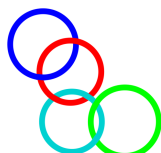


Δακτύλιοι αλεξίπτωτου

Μία πρώτη και αρκετά περίπλοκη έκδοση αυτού που σήμερα ονομάζουμε αλεξίπτωτο περιγράφεται στον *Ατλαντικό Κώδικα* του Λεονάρντο (περίπου 1485). Το αλεξίπτωτο του Λεονάρντο αποτελείτο από ένα κομμάτι πανί που το κρατούσε ανοιχτό μία ξύλινη κατασκευή σε σχήμα πυραμίδας.

Συνδεδεμένοι δακτύλιοι

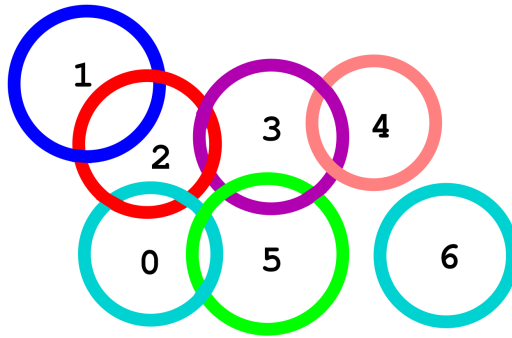
Ο αλεξιπτωτιστής Adrian Nicholas υλοποίησε το σχέδιο του Λεονάρντο περισσότερα από 500 χρόνια αργότερα. Για το σκοπό αυτό, μία μοντέρνα ελαφριά κατασκευή έδενε το αλεξίπτωτο του Λεονάρντο πάνω στο ανθρώπινο σώμα. Θέλουμε να χρησιμοποιήσουμε συνδεδεμένους δακτύλιους για να συγκρατούν το πανί. Κάθε δακτύλιος αποτελείται από εύκαμπτο και ανθεκτικό υλικό. Οι δακτύλιοι μπορούν εύκολα να συνδεθούν μεταξύ τους, καθώς καθένας από αυτούς μπορεί να ανοίξει και να ξανακλείσει. Μία ειδική διάταξη συνδεδεμένων δακτυλίων είναι η "αλυσίδα". Η *αλυσίδα* είναι μια ακολουθία δακτυλίων στην οποία κάθε δακτύλιος συνδέεται μόνο με τους γείτονές του (το πολύ δύο), όπως φαίνεται παρακάτω. Η αλυσίδα αυτή πρέπει να έχει μία αρχή και ένα τέλος (δακτύλιοι που είναι συνδεδεμένοι με το πολύ έναν άλλο δακτύλιο καθένας). Ειδικότερα, ένας μόνος δακτύλιος θεωρούμε ότι αποτελεί αλυσίδα.



Και άλλες διατάξεις είναι προφανώς δυνατές, αφού ένας δακτύλιος μπορεί να συνδέεται και σε τρεις ή περισσότερους άλλους δακτύλιους. Λέμε ότι ένας δακτύλιος είναι *κρίσιμος* αν, όταν αφαιρεθεί, όλοι οι υπόλοιποι δακτύλιοι σχηματίζουν ένα σύνολο από αλυσίδων (ή δεν υπάρχουν άλλοι δακτύλιοι). Με άλλα λόγια, δεν έχει μείνει τίποτα άλλο εκτός από αλυσίδες.

Παράδειγμα

Έστω οι 7 δακτύλιοι του επόμενου σχήματος, αριθμημένοι από 0 έως 6. Υπάρχουν δύο κρίσιμοι δακτύλιοι. Ο ένας είναι ο 2: αν αφαιρεθεί, οι υπόλοιποι δακτύλιοι σχηματίζουν τις αλυσίδες [1], [0, 5, 3, 4] και [6]. Ο άλλος κρίσιμος δακτύλιος είναι ο 3: αν αφαιρεθεί, οι υπόλοιποι δακτύλιοι σχηματίζουν τις αλυσίδες [1, 2, 0, 5], [4] και [6]. Αν αφαιρέσουμε οποιονδήποτε άλλο δακτύλιο, δεν προκύπτει σύνολο ασύνδετων αλυσίδων. Για παράδειγμα, αν αφαιρεθεί ο δακτύλιος: παρότι η [6] είναι αλυσίδα, οι συνδεδεμένοι δακτύλιοι 0, 1, 2, 3 και 4 δε σχηματίζουν αλυσίδα.



Πρόβλημα

Ζητείται να μετρήσετε πόσοι κρίσιμοι δακτύλιοι υπάρχουν στη διάταξη που μεταδίδεται στο πρόγραμμά σας.

