

D. Popis (census)

Vremensko ograničenje: 1 sekunda

Memorijsko ograničenje: 128 MiB

Manje poznata činjenica o Cesenaticu je da je dom tajnom društvu od N informatičarki. Ovo društvo je zaista vrlo tajno; nijedna članica ne poznaje nijednu drugu. Svaka članica ima jedinstven ID: nenegativan cijeli broj I .

Jedina komunikacija među članicama je indirektna, putem brojeva ispisanih kredom na različitim lokacijama širom grada. Svakih 100 godina, društvo vrši popis kako bi prebrojalo svoje članice. Nakon što se popis završi, svaka članica treba znati ukupan broj članica u društvu.

Popis se odvija tokom više dana. Svakog dana, svaka članica koja još uvijek učestvuje u procesu će odabrati i izvršiti tačno jednu akciju: **čitati**, **pisati** ili **prestat** učestvovati u njemu.

- Ako članica odabere **čitati**, ona bira lokaciju P . Tokom dana posjeti lokaciju P i pročita broj koji je tamo ispisan.
- Ako članica odabere **pisati**, ona bira lokaciju P i broj V . Kasno navečer, ona posjeti lokaciju P i mijenja broj koji je tamo bio ispisan u V . Budući da je već mrak, ona ne može pročitati stari broj prije nego što upiše novi.
- Ako članica odabere **prestat**, ona više ne preduzima nikakve akcije u narednim danima.

Ako jedna članica vidi drugu kako piše broj, mogla bi je prepoznati. Stoga je strogo zabranjeno da dvije ili više članica odaberu pisati na istoj lokaciji istog dana. (Ne postoji takvo ograničenje za čitanje, jer se to može uraditi diskretno.)

Ako jedna ili više članica čitaju sa lokacije gdje druga članica želi pisati istog dana, sva čitanja se dešavaju prije pisanja.

Kako bi društvo trebalo planirati svoj proces popisa kako bi se minimalizirao broj dana dok svi ne saznaju tačan broj članica?

Implementacija

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

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

Nova vrijednost V upisana 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 sekciju Bodovanje za više detalja.

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

Zatim, za svaki dan u procesu popisa, vaš program treba odabrati akciju koju želi izvršiti i ispisati liniju u skladu s tim:

Akcija	Značenje
$r\ P$	Pročitaj lokaciju P . Nakon štampanja ove linije, vaš program treba pročitati liniju sa trenutnom vrijednošću upisanom na P .
$w\ P\ V$	Zapiši na lokaciji P novu vrijednost V . Ako više instanci piše na istoj lokaciji P istog dana, dobićete rezultat <i>Nije tačno</i> (Not correct). Osim za primjere i podzadatak 3, morate pisati $0 \leq V \leq 1$; pogledajte sekciju Bodovanje.
$!\ N$	Odgovori i stani: prijavi da postoji N članica i prestani učestvovati u popisu. Nakon odgovaranja, vaš program treba normalno završiti . (Imajte na umu da druge instance vašeg programa mogu nastaviti raditi još nekoliko dana prije nego što odgovore i završe.)

Ako bilo koja instanca vašeg programa odgovori pogrešnom vrijednošću N , prekrši protokol, potroši više od 500 dana ili prekorači (po procesu) vremensko/memorijsko ograničenje, vaša prijava će biti ocijenjena kao *Nije tačno* (Not correct) za dati test slučaj.

U suprotnom, vaš program će biti (*Djelimično*) *Tačan* na test slučaju i bodovan na osnovu vrijednosti D : maksimalnog broja dana koji je bilo kojoj instanci trebalo da odgovori. Za maksimalan broj bodova, morate riješiti svaki test slučaj sa $D \leq 61$ i $V \leq 1$. Pogledajte sekciju Bodovanje za detalje.

Ispiranje (Flushing). Ako ne koristite ponuđene templejte, provjerite jeste li isprali (flush) standardni izlaz nakon štampanja svake linije, inače bi vaš program mogao biti ocijenjen kao *Nije tačno* (Not correct). U Pythonu se ovo dešava automatski ako koristite `input()` za čitanje linija. U C++, `cout << endl`; ispire pored štampanja novog reda; ako koristite `printf`, koristite `fflush(stdout)`.

Ograničenja

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

Bodovanje

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

- **Podzadatak 0 [0 bodovi]:** Primjeri (možete upisati bilo koji cijeli broj $0 \leq V \leq 1\,000\,000\,000$).
- **Podzadatak 1 [11 bodovi]:** $M \leq 100$, i N članica imaju ID-jeve $0, 1, \dots, N - 1$.
- **Podzadatak 2 [12 bodovi]:** $1 \leq N \leq 2$.
- **Podzadatak 3 [22 bodovi]:** $M \leq 8000$, i možete upisati bilo koji cijeli broj $0 \leq V \leq 1\,000\,000\,000$.
- **Podzadatak 4 [55 bodovi]:** Bez 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 vaš program potroši na testu u podzadatku s . Tada:

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

Vrijednost rezultat_s se zaokružuje na najbliži cijeli broj po podzadatku, a vaš ukupan rezultat je suma ovih vrijednosti. Da biste dobili puni broj bodova za zadatak, potrebno je $D \leq 61$ i $V \leq 1$ na svakom test slučaju.

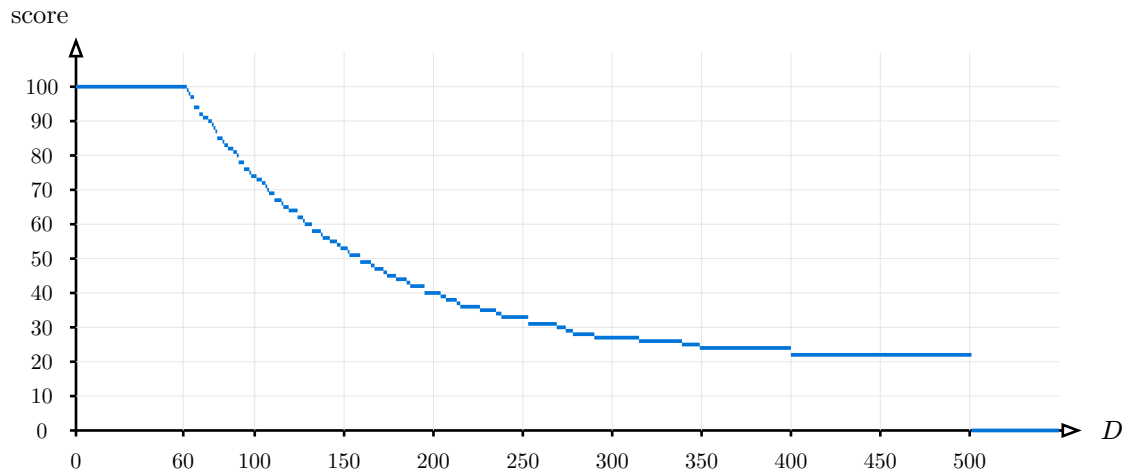


Figure 1: Ukupan rezultat, pod pretpostavkom da je svaki podzadatak riješen sa istim maksimalnim D .

Primjeri

Prvi primjer. Svaki par kolona prikazuje komunikaciju između gradera 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.

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

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

Objašnjenje

Prvi primjer. Imamo $N = 5$ članica sa uzastopnim ID-jevima 0, 1, 2, 3, 4 i $M = 100$ (važeće za podzadatke 1, 3 i 4). Instanca i odgovara članici sa ID-jem i . Interakcija iznad je samo jedan mogući legalni niz operacija i **nije** zamišljena kao efikasna ili razumna strategija; prikazana je samo da ilustruje kako protokol radi.

Drugi primjer. Imamo $N = 2$ članice, sa ID-jevima 0 i 3, i $M = 8000$ (važeće za podzadatke 2, 3 i 4). Prvog dana, članica sa ID-jem 0 piše 0 na lokaciji 0 (nema promjene), a članica sa ID-jem 3 piše 1 na lokaciji 2.

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

Drugog dana, ID 0 piše 1 na lokaciji 1, a ID 3 čita tu istu lokaciju. Napomena: čitanje se dešava tokom dana, prije pisanja navečer. Stoga, ID 3 i dalje vidi 0.

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

Trećeg dana, obje čitaju lokaciju 2, gdje je upisana jedinica.

Četvrtog dana, ID 0 odgovara da postoje 2 članice (tačno), dok ID 3 čita 1 na lokaciji 1. ID 0 odmah izlazi nakon ovoga i ne učestvuje u narednim danima.

Konačno, na dan $D = 5$, preostala članica također tačno odgovara $N = 2$.

Testiranje

Kako bismo olakšali testiranje vašeg rješenja, pružamo jednostavan alat koji možete preuzeti sa CMS-a. Korištenje alata je opcionalno. Napominjemo da se zvanični grader na CMS-u razlikuje od alata za testiranje.

Da biste koristili alat, potreban vam je ulazni fajl. Možete koristiti priložene primjere ulaza `census.input0.txt` i `census.input1.txt`, ili napraviti svoj. Ulazni fajl treba započeti brojem članica N i mogućim ID-jevima M , praćeno linijom sa N brojeva koji specificiraju ID-jeve članica 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
```

Za C++ programe, prvo kompilirajte svoje rješenje:

```
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 sa graderom, pa se ne smije koristiti za debugovanje. Umjesto toga, možete koristiti standardni izlaz za greške (stderr). U C++ možete koristiti `cerr << msg << endl;`. U Pythonu možete koristiti `print(msg, file=sys.stderr)`.

Alat za testiranje će pročitati i prikazati ove stderr poruke zajedno sa upitima koje su izvršile sve instance vašeg programa. Imajte na umu da se iz tehničkih razloga mogu pojaviti blago nesinhronizovane jedna sa drugom.