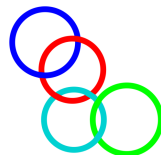


パラシュートリング (Parachute rings)

我々が今日パラシュートと呼んでいるものの初期のとても洗練されたバージョンは、レオナルド (Leonardo) の *Codex Atlanticus* (1485 年頃) に描かれている。レオナルドのパラシュートは、ピラミッド型の木製の骨組みに麻の布が貼り付けられた形をしている。

つながったリング (Linked rings)

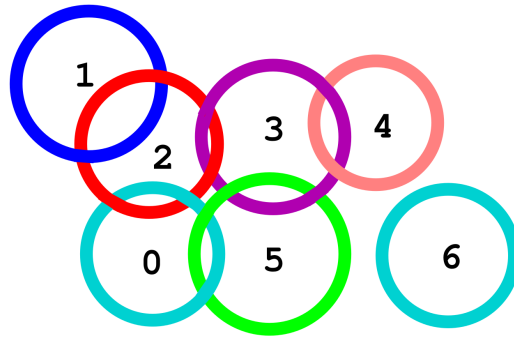
スカイダイバーのアドリアン・ニコラス (Adrian Nicholas) は、レオナルドの設計を 500 年以上後になってから試した。そのために、現代的な軽量の骨組みにレオナルドのパラシュートと人間の体を括り付けた。我々は麻の布の留め金としてつながったリングを使いたい。各リングは柔軟で丈夫な素材で出来ている。どのリングも開閉が可能なので、リング同士を簡単につなげることができる。つながったリングの特別な配置が鎖 (chain) である。鎖とは、リングの列であって、下図のように各リングが (高々 2 つの) 隣り合ったリングのみとつながっているものである。この列には始点と終点 (高々 1 個の他のリングとつながっているリング) がなければならない。特に、1 つだけのリングも鎖である。



リングは 3 つ以上の他のリングと直接つなげることできるので、鎖でない配置も当然可能である。あるリングが重大 (critical) であるとは、そのリングを開いて取り除いたとき、残りのリングがいくつかの鎖を形作る (または、リングが 1 つも残らない) ことをいうことにする。つまり、取り除いたときに鎖以外のものが残らないことをいう。

例 (Example)

下図のように 0 から 6 までの番号がついた 7 つのリングを考えよう。このとき重大なリングは 2 つある。1 つはリング 2 である：これを取り除くと、残りのリングは鎖 [1], [0, 5, 3, 4], [6] を形作る。もう 1 つの重大なリングはリング 3 である：これを取り除くと、残りのリングは鎖 [1, 2, 0, 5], [4], [6] を形作る。それ以外のリングを取り除いた場合は、残りのリングがいくつかの互いに交わらない鎖を形作ることはない。例えばリング 5 を取り除くと、[6] は鎖だが、1 つにつながったリング 0, 1, 2, 3, 4 は鎖を形作らない。



問題文 (Statement)

あなたの課題は，あなたのプログラムに伝えられるリングの配置における重大なリングの数を数えることである．

最初にいくつかのリングがどの 2 つもつながれていない状態で存在し，その後リングがつながれていく．任意のタイミングであなたはその時点のリングの配置での重大なリングの数を聞かれうる．具体的には，あなたは 3 つのルーチンを実装しなければならない．

- `Init(N)` — このルーチンは最初に 1 回だけ呼ばれ，初期配置では 0 から $N - 1$ までの番号がついた N 個のリングがどの 2 つもつながれていない状態で存在することを表す．
- `Link(A, B)` — リング A とリング B をつなぐ． A と B は異なり，また互いに直接はつながれていないと仮定してよい． A と B についてのこれ以外の条件は（物理的な制約に由来するものも含めて）存在しない．また，明らかに `Link(A, B)` と `Link(B, A)` は同値である．
- `CountCritical()` — 現在のリングの配置における重大なリングの数を返す．

例 (Example)

最初はどの 2 つもつながれていない $N = 7$ 個のリングを考える．以下は可能な呼び出しの列の例であり，最後の呼び出しの後には上図の状況になる．

呼び出し	返り値
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

小課題 1 [20 点]

- $N \leq 5\,000$.
- 関数 `CountCritical` は、他のすべての関数呼び出しの後に 1 回だけ呼ばれる .
関数 `Link` は高々 5 000 回呼ばれる .

小課題 2 [17 点]

- $N \leq 1\,000\,000$.
- 関数 `CountCritical` は、他のすべての関数呼び出しの後に 1 回だけ呼ばれる .
関数 `Link` は高々 1 000 000 回呼ばれる .

小課題 3 [18 点]

- $N \leq 20\,000$.
- 関数 `CountCritical` は高々 100 回呼ばれる . 関数 `Link` は高々 10 000 回呼ばれる .

小課題 4 [14 点]

- $N \leq 100\,000$.
- 関数 `CountCritical` と `Link` は合計で高々 100 000 回呼ばれる .

小課題 5 [31 点]

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

- 関数 `CountCritical` と `Link` は合計で高々 1 000 000 回呼ばれる．

実装の詳細 (Implementation details)

あなたは `rings.c`, `rings.cpp` または `rings.pas` という名前のファイルをちょうど 1 つ提出しなければならない．そのファイルは上記で説明されたサブプログラムを以下のシグネチャを用いて実装しなければならない．

C/C++ プログラム

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

Pascal プログラム

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

これらのサブプログラムは上記で説明された通りに動作しなければならない．もちろん，内部での使用のために他のサブプログラムを実装することは自由である．あなたの提出は標準入力・標準出力，あるいは他のファイルといかなる方法でもやりとりしてはならない．

採点プログラムのサンプル (Sample grader)

採点プログラムのサンプルは以下の書式の入力を読み込む：

- 1 行目： N, L ．
- 2 行目から $L + 1$ 行目まで：
 - `CountCritical` を呼ぶことを表す -1 ，または，
 - `Link` へのパラメータ A, B ．

採点プログラムのサンプルは `CountCritical` からのすべての返り値を出力する．