

D. Popis (census)

Vremensko ograničenje: 1 sekundi

Memorijsko ograničenje: 128 MiB

Manje poznata činjenica o Cesenaticu je da je on dom tajnom društvu od N informatičarki. Ovo društvo je uistinu vrlo tajno; nijedan član ne poznaje nikoga drugog. Svaki član ima jedinstveni ID: nenegativni cijeli broj I .

Jedina komunikacija među članovima je neizravna, putem brojeva ispisanih kredom na različitim lokacijama po gradu. Svaki 100 godina društvo provodi popis kako bi prebrojalo svoje članove. Nakon što je popis gotov, svaki član bi trebao znati ukupan broj članova u društvu.

Popis se odvija tijekom više dana. Svakog dana, svaki član koji još uvijek sudjeluje u procesu odabrat će i izvršiti točno jednu radnju: **čitati**, **pisati** ili **prestati** sudjelovati.

- Ako član odabere **čitati**, odabire lokaciju P . Tijekom dana posjećuje lokaciju P i čita broj koji je tamo napisan.
- Ako član odabere **pisati**, odabire lokaciju P i broj V . Kasno navečer posjećuje lokaciju P i mijenja broj koji je tamo bio napisan u V . Budući da je već mrak, ne može pročitati stari broj prije nego što upiše novi.
- Ako član odabere **prestati**, više ne poduzima nikakve radnje narednih dana.

Ako jedan član vidi drugog kako piše broj, možda ga prepozna. Stoga je strogo zabranjeno da dva ili više članova odaberu pisati na istoj lokaciji istog dana. (Ne postoji takvo ograničenje za čitanje, budući da se to može učiniti diskretno.)

Ako jedan ili više članova čitaju s lokacije na kojoj drugi član želi pisati istog dana, sva čitanja se događaju prije pisanja.

Kako društvo treba planirati svoj proces popisa kako bi minimiziralo broj dana dok svatko ne sazna točan broj članova?

Implementacija

⇒ Ovo je interaktivni problem u kojem će se nepoznat broj instanci ($1 \leq N \leq 100$) vašeg programa izvršavati istovremeno. Svaka instanca simulira jednog člana društva.

Postoji 10^{18} lokacija. Broj P lokacije mora zadovoljavati $0 \leq P < 10^{18}$. Početno, vrijednost napisana na svim lokacijama je $V = 0$.

Nova vrijednost V zapisana na lokaciji mora uvijek biti cijeli broj takav da $0 \leq V \leq 10^9$. U većini podzadataka, V može biti samo 0 ili 1. Pogledajte odjeljak Bodovanje za više detalja.

Kada se instanca vašeg programa pokrene, prvo treba pročitati liniju s dva cijela broja, I i M ($0 \leq I \leq M - 1$): jedinstveni ID člana društva kojeg predstavlja ova instanca i ukupan broj mogućih ID-ova. Unutar svakog test slučaja, sve instance će dobiti istu vrijednost M i različite vrijednosti I . Napomena: mogu postojati ID-ovi koji nisu dodijeljeni nijednom članu.

Zatim, za svaki dan u procesu popisa, vaš program treba odabrati radnju koju želi izvršiti i ispisati odgovarajuću liniju:

Akcija	Značenje
$r\ P$	Čitaj lokaciju P . Nakon ispisa ovog retka, vaš program treba pročitati liniju s trenutnom vrijednošću napisanom na P .
$w\ P\ V$	Piši na lokaciju P novu vrijednost V . Ako više instanci piše na istom P istog dana, dobit ćete presudu <i>wrong answer</i> . Osim za primjere i podzadatak 3, morate pisati $0 \leq V \leq 1$; pogledajte odjeljak Bodovanje.
$!\ N$	Odgovori i prestani: prijavite da ima N članova i prestanite sudjelovati u popisu. Nakon odgovora, vaš program treba normalno završiti . (Napomena: druge instance vašeg programa mogu nastaviti raditi dodatne dane prije nego što odgovore i završe.)

