

D. Recensământ (census)

Limită de timp: 1 secunde

Limită de memorie: 128 MiB

Un fapt mai puțin cunoscut despre Cesenatico este că găzduiește o societate secretă de N informaticieni. Această societate este într-adevăr foarte secretă; niciun membru nu știe cine sunt ceilalți membri. Fiecare membru are un identificator unic (ID), reprezentat printr-un număr natural I .

Singura comunicare între membri este indirectă, prin numere scrise cu cretă în diferite locații din oraș. La fiecare 100 de ani, societatea efectuează un recensământ pentru a-și număra membrii. După finalizarea recensământului, fiecare membru ar trebui să cunoască numărul total de membri ai societății.

Recensământul are loc pe parcursul mai multor zile. În fiecare zi, fiecare membru care încă participă la proces va alege și va efectua exact o acțiune: să citească (operația **read**), să scrie (operația **write**) sau să se oprească (operația **stop**) din participare.

- Dacă un membru alege să citească (operația **read**), acesta va selecta o locație P . În timpul zilei, vizitează locația P și citește numărul scris acolo.
- Dacă un membru alege să scrie (operația **write**), acesta va selecta o locație P și un număr V . Seara târziu, vizitează locația P și schimbă numărul scris acolo cu V . Deoarece este deja întuneric, nu poate citi numărul vechi înainte de a-l scrie pe cel nou.
- Dacă un membru alege să se oprească (operația **stop**), nu mai întreprinde nicio acțiune în zilele următoare.

Dacă un membru vede un alt membru scriind un număr, ar putea o recunoaște. Prin urmare, este strict interzis ca doi sau mai mulți membri să aleagă să scrie în aceeași locație în aceeași zi. (Nu există o astfel de restricție pentru citire, deoarece aceasta se poate face discret.)

Dacă unul sau mai mulți membri citesc dintr-o locație unde un alt membru dorește să scrie în aceeași zi, toate citirile au loc înaintea scrierii.

Cum ar trebui ca societatea să-și planifice recensământul pentru a minimiza numărul de zile până când toată lumea află numărul corect de membri?

Implementare

⇒ Aceasta este o problemă interactivă, în care un număr necunoscut de instanțe ($1 \leq N \leq 100$) ale programului tău vor fi executate simultan. Fiecare instanță simulează un membru al societății.

Există 10^{18} locații. Numărul P al unei locații trebuie să satisfacă $0 \leq P < 10^{18}$. Inițial, valoarea scrisă în toate locațiile este $V = 0$.

Noua valoare V scrisă într-o locație trebuie să fie întotdeauna un număr întreg astfel încât $0 \leq V \leq 10^9$. În majoritatea subtask-urilor, V poate fi doar 0 sau 1. Vezi secțiunea de punctare pentru mai multe detalii.

Când pornește o instanță a programului tău, ar trebui mai întâi să citească o linie cu două numere întregi, I și M ($0 \leq I \leq M - 1$): ID-ul unic al membrului societății reprezentat de această instanță și numărul total de ID-uri posibile. În cadrul fiecărui test, toate instanțele vor primi aceeași valoare M și valori distincte I . Reține că pot exista ID-uri care nu sunt alocate niciunui membru.

Atunci, pentru fiecare zi din procesul de recensământ, programul tău ar trebui să aleagă acțiunea pe care dorește să o efectueze și să afișeze o linie corespunzătoare:

Acțiune	Semnificație
$r \ P$	Citește (read) locația P . După afișarea acestei linii, programul tău ar trebui să citească o linie cu valoarea curentă scrisă la P .
$w \ P \ V$	Scrie (write) la locația P noua valoare V . Dacă mai multe instanțe scriu la aceeași locație P în aceeași zi, vei primi verdictul <i>Not correct</i> . Cu excepția exemplelor și a subtask-ului 3, trebuie să scrii $0 \leq V \leq 1$; vezi secțiunea de punctare.
$! \ N$	Răspunde și oprește-te (stop): raportează că există N membri și încetează să mai participi la recensământ. După ce răspunzi, programul tău ar trebui să se încheie normal . (Reține că alte instanțe ale programului tău pot continua să ruleze pentru zile suplimentare înainte de a răspunde și de a se încheia.)

Dacă vreo instanță a programului tău răspunde cu o valoare greșită a lui N , încalcă protocolul, folosește mai mult de 500 de zile sau depășește limita de timp/memorie (per proces), soluția ta va primi verdictul *Not correct* pe testul respectiv.

