D. Преброяване (census)

Ограничение по време: 1 секунди
Ограничение по памет: 128 MiB

Малко известен факт за Чезенатико е, че той е дом на тайно общество от N информатички. Това общество е много потайно; никой член не знае самоличността на останалите. Всеки член има уникално ID: неотрицателно цяло число I.

Единствената комуникация между членовете е косвена, чрез числа, надраскани с тебешир на различни места из града. Веднъж на 100 години обществото провежда преброяване на членовете си. След приключване на преброяването, всеки член трябва да знае общия брой на членовете в обществото.

Преброяването протича в рамките на няколко дни. Всеки ден всеки член, който все още участва в процеса, избира и извършва точно едно действие: да **чете**, да **пише** или да **спре** участието си.

- Ако един член избере да **чете**, той избира място P. През деня той посещава мястото P и прочита числото, което е написано там.
- Ако един член избере да **пише**, той избира място P и число V. Късно вечерта той посещава мястото P и променя числото, което е било написано там, на V. Тъй като вече е тъмно, той не може да прочете старото число, преди да напише новото.
- Ако един член избере да **спре**, той няма да предприема действия през следващите дни.

Ако един член види друг да пише число, той ще да научи самоличността му. Затова е строго забранено двама или повече членове да изберат да пишат на едно и също място в един и същи ден. (Няма такова ограничение за четене, тъй като това може да се прави дискретно.)

Ако един или повече членове четат от място, където друг член иска да пише в същия ден, всички четения се случват преди писането.

Как обществото трябва да планира своя процес на преброяване, за да минимизира броя на дните, докато всеки научи правилния брой на членовете?

Имплементация

> ⇨ Това е интерактивна задача, в която неизвестен брой копия ($1 \leq N \leq 100$) на Вашата програма ще бъдат изпълнявани едновременно. Всяко копие симулира един член на обществото.

Има 10^{18} места. Числото P на мястото трябва да отговаря на $0 \leq P < 10^{18}$. Първоначално стойността, записана на всички места, е $V = 0$.

Новата стойност V, записана на дадено място, трябва винаги да бъде цяло число, такова че $0 \leq V \leq 10^9$. В повечето подзадачи V може да бъде 0 или 1. Вижте секцията „Оценяване“ за повече подробности.

Когато стартира копие на Вашата програма, тя първо трябва да прочете ред с две цели числа, I и M ($0 \leq I \leq M - 1$): уникалното ID на члена на обществото, представен от това копие, и общия брой на възможните ID-та. В рамките на всеки тестов пример всички копия ще получат една

и съща стойност M и различни стойности за I. Забележете, че може да има ID-та, които не са присвоени на никой член.

След това, за всеки ден от процеса на преброяване, Вашата програма трябва да избере действието, което иска да извърши, и съответно да отпечата ред:

Действие	Значение
`r` P	**Четене** от място P. След отпечатването на този ред, Вашата програма трябва да прочете ред с текущата стойност, записана на P.
`w` P V	**Писане** на място P на новата стойност V. Ако няколко копия пишат на едно и също P в един и същи ден, ще получите съобщение *Not correct*. С изключение на примерите и подзадача 3, трябва да пишете $0 \leq V \leq 1$; вижте секцията „Оценяване“.
`!` N	**Отговор и спиране**: съобщете, че има N членове и спрете участието си в преброяването. След отговора, **Вашата програма трябва да приключи нормално**. (Забележете, че други копия на вашата програма може да продължат да работят още няколко дни, преди да отговорят и да приключат.)

Ако някое копие на Вашата програма отговори с грешна стойност за N, наруши протокола, използва повече от 500 дни или надвиши ограничението за време/памет (на процес), Вашето решение ще бъде оценено като *Not correct* за съответния тестов пример.

В противен случай, Вашата програма ще бъде *(Partially) correct* за тестовия пример и оценена въз основа на стойността D: максималният брой дни, които някое от копията е използвало, за да отговори. За пълен брой точки трябва да решите всеки тестов пример с $D \leq 61$ и $V \leq 1$. Вижте секцията „Оценяване“ за подробности.

Изчистване на буфера. Ако не използвате предоставените шаблони, уверете се, че изчиствате стандартния изход (flush) след отпечатването на всеки ред, в противен случай програмата ви може да бъде оценена като *Not correct*. В Python това се случва автоматично, ако използвате `input()` за четене на редове. В C++, `cout << endl;` изчиства буфера в допълнение към отпечатването на нов ред; ако използвате printf, използвайте `fflush(stdout)`.