Στην αρχή, υπάρχει ένας δεδομένος αριθμός ασύνδετων δακτύλιων. Στη συνέχεια, οι δακτύλιοι συνδέονται μεταξύ τους. Κάθε στιγμή μπορεί να χρειαστεί να επιστρέψετε το πλήθος των κρίσιμων δακτύλιων στην τρέχουσα διάταξη. Συγκεκριμένα, πρέπει να υλοποιήσετε τρεις συναρτήσεις.

- `Init(N)` — καλείται ακριβώς μία φορά, στην αρχή, με σκοπό να δηλώσει ότι υπάρχουν N ασύνδετοι δακτύλιοι αριθμημένοι από 0 έως $N - 1$ (συμπεριλαμβανομένων) στην αρχική διάταξη.
- `Link(A, B)` — οι δύο δακτύλιοι με αριθμούς A και B συνδέονται μεταξύ τους. Θεωρήστε δεδομένο ότι τα A και B είναι διαφορετικά και ότι οι δύο δακτύλιοι δεν είναι ήδη άμεσα συνδεδεμένοι. Εκτός αυτών, δεν υπάρχουν άλλοι περιορισμοί για τα A και B , ιδιαίτερα δεν υπάρχει κανένας φυσικός περιορισμός. Προφανώς, `Link(A, B)` και `Link(B, A)` είναι ισοδύναμα.
- `CountCritical()` — επιστρέφει το πλήθος των κρίσιμων δακτύλιων για την τρέχουσα διάταξη συνδεδεμένων δακτύλιων.

Παράδειγμα

Έστω το σχήμα με τους $n = 7$ δακτύλιους και έστω ότι αρχικά είναι ασύνδετοι. Δείχνουμε μία δυνατή ακολουθία κλήσεων, στο τέλος της οποίας η διάταξη έχει τη μορφή του σχήματος.

Κλήση	Επιστρέφει
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

Υποπρόβλημα 1 [20 βαθμοί]

- $N \leq 5\,000$.
- Η συνάρτηση `CountCritical` καλείται μόνο μία φορά, μετά από όλες τις άλλες κλήσεις. Η συνάρτηση `Link` καλείται το πολύ 5 000 φορές.

Υποπρόβλημα 2 [17 βαθμοί]

- $N \leq 1\,000\,000$.
- Η συνάρτηση `CountCritical` καλείται μόνο μία φορά, μετά από όλες τις άλλες κλήσεις. Η συνάρτηση `Link` καλείται το πολύ 1 000 000 φορές.

Υποπρόβλημα 3 [18 βαθμοί]

- $N \leq 5\,000$.
- Η συνάρτηση `CountCritical` καλείται το πολύ 100 φορές. Η συνάρτηση `Link` καλείται το πολύ 10 000 φορές.

Υποπρόβλημα 4 [14 βαθμοί]

- $N \leq 100\,000$.
- Οι συναρτήσεις `CountCritical` και `Link` καλούνται, συνολικά, το πολύ 100 000 φορές.

Υποπρόβλημα 5 [31 βαθμοί]

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

- Οι συναρτήσεις `CountCritical` και `Link` καλούνται, συνολικά, το πολύ 1 000 000 φορές.

Λεπτομέρειες υλοποίησης

Πρέπει να υποβάλετε ακριβώς ένα αρχείο, με όνομα `rings.c`, `rings.cpp` ή `rings.pas`. Αυτό το αρχείο πρέπει να υλοποιεί τις συναρτήσεις που περιγράφηκαν παραπάνω, με τις ακόλουθες επικεφαλίδες.

Προγράμματα C/C++

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

Προγράμματα Pascal

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

Αυτές οι συναρτήσεις πρέπει να συμπεριφέρονται όπως αναφέρεται παραπάνω. Φυσικά είστε ελεύθεροι να υλοποιήσετε και άλλες συναρτήσεις για εσωτερική χρήση. Οι υποβολές σας δεν πρέπει να αλληλεπιδρούν με κανέναν τρόπο με το `standard input/output`, ούτε με κανένα άλλο αρχείο.

Ενδεικτικός βαθμολογητής

Ο ενδεικτικός βαθμολογητής διαβάζει την είσοδο με την εξής μορφή:

- γραμμή 1: `N, L`;
- γραμμές 2, ..., `L + 1`:
 - -1 για κλήση της `CountCritical`;
 - `A, B` οι παράμετροι της `Link`.

Ο ενδεικτικός βαθμολογητής εκτυπώνει όλα τα αποτελέσματα που επιστρέφει η `CountCritical`.