În caz contrar, programul tău va fi considerat (*Partially*) *Correct* pe testul respectiv și punctat pe baza unei valori D reprezentând numărul maxim de zile în care orice instanță a dat un răspuns. Pentru punctajul maxim, trebuie ca $D \leq 61$ și $V \leq 1$. Vezi secțiunea de punctare pentru detalii.

Golirea bufferului (Flushing). Dacă nu folosești template-urile furnizate, asigură-te că golești bufferul de ieșire standard după ce afișezi fiecare linie, altfel programul tău ar putea fi considerat *Not correct*. În Python, acest lucru se întâmplă automat dacă folosești `input()` pentru a citi linii. În C++, `cout << endl`; golește bufferul pe lângă afișarea unui rând nou; respectiv `fflush(stdout)` dacă folosești `printf`.

Constrângeri

- $1 \leq N \leq 100$.
- $1 \leq M \leq 100\,000$.
- Poți folosi cel mult 500 de zile.

Punctaj

Programul tău va fi testat pe mai multe teste grupate în subtask-uri. Pentru a obține punctajul pentru un subtask, trebuie să rezolvi corect toate testele pe care le conține.

- **Subtask-ul 0 [0 puncte]:** Exemple (poți scrie orice număr întreg $0 \leq V \leq 1\,000\,000\,000$).
- **Subtask-ul 1 [11 puncte]:** $M \leq 100$, iar cei N membri au ID-urile $0, 1, \dots, N - 1$.
- **Subtask-ul 2 [12 puncte]:** $1 \leq N \leq 2$.
- **Subtask-ul 3 [22 puncte]:** $M \leq 8000$, și poți scrie orice număr întreg $0 \leq V \leq 1\,000\,000\,000$.
- **Subtask-ul 4 [55 puncte]:** Fără restricții suplimentare.

În subtask-urile 1, 2 și 4, poți scrie doar $V = 0$ sau $V = 1$ în fiecare acțiune de scriere.

Fie X_s punctajul subtask-ului s (vezi mai sus), și D_s cel mai mare număr de zile pe care o instanță a soluției tale îl folosește pe un test din subtask-ul s . Atunci:

$$\text{score}_s = \begin{cases} X_s & \text{dacă } D_s \leq 61 \\ X_s \cdot (0.2 + 0.8 \cdot 1.01^{(60-D_s)}) & \text{dacă } 61 < D_s \leq 500 \\ 0 & \text{dacă } 500 < D_s. \end{cases}$$

Valoarea score_s este rotunjită la cel mai apropiat întreg pentru fiecare subtask, iar punctajul total este suma acestora. O soluție care obține 100 de puncte, trebuie să respecte $D \leq 61$ și $V \leq 1$ pe fiecare test.

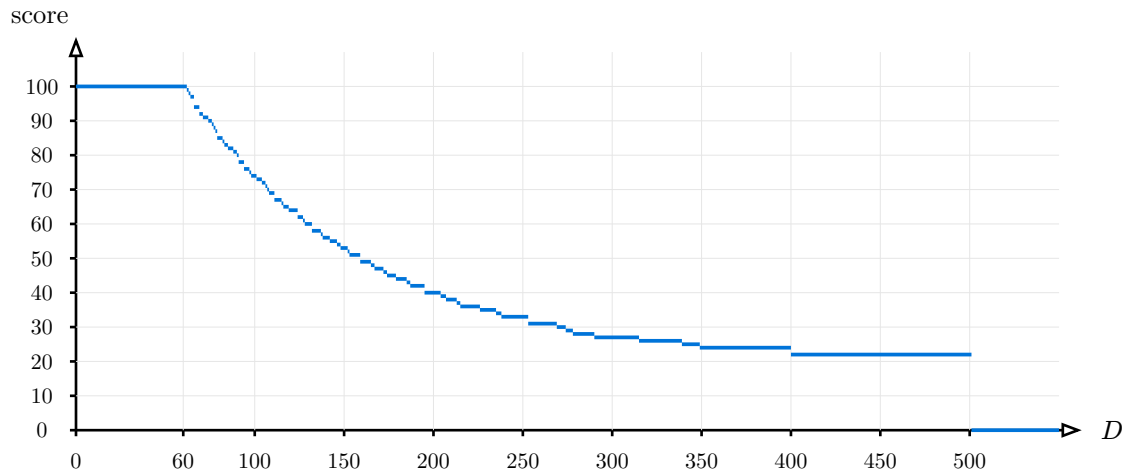


Figura 1: Punctajul total, presupunând că fiecare subtask este rezolvat cu același maxim D .

Exemple de intrare/ieșire

Primul exemplu. Fiecare pereche de coloane reprezintă comunicarea dintre o instanță și grader.

Gra.	Inst. 0	Gra.	Inst. 1	Gra.	Inst. 2	Gra.	Inst. 3	Gra.	Inst. 4
0 100		1 100		2 100		3 100		4 100	
	w 12 1		w 50 1		w 99 0		w 7 1		r 5
								0	
	r 50		r 7		r 12		w 1 1		! 5
1		1		1					
	! 5		r 1		w 0 0		! 5		
		1							
		! 5			! 5				

Al doilea exemplu.

Grader	Instanță 0
0 8000	
	w 0 0
	w 1 1
	r 2
1	
	! 2

Grader	Instanță 1
3 8000	
	w 2 1
	r 1
0	
	r 2
1	
	r 1
1	
	! 2

Explicație

Primul Exemplu. Avem $N = 5$ membri cu ID-uri consecutive 0, 1, 2, 3, 4 și $M = 100$ (valid pentru subtask-urile 1, 3 și 4). Instanța i corespunde membrului cu ID-ul i . Interacțiunea de mai sus este doar o secvență posibilă și legală de operații și **nu** este menită să fie o strategie eficientă sau rezonabilă; este arătată doar pentru a ilustra cum funcționează protocolul.

Al Doilea Exemplu. Avem $N = 2$ membri, cu ID-urile 0 și 3, și $M = 8000$ (valid pentru subtask-urile 2, 3 și 4). În prima zi, membrul cu ID-ul 0 scrie un 0 la locația 0 (fără schimbare), iar membrul cu ID-ul 3 scrie un 1 la locația 2.

location	0	1	2	3	4	...
value	0	0	1	0	0	...

În a doua zi, ID-ul 0 scrie un 1 la locația 1, iar ID-ul 3 citește aceeași locație. Reține că citirea are loc în timpul zilei, înainte de scrierea din seară. Prin urmare, ID-ul 3 vede în continuare un 0.

location	0	1	2	3	4	...
value	0	1	1	0	0	...

În a treia zi, ambii citesc locația 2, unde este scris un 1.

În a patra zi, ID-ul 0 răspunde că sunt 2 membri (corect), în timp ce ID-ul 3 citește 1-ul de la locația 1. ID-ul 0 iese imediat după aceasta și nu mai participă în zilele următoare.

În final, în ziua $D = 5$, membrul rămas răspunde și el corect $N = 2$.

Testare

Pentru a facilita testarea soluției tale, oferim un instrument simplu pe care îl poți descărca de pe CMS. Instrumentul este opțional. Reține că evaluatorul oficial de pe CMS este diferit de instrumentul de testare.

Pentru a folosi instrumentul, ai nevoie de un fișier de intrare. Poți folosi exemplele de intrare furnizate `census.input0.txt` și `census.input1.txt`, sau să-ți creezi propriile fișiere. Fișierul de intrare ar trebui să înceapă cu numărul de membri N și ID-urile posibile M , urmat de o linie cu N numere care specifică ID-urile membrilor societății.

Pentru programele Python, de exemplu `census.py` (de obicei rulat ca `pypy3 census.py`) rulează instrumentul de testare astfel:

```
python3 testing_tool.py pypy3 census.py < census.input0.txt
```

Pentru programele C++, mai întâi compilează soluția:

```
g++ -DEVAL -std=gnu++20 -O2 -pipe -static -s -o census census.cpp
```

apoi rulează instrumentul de testare:

```
python3 testing_tool.py ./census < census.input0.txt
```

Observă că în această problemă, ieșirea standard este folosită pentru comunicarea cu grader-ul, prin urmare nu are trebui să fie folosită pentru depănarea codului. În schimb, puteți folosi standard error output (stderr). În C++ se poate folosi `cerr << msg << endl;`, iar în Python, `print(msg, file=sys.stderr)`.

Programul `testing_tool` va citi și afișa orice mesaj trimis către `stderr` împreună cu comunicarea dintre instanțe și grader. Din motive tehnice, cele două buffere de afișare (stdout și stderr) pot să apară desincronizate.