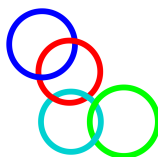


## Ogniwa

Okolo roku 1485 Leonardo da Vinci opisał dosyć skomplikowaną wersję tego, co obecnie nazywamy spadochronem. Spadochron Leonarda składał się z materiału rozłożonego na konstrukcji w kształcie piramidy.

### Połączone ogniwa

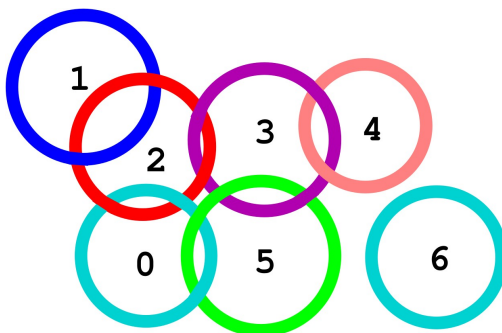
Ponad 500 lat później spadochroniarz Adrian Nicholas zrealizował projekt Leonarda. Stworzył nowoczesną, lekką konstrukcję, która pozwalała przymocować spadochron Leonarda do ludzkiego ciała. Częścią konstrukcji są połączone ze sobą ogniwa, do których mocuje się materiał. Każde ogniwo to mały pierścień ze sprężystego, wytrzymałego materiału. Ogniwa można łatwo łączyć, gdyż każde z nich można otworzyć, a następnie ponownie zamknąć. *Łańcuchem* nazywamy taki układ połączonych ogniw, w którym każde ogniwo jest połączone z co najwyżej dwoma sąsiadami. Łańcuch posiada początek i koniec, czyli ogniwa mające co najwyżej jednego sąsiada. W szczególności pojedyncze ogniwo również tworzy łańcuch.



Oczywiście możliwe są także inne konfiguracje ogniw, bo każde ogniwo może być połączone z dowolnie dużą liczbą ogniw. Ogniwo nazywamy *krytycznym*, jeśli po jego otwarciu i usunięciu wszystkie pozostałe ogniwa tworzą zbiór łańcuchów (w szczególności, jeśli istnieje tylko jedno ogniwo, to jest ono krytyczne).

### Przykład

Spójrzmy na poniższy obrazek z 7 ogniwami ponumerowanymi od 0 do 6. Dwa ogniwa są krytyczne. Jedno z nich to ogniwo 2: po jego usunięciu pozostałe ogniwa tworzą trzy łańcuchy: [1], [0, 5, 3, 4] oraz [6]. Drugie krytyczne ogniwo to 3: po jego usunięciu pozostałe ogniwa tworzą trzy łańcuchy: [1, 2, 0, 5], [4] i [6]. Jeśli usuniemy jakiegokolwiek inne ogniwo, nie otrzymamy zbioru rozłącznych łańcuchów. Załóżmy na przykład, że usuwamy ogniwo 5. Wprawdzie zostanie łańcuch [6], jednak ogniwa 0, 1, 2, 3 i 4 nie tworzą łańcucha.



## Zadanie

Twoim zadaniem jest napisanie programu, który będzie liczył krytyczne ogniwa w podanej konfiguracji ogniw.

Początkowa konfiguracja składa się z pewnej liczby niepołączonych ogniw. Następnie ogniwa są ze sobą łączone. W każdym momencie Twój program może zostać poproszony o obliczenie liczby krytycznych ogniw w aktualnej konfiguracji. Powinieneś zaimplementować trzy funkcje:

- `Init(N)` — funkcja ta zostanie wywołana dokładnie raz, na samym początku działania programu. Parametr `N` oznacza liczbę niepołączonych ogniw w początkowej konfiguracji. Ogniwa są ponumerowane liczbami od 0 do `N-1`.
- `Link(A, B)` — ogniwa o numerach `A` i `B` zostają ze sobą połączone. Możesz założyć, że `A` jest różne od `B` i ogniwa te nie zostały ze sobą wcześniej (bezpośrednio) połączone. Poza tym, ogniwa można łączyć zupełnie dowolnie, kompletnie ignorując prawa fizyki i zdrowy rozsądek. Oczywiście wywołania `Link(A, B)` i `Link(B, A)` są ze sobą równoważne.
- `CountCritical()` — zwraca liczbę krytycznych ogniw w aktualnej konfiguracji.

## Przykład

Rozważmy przykład z `N = 7` ogniwami. Początkowo są one niepołączone. Pokażemy możliwy ciąg wywołań, po którym otrzymamy konfigurację z rysunku.

Wywołanie	Zwrócona wartość
<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

## Podzadanie 1 [20 punktów]

- $N \leq 5\,000$ .
- Funkcja `CountCritical` zostanie wywołana tylko raz, po wszystkich innych wywołaniach; funkcja `Link` zostanie wywołana co najwyżej 5 000 razy.

## Podzadanie 2 [17 punktów]

- $N \leq 1\,000\,000$ .
- Funkcja `CountCritical` zostanie wywołana tylko raz, po wszystkich innych wywołaniach; funkcja `Link` zostanie wywołana co najwyżej  $1\,000\,000$  razy.

## Podzadanie 3 [18 punktów]

- $N \leq 20\,000$ .
- Funkcja `CountCritical` zostanie wywołana co najwyżej 100 razy; funkcja `Link` zostanie wywołana co najwyżej 10 000 razy.

## Podzadanie 4 [14 punktów]

- $N \leq 100\,000$ .
- Funkcje `CountCritical` i `Link` zostaną wywołane (łącznie) co najwyżej 100 000 razy.

## Podzadanie 5 [31 punktów]

- $N \leq 1\,000\,000$ .
- Funkcje `CountCritical` i `Link` zostaną wywołane (łącznie) co najwyżej  $1\,000\,000$  razy.

## Szczegóły implementacji

Należy zgłosić dokładnie jeden plik o nazwie `rings.c`, `rings.cpp` lub `rings.pas`. Powinien on zawierać implementacje opisanych powyżej funkcji.

### Programy w C/C++

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

### Programy w Pascalu

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

Funkcje powinny działać dokładnie tak, jak opisano powyżej. Twój program nie powinien korzystać ze standardowego wejścia, standardowego wyjścia lub jakichkolwiek plików.

### Przykładowy moduł oceniający

Przykładowy moduł oceniający wczytuje dane w następującym formacie:

- wiersz 1:  $N, L$ ;
- wiersze 2, ...,  $L + 1$ :
- -1 oznacza wywołanie `CountCritical`;
- dwie liczby  $A, B$  oznaczają wywołanie `Link` z parametrami  $A$  i  $B$ .

Przykładowy moduł oceniający wypisuje wszystkie wyniki zwrócone przez `CountCritical`.