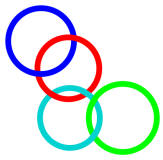
Парашутни кръгове

Една ранна и доста изобретателна конструкция, на това, което днес наричаме "парашут" е описана в книгата на Леонардо *Codex Atlanticus* (около 1485 г.) Парашутът на Леонардо се състои от чаршафи, залепени върху открита дървена конструкция с форма на пирамида.

"Сързани кръгове"

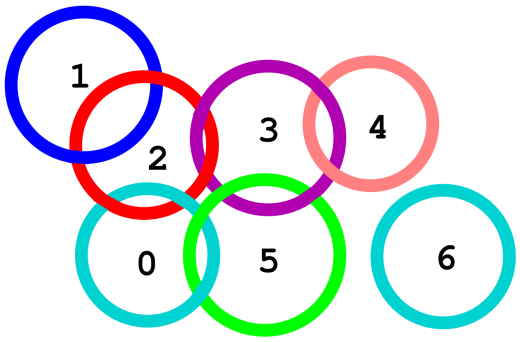
Парашутистът Ейдриън Никълъс тества дизайна на Леонардо повече от 500 години по-късно. За целта една съвременна модерна лека структура привързва парашут на Леонардо към човешкото тяло. Искаме да използваме свързани пръстени, които се закачат с куки за залепените чаршафи. Всеки пръстен е малък карабинер изработен от гъвкав и здрав материал. Пръстените могат лесно да бъдат свързани помежду си, като всеки пръстен може да се отвори и затвори. Една специална конфигурация от свързани пръстени се нарича *верига*: поредица от един или повече свързани пръстени, в които всеки пръстен е свързан с два други пръстена, с изключение на първия и последнния, които са свързани само с един друг пръстен, както е показано по-долу. Един единствен пръстен също образува верига.



Възможни са и други конфигурации, тъй като пръстен може да бъде свързан с три или повече други пръстени. Ние казваме, че един пръстен е "критичен", ако след премахването му, всички останали пръстени образуват набор от непресичащи вериги (или това е бил единственият пръстен в конфигурацията).

Пример

Да разгледаме 7 пръстени на следващата фигура, номерирани от 0 до 6. Има два критични пръстена. Единият критичен пръстен е 2: след отстраняването му, останалите пръстени образуват вериги [1], [0, 5, 3, 4] и [6]. Друг критичен пръстен е 3: след отстраняването му, останалите пръстени образуват веригите [1, 2, 0, 5], [4] и [6]. Ако премахнем някой от останалите пръстени не се получава набор от непресичащи вериги. Например, след премахване на пръстен 5: въпреки че пръстенът [6] сам представлява верига, останалите пръстени 0, 1, 2, 3 и 4 са свързани, но не образуват верига.



Задача

Вашата задача е да се преброят критичните пръстени в дадена конфигурация, която ще бъде съобщена на вашата програма.

В началото, има определен брой непресичащи се пръстени. След това, някои от пръстените се свързват помежду си. Във всеки един момент, може да бъде поискано да се пресметне броя на критичните пръстени в текущата конфигурация. Конкретно, трябва да се приложат три процедури.

- `Init`(N) — тази функция се извиква еднократно в началото. Параметърът показва, че има N непресичащи пръстена, номерирани от 0 до N - 1 (включително) в първоначалната конфигурация.
- `Link`(A, B) — двата пръстена с номера A и B се свързват помежду си. Гарантирано е, че пръстените A и B са различни и не са свързани пряко; няма никакви други условия за A и B, в частност не възникват проблеми при свързването, вследствие на физически причини. Ясно е, че двете команди `Link`(A, B) и `Link`(B, A) са еквивалентни..
- `CountCritical`() — връща броя на критичните пръстени за текущата конфигурация от свързани пръстени.

Пример

Да разгледаме фигурата със 7 пръстена. Отначало пръстените не са свързани. Показана е една възможна поредица от команди, които водят до конфигурацията на фигурата.

Заявка	Връща
`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 точки]

Подзадача 1 (20 точки)

- Функцията `CountCritical` се извиква еднократно, след извикването на всички други функции.

Функцията `Link` се извиква най-много 5 000 пъти.

Subtask 2 [17 points]

- $N \leq 1\,000\,000$.
- Функцията `CountCritical` се извиква еднократно, след извикването на всички други функции.

Функцията `Link` се извиква най-много 1 000 000 пъти.

Subtask 3 [18 points]

- $N \leq 20\,000$.
- Функцията `CountCritical`се извиква най-много 100 пъти; Функцията `Link` се извиква най-много 1000пъти..

Subtask 4 [14 points]

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

Subtask 5 [31 points]

- $N \leq 1\,000\,000$.
- 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++ programs

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

Pascal programs

```
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.

Sample grader

The sample grader reads the input in the following format:

- line 1: N, L;
- lines 2, …, L + 1:
  - -1 to invoke `CountCritical`;
  - A, B parameters to `Link`.

The sample grader will print all results from `CountCritical`.