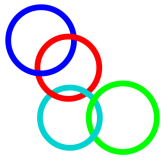
Parachute rings

לאונרדו תכנן מצנח. המצנח היה מורכב ממבנה מיוחד מעץ ומבד.

טבעות מחוברות

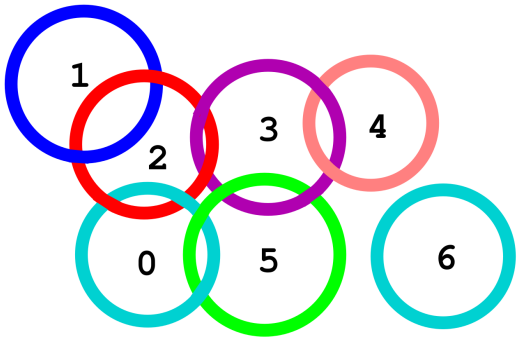
מנסים לבנות את המצנח שלאונרדו תכנן מחלקים שכל אחד מהם הוא טבעת (Ring). אפשר לקחת שתי טבעות ולחבר אותן אחת לשנייה. כך נוצרת קונפיגורציה של טבעות. יש סוג מיוחד של קונפיגורציה שנקרא "שרשרת" (Chain). שרשרת מורכבת מסדרה של טבעות, שכל אחת מהן מחוברת לשתי טבעות בלבד (הבאה והקודמת בשרשרת), פרט לטבעת הראשונה ולטבעת האחרונה בשרשרת, שמחוברות כל-אחת רק לטבעת אחת. טבעת בודדת, שלא מחוברת לכלום, היא גם שרשרת.



קיימות גם קונפיגורציות של טבעות שאינן בצורת שרשרת. למשל, כל קונפיגורציה שבה יש טבעת שמחוברת לפחות לשלוש טבעות אחרות, אינה שרשרת. טבעת תיקרא "קריטית" (Critical), אם כשמוחקים אותה, מתקבלת קונפיגורציה שהיא אוסף של שרשראות זרות.

דוגמא

באיור הבא מופיעות 7 טבעות, הממוספרות מ- 0 עד 6. יש שתי טבעות קריטיות. טבעת קריטית אחת היא טבעת מספר 2: אחרי שמוחקים אותה, הטבעות האחרות יוצרות את השרשראות [1], [0,5,3,4] ו- [6]. הטבעת הקריטית השנייה היא טבעת מספר 3: אחרי שמוחקים אותה, הטבעות האחרות יוצרות את השרשראות [1,2,0,5], [4] ו- [6]. כל טבעת אחרת שנמחק, תשאיר קונפיגורציה שאינה קבוצה של שרשראות זרות. למשל, אם נמחק את טבעת מספר 5, אמנם נקבל את השרשרת [6], אבל הטבעות הקשורות 0,1,2,3,4 לא מהוות שרשרת (טבעת 2 מחוברת ל - 3 טבעות אחרות).



תיאור הבעייה

עליכם לכתוב תוכנית שתדע לחשב את מספר הטבעות הקריטיות בקונפיגורציות, בשלבים שונים תוך כדי בנייתן.

בהתחלה, יש אוסף של טבעות, שאף אחת מהן לא מחוברת לאף אחת אחרת. אחר-כך, מתחילים לחבר טבעות אחת לשנייה. מדי פעם, בין פעולות חיבור טבעות, תתבקשו לחשב את מספר הטבעות הקריטיות שקיימות כרגע בקונפיגורציה. כלומר, עליכם לממש את שלושת הפונקציות הבאות.

- `Init`(N)

המערכת תקרא לפונקציה הזאת פעם אחת, בהתחלה. המספר N הוא מספר הטבעות. בהתחלה, אין אף חיבור ביניהן. הטבעות ממוספרות מ- 0 עד ל- N - 1 (כולל).

- `Link`(A, B)

המערכת קוראת לפקודה הזאת כדי לחבר את הטבעת A לטבעת B. מובטח ש- A ו- B הן שתי טבעות שונות, ושהן לא מחוברות באופן ישיר לפני הקריאה לפונקציה.

- `CountCritical`()

כשהמערכת קוראת לפונקציה הזאת, הפונקציה מחשבת ומחזירה את מספר הטבעות הקריטיות בקונפיגורציה הנוכחית.

Example

Consider our figure with N = 7 rings and suppose that they are initially unlinked. We show a possible sequence of calls, where after the last call we obtain the situation depicted in our figure.

Returns	Call
	`Init`(7)
7	`CountCritical`()
	`Link`(1, 2)
7	`CountCritical`()
	`Link`(0, 5)
7	`CountCritical`()
	`Link`(2, 0)
7	`CountCritical`()
	`Link`(3, 2)
4	`CountCritical`()
	`Link`(3, 5)
3	`CountCritical`()
	`Link`(4, 3)
2	`CountCritical`()

[Subtask 1 [20 points

- $N \leq 5\,000$.
- The function `CountCritical` is called only once, after all the other calls; the function `Link` is called at most 5 000 times.

[Subtask 2 [17 points

- $N \leq 1\,000\,000$.
- The function `CountCritical` is called only once, after all the other calls; the function `Link` is called at most 1 000 000 times.

[Subtask 3 [18 points

- $N \leq 20\,000$.
- The function `CountCritical` is called at most 100 times; the function `Link` is called at most 10 000 times.

[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
- ;to invoke `CountCritical` 1-
- .A, B parameters to `Link`

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