Ako bilo koja instanca vašeg programa odgovori pogrešnu vrijednost N , prekrši protokol, iskoristi više od 500 dana ili premaši (po procesu) vremensko/memorijsko ograničenje, vaša prijava će biti ocijenjena kao *wrong answer* za dani test slučaj.

Inače, vaš program će biti (*djelomično*) *točan* na test slučaju i bodovan na temelju vrijednosti D : maksimalnog broja dana koji je bilo kojoj instanci trebalo za odgovor. Za puni broj bodova morate riješiti svaki test slučaj s $D \leq 61$ i $V \leq 1$. Pogledajte odjeljak Bodovanje za detalje.

Flushanje. Ako ne koristite ponudene predloške, pobrinite se da izvršite `flush` standardnog izlaza nakon ispisa svakog retka, inače bi vaš program mogao biti ocijenjen kao *wrong answer*. U Pythonu, to se događa automatski ako koristite `input()` za čitanje linija. U C++, `cout << endl`; flusha uz ispis znaka za novi red; ako koristite `printf`, koristite `fflush(stdout)`.

Ograničenja

- $1 \leq N \leq 100$.
- $1 \leq M \leq 100\,000$.
- Možete iskoristiti najviše 500 dana.

Bodovanje

Vaš program će se testirati na nekoliko test slučajeva grupiranih u podzadatke. Da biste dobili bodove za podzadatak, morate točno riješiti sve testove koje on sadrži.

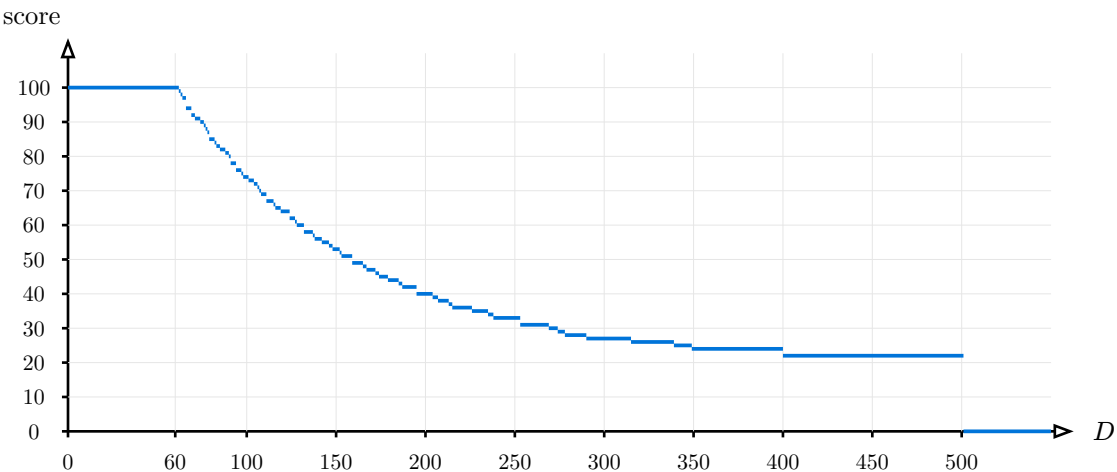
- Podzadatak 0 [0 bodova]:** Primjeri (možete pisati bilo koji cijeli broj $0 \leq V \leq 1\,000\,000\,000$).
- Podzadatak 1 [11 bodova]:** $M \leq 100$, a N članova ima ID-ove $0, 1, \dots, N - 1$.
- Podzadatak 2 [12 bodova]:** $1 \leq N \leq 2$.
- Podzadatak 3 [22 bodova]:** $M \leq 8000$, i možete pisati bilo koji cijeli broj $0 \leq V \leq 1\,000\,000\,000$.
- Podzadatak 4 [55 bodova]:** Nema dodatnih ograničenja.

U podzadacima 1, 2 i 4, možete pisati samo $V = 0$ ili $V = 1$ u svakoj akciji pisanja (Write).

Neka je X_s maksimalan broj bodova za podzadatak s (prikazan iznad), a D_s najveći broj dana koji bilo koji od vaših programa koristi na testu u podzadatku s . Tada:

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

Vrijednost score_s se zaokružuje na najbliži cijeli broj po podzadatku, a vaš ukupni rezultat je zbroj tih bodova. Da biste dobili puni rezultat za zadatak, trebate $D \leq 61$ i $V \leq 1$ na svakom test slučaju.



Slika 1: Ukupan rezultat, pod pretpostavkom da je svaki podzadatak riješen s istim maksimalnim D .

Primjeri

Prvi primjer. Svaki par stupaca prikazuje komunikaciju između ocjenjivača i jedne instance.

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				

Drugi primjer.

Ocjenjivač	Instanca 0
0 8000	
	w 0 0
	w 1 1
	r 2
1	
	! 2

Ocjenjivač	Instanca 1
3 8000	
	w 2 1
	r 1
0	
	r 2
1	
	r 1
1	
	! 2

Objašnjenje

Prvi primjer. Imamo $N = 5$ članova s uzastopnim ID-ovima 0, 1, 2, 3, 4 i $M = 100$ (vrijedi za podzadatke 1, 3 i 4). Instanca i odgovara članu s ID-om i . Interakcija iznad je samo jedan mogući legalni niz operacija i **nije** zamišljena kao učinkovita ili razumna strategija; prikazana je samo da ilustrira kako protokol radi.

Drugi primjer. Imamo $N = 2$ člana, s ID-ovima 0 i 3, i $M = 8000$ (vrijedi za podzadatke 2, 3 i 4). Prvog dana, član s ID-om 0 zapisuje 0 na lokaciju 0 (nema promjene), a član s ID-om 3 zapisuje 1 na lokaciju 2.

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

Drugog dana, ID 0 zapisuje 1 na lokaciju 1, a ID 3 čita tu istu lokaciju. Primijetite da se čitanje događa tijekom dana, prije pisanja navečer. Stoga, ID 3 još uvijek vidi 0.

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

Trećeg dana, oboje čitaju lokaciju 2, gdje je zapisana 1.

Četvrtog dana, ID 0 odgovara da postoje 2 člana (točno), dok ID 3 čita 1 na lokaciji 1. ID 0 odmah nakon ovoga završava i ne sudjeluje u nadolazećim danima.

Konačno, na dan $D = 5$, preostali član također točno odgovara $N = 2$.

Testiranje

Kako bismo olakšali testiranje vašeg rješenja, pružamo jednostavan alat koji možete preuzeti s CMS-a. Korištenje alata je opcionalno. Napomena: službeni ocjenjivač na CMS-u se razlikuje od alata za testiranje.

Za korištenje alata trebate ulaznu datoteku. Možete koristiti priložene primjere ulaza `census.input0.txt` i `census.input1.txt`, ili napraviti svoje. Ulazna datoteka treba započeti s brojem članova N i mogućim ID-ovima M , nakon čega slijedi linija s N brojeva koji određuju ID-ove članova društva.

Za Python programe, recimo `census.py` (obično se pokreće kao `pypy3 census.py`) pokrenite alat za testiranje na sljedeći način:

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

For C++ programs, first compile your solution:

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

a zatim pokrenite alat za testiranje:

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

Napomena: u ovom problemu standardni izlaz se koristi za komunikaciju s ocjenjivačem, stoga se ne bi trebao koristiti za debugiranje. Umjesto toga, možete koristiti izlaz za pogreške (stderr). U C++-u možete koristiti `cerr << msg << endl;`. U Pythonu možete koristiti `print(msg, file=sys.stderr)`.

Alat za testiranje će čitati i prikazati ove stderr poruke zajedno s upitima koje su izvršile sve instance vašeg programa. Napomena: iz tehničkih razloga, možda će se činiti blago izvan sinkronizacije jedna s drugom.