítico* si después de abrirlo y quitarlo todos los anillos restantes forman un conjunto disjunto de cadenas (o no quedan más anillos). En otras palabras, sólo pueden quedar cadenas.

Ejemplo

Considere los siete anillos de la siguiente figura, numerados de 0 a 6. Hay dos anillos críticos. Un anillo crítico es el número 2: después de quitarlo, los anillos restantes forman las cadenas [1], [0,5,3,4] y [6]. El otro anillo crítico es el número 3: después de quitarlo, los anillos restantes forman las cadenas [1,2,0,5], [4] y [6]. Si quitamos cualquier otro anillo el conjunto restante no es un conjunto de cadenas disjuntas. Por ejemplo, después de quitar el anillo 5: aunque el anillo [6] es una cadena, los anillos entrelazados 0, 1, 2, 3 y 4 no forman una cadena.



Enunciado

Su tarea es contar el número de anillos críticos en una configuración dada que le será comunicada a su programa.

Al principio, habrá un número determinado de anillos disjuntos. Después de eso se entrelazarán los anillos. En cualquier momento se le preguntará al programa el número de anillos críticos en la configuración actual. Específicamente, debe implementar tres rutinas.

- `Init(N)` — Será llamada una única vez al principio del programa para comunicar que hay N anillos disjuntos numerados de 0 a $N - 1$ (inclusive) en la configuración inicial.
- `Link(A, B)` — Los anillos A y B se entrelazan. Se garantiza que A y B son diferentes y que no se encuentran entrelazados previamente; note que no hay condiciones adicionales, en particular no hay condiciones referentes a las limitaciones físicas. Claramente, `Link(A, B)` y `Link(B, A)` son equivalentes.
- `CountCritical()` — Devuelve el número de anillos críticos en la configuración actual de anillos entrelazados.

Ejemplo

Considere la figura anterior con $N = 7$ anillos y suponga que inicialmente se encuentran sin entrelazar. Mostramos una posible secuencia de llamados, donde después del último llamado llegamos a la situación descrita en nuestra figura.

Llamado	Devuelve
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

Subtarea 1 [20 puntos]

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

Subtarea 2 [17 puntos]

- $N \leq 1\,000\,000$.
- La función `CountCritical` se llama sólo una vez, después de todas las otras llamadas; la función `Link` se llama a lo mucho 1 000 000 de veces.

Subtarea 3 [18 puntos]

- $N \leq 20\,000$.
- La función `CountCritical` se llama a lo mucho 100 veces; la función `Link` se llama a lo mucho 10 000 veces.

Subtarea 4 [14 puntos]

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

Subtarea 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

Tiene que enviar exactamente un archivo, `rings.c`, `rings.cpp` o `rings.pas`. Este archivo implementa las subrutinas descritas 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;
```

Estas subrutinas se deben comportar como describe el enunciado. Por supuesto, usted es libre de implementar cualquier otra subrutina para uso interno. Sus envíos no deben interactuar de ninguna forma con la entrada o salida estándar, o con algún otro archivo.

Calificador ejemplo

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

- Línea 1: N, L ;
- Línea 2, ..., $L + 1$:
 - -1 para llamar `CountCritical`;
 - A, B los parámetros para `Link`.

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