Ограничения

- $1 \leq N \leq 100$.
- $1 \leq M \leq 100\,000$.
- Можете да използвате най-много 500 дни.

Оценяване

Вашата програма ще бъде тествана върху няколко тестови примера, групирани в подзадачи. За да получите точките за дадена подзадача, трябва да решите правилно всички тестове, които тя съдържа.

- **Подзадача 0 [0 точки]**: Примери (можете да записвате всяко цяло число $0 \leq V \leq 1\,000\,000\,000$).
- **Подзадача 1 [11 точки]**: $M \leq 100$ и N членове имат ID-та $0, 1, ..., N-1$.
- **Подзадача 2 [12 точки]**: $1 \leq N \leq 2$.

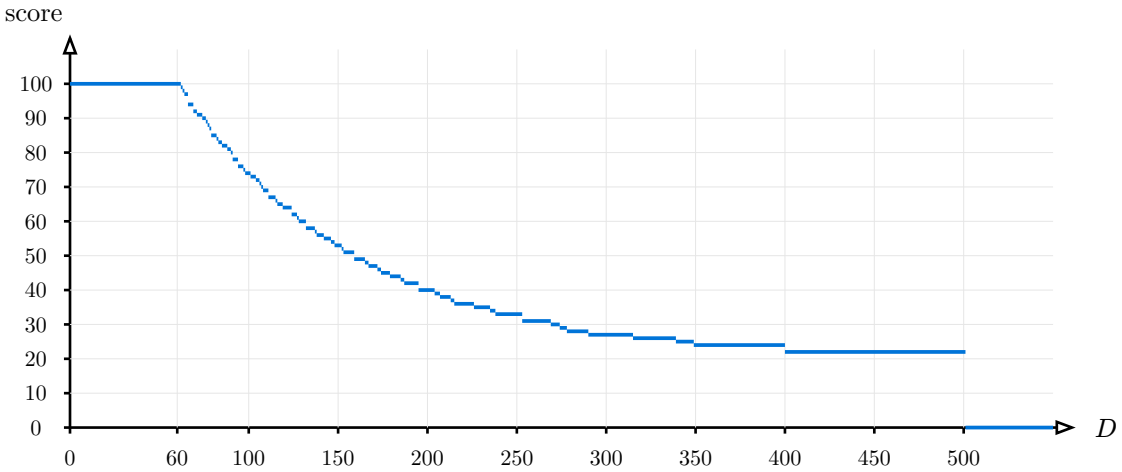
- **Подзадача 3 [22 точки]**: $M \leq 8000$ и можете да записвате всяко цяло число $0 \leq V \leq 1\,000\,000\,000$.
- **Подзадача 4 [55 точки]**: Без допълнителни ограничения.

В подзадачи 1, 2 и 4 можете да записвате само $V = 0$ или $V = 1$ при всяко действие „Писане“.

Нека X_s са максималните точки за подзадача s (показани по-горе), а D_s е най-големият брой дни, които някоя от Вашите програми използва при тест от подзадача s. Тогава:

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

Стойността на score_s се закръгля до най-близкото цяло число за всяка подзадача, а общият ви резултат е сумата от тях. За пълен брой точки трябва $D \leq 61$ и $V \leq 1$ при всеки тестов пример.



Фиг. 1: Общ резултат, приемайки че всяка подзадача е решена с един и същ максимален D.

Примерни входове/изходи

First example: 5 members, 100 possible IDs

Оцен.	Реш. 0	Оцен.	Реш. 1	Оцен.	Реш. 2	Оцен.	Реш. 3	Оцен.	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
	r 50		r 7		r 12		w 1 1	0	
1		1		1					! 5
	! 5		r 1		w 0 0		! 5		
		1							
			! 5		! 5				

Second example: 2 members, 8000 possible IDs

Оценител	Решение 0
0 8000	
	w 0 0
	w 1 1
	r 2
1	
	! 2

Оценител	Решение 1
3 8000	
	w 2 1
	r 1
0	
	r 2
1	
	r 1
1	
	! 2

Обяснение

Първи пример. Имаме $N = 5$ членове с последователни ID-та $0, 1, 2, 3, 4$ и $M = 100$ (валидно за подзадачи 1, 3 и 4). Решението i съответства на члена с ID i. Взаимодействието по-горе е само една възможна валидна поредица от операции и **не** е предвидено да бъде ефективна или разумна стратегия; показано е само за да илюстрира как работи протоколът.

Втори пример. Имаме $N = 2$ членове, с ID-та 0 и 3, и $M = 8000$ (валидно за подзадачи 