

D. Попис (census)

Временско ограничење: 1 секунда

Меморијско ограничење: 128 MiB

Мање позната чињеница о Чезенатику је да је он дом тајног друштва од N информатичарки. Ово друштво је заиста веома тајно; ниједна чланица не зна идентитет ниједне друге. Свака чланица има јединствен ID: ненегативан број I .

Једина комуникација међу чланицама је индиректна, путем бројева нажврљаних кредом на различитим локацијама широм града. Сваких 100 година, друштво врши попис како би пребројало своје чланице. Након што се попис заврши, свака чланица треба да зна укупан број чланица у друштву.

Попис се одвија током више дана. Сваког дана, свака чланица која још увек учествује у процесу ће изабрати и извршити тачно једну акцију: да **чита**, да **пише**, или да **престане** са учешћем у процесу.

- Ако чланица изабере **читање**, она бира локацију P . Током дана, она посећује локацију P и чита број који је тамо написан.
- Ако чланица изабере **писање**, она бира локацију P и број V . Касно увече, она посећује локацију P и мења број који је био тамо написан у V . Пошто је већ мрак, она не може да прочита стари број пре него што напише нови.
- Ако чланица изабере **престанак**, она више не предузима никакве акције у наредним данима.

Ако једна чланица види како друга пише број, она би могла да сазна њен идентитет. Стога је строго забрањено да две или више чланица бирају да пишу на истој локацији истог дана. (Не постоји такво ограничење за читање, пошто се оно може обавити дискретно.)

Ако једна или више чланица чита са локације на којој друга чланица жели да пише истог дана, сва читања се обављају пре писања.

Како би друштво требало да испланира процес пописа како би се минимализовао број дана док сви не сазнају тачан број чланица?

Имплементација

⇒ Ово је интерактивни проблем, у којем ће се истовремено извршавати непознат број инстанци ($1 \leq N \leq 100$) вашег програма. Свака инстанца симулира једну чланицу друштва.

Постоји 10^{18} локација. Број P локације мора задовољавати $0 \leq P < 10^{18}$. Почетно, вредност уписана на свим локацијама је $V = 0$.

Нова вредност V уписана на локацији мора увек бити цео број такав да је $0 \leq V \leq 10^9$. У већини подзатака, V може бити само 0 или 1. Погледајте секцију Бодовање за више детаља.

Када се инстанца вашег програма покрене, она треба прво да прочита линију са два цела броја, I и M ($0 \leq I \leq M - 1$): јединствени ID чланице друштва коју представља ова инстанца и укупан

број могућих ID-ева. У оквиру сваког тестног случаја, све инстанце ће добити исту вредност M и различите вредности I . Приметите да могу постојати ID-еви који нису додељени ниједној чланици.

Затим, за сваки дан у процесу пописа, ваш програм треба да изабере акцију коју жели да изврши и одштампа линију у складу са тим:

Акција	Значење
$r \ P$	Читање локације P . Након штампања ове линије, ваш програм треба да прочита линију са тренутном вредношћу која је уписана на P .
$w \ P \ V$	Писање на локацији P нове вредности V . Ако више инстанци пише на истој локацији P истог дана, добићете пресуду <i>Нећачно</i> . Осим за примере и подзатак 3, морате писати $0 \leq V \leq 1$; погледајте секцију Бодовање.
$! \ N$	Одговор и прекид : пријавите да има N чланица и престаните са учешћем у попису. Након што дате одговор, ваш програм треба нормално да се заврши . (Приметите да друге инстанце вашег програма могу наставити да раде још неколико дана пре него што одговоре и заврше се.)

Ако било која инстанца вашег програма одговори погрешну вредност за N , прекрши протокол, искористи више од 500 дана, или прекорачи (по процесу) временско/меморијско ограничење, ваше решење ће бити оцењено као *Нећачно* за дати тест случај.

У супротном, ваш програм ће бити (*Делимично*) *Тачан* на тест случају и бодован на основу вредности D : максималног броја дана који је било која инстанца утрошила да одговори. За пун број поена, морате решити сваки тест случај са $D \leq 61$ и $V \leq 1$. Погледајте секцију Бодовање за детаље.

Флашовање (излаза). Ако не користите обезбеђене темплејтове, обавезно флашујте стандардни излаз (flush) након штампања сваке линије, у супротном ваш програм може бити оцењен као *Нећачно*. У Python-у, ово се дешава аутоматски ако користите `input()` за читање линија. У C++, `cout << endl`; чисти `bafer` поред штампања новог реда; ако користите `printf`, користите `fflush(stdout)`.

Ограничења

- $1 \leq N \leq 100$.
- $1 \leq M \leq 100\,000$.
- Можете искористити највише 500 дана.

Бодовање

Ваш програм ће бити тестиран на неколико тест примера груписаних у подзатке. Да бисте добили поене за подзатак, морате исправно решити све тест примере које он садржи.

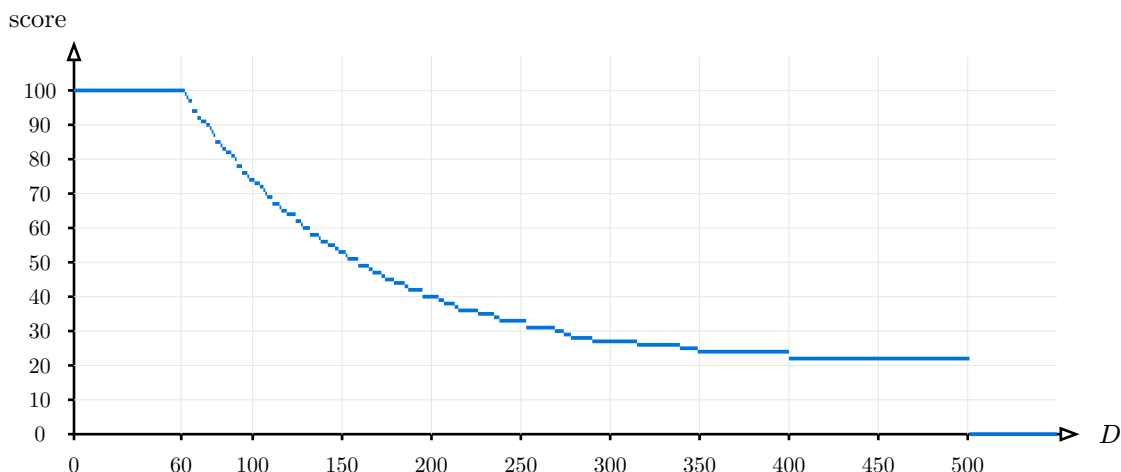
- **Подзатак 0 [0 поена]**: Примери (можете писати било који цео број $0 \leq V \leq 1\,000\,000\,000$).
- **Подзатак 1 [11 поена]**: $M \leq 100$, и N чланица имају ID-ове $0, 1, \dots, N - 1$.
- **Подзатак 2 [12 поена]**: $1 \leq N \leq 2$.
- **Подзатак 3 [22 поена]**: $M \leq 8000$, и можете писати било који цео број $0 \leq V \leq 1\,000\,000\,000$.
- **Подзатак 4 [55 поена]**: Без додатних ограничења.

У подзадацима 1, 2 и 4, можете писати само $V = 0$ или $V = 1$ у свакој акцији писања.

Нека је X_s максималан број поена за подзатак s (приказан изнад), и D_s највећи број дана који било који ваш програм користи на тесту у подзатку s . Тада:

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

Вредност score_s се заокружује на најближи цео број по подзатку, а ваш укупан резултат је збир ових вредности. Како би добили све поене на овом задатку, потребно је $D \leq 61$ и $V \leq 1$ на сваком тест случају.



Слика 1: Укупан резултат, под претпоставком да је сваки подзатак решен са истим максималним D .

Примери улаза/излаза

First example: 5 members, 100 possible IDs

Греј.	Реш. 0	Греј.	Реш. 1	Греј.	Реш. 2	Греј.	Реш. 3	Греј.	Реш. 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				

Second example: 2 members, 8000 possible IDs

Грејдер	Решење 0
0 8000	
	w 0 0
	w 1 1
	r 2
1	
	! 2

Грејдер	Решење 1
3 8000	
	w 2 1
	r 1
0	
	r 2
1	
	r 1
1	
	! 2

Објашњење

Први пример. Имамо $N = 5$ чланица са узастопним ID-овима 0, 1, 2, 3, 4 и $M = 100$ (важи за подзатке 1, 3 и 4). Решење i одговара чланици са ID-ом i . Интеракција изнад је само један могући валидан редослед операција и **није** замишљена као ефикасна или разумна стратегија; приказана је само да илуструје како протокол функционише.

Други пример. Имамо $N = 2$ чланице, са ID-овима 0 и 3, и $M = 8000$ (важи за подзатке 2, 3 и 4). Првог дана, чланица са ID-ом 0 пише 0 на локацији 0 (нема промене), а чланица са ID-ом 3 пише 1 на локацији 2.

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

Другог дана, ID 0 пише 1 на локацији 1, а ID 3 чита ту исту локацију. Приметите да се читање дешава током дана, пре писања увече. Стога ID 3 и даље види 0.

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

Трећег дана, обе читају локацију 2, где је уписана 1.

Четвртог дана, ID 0 одговара да има 2 чланице (тачно), док ID 3 чита 1 на локацији 1. ID 0 одмах излази након овога и не учествује у наредним данима.

Коначно, на дан $D = 5$, преостала чланица такође исправно одговара $N = 2$.

Алат за тестирање

Да бисмо олакшали тестирање вашег решења, обезбеђујемо једноставан алат који можете преузети са CMS-а. Коришћење алата је опционално. Приметите да се званични оцењивач на CMS-у разликује од овог алата за тестирање.

Да бисте користили алат потребан вам је улазни фајл. Можете користити обезбеђене примере улаза `sensus.input0.txt` и `sensus.input1.txt`, или направити своје. Улазни фајл треба да почне бројем чланица N и могућим ID-овима M , праћен линијом са N бројева који прецизирају ID-ове чланица друштва.

За Python програме, рецимо `sensus.py` (обично се покреће као `руру3 sensus.py`) покрените алат за тестирање на следећи начин:

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

За C++ програме, прво компајлирајте своје решење:

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

а затим покрените алат за тестирање:

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

Приметити да се у овом задатку стандардни излаз користи за комуникацију са грејдером, тако да га не треба користити за дебаговање. Уместо тога, можеш користити стандардни излаз за грешке (stderr). У C++-у можеш користити `cerr << msg << endl;`. У Python-у можеш користити `print(msg, file=sys.stderr)`.

Алат за тестирање ће прочитати и презентовати ове stderr поруке заједно са упитима изведених од свих инстанци твог програма. Приметити да због техничких разлога они могу да се прикажу не синхронизовани једни са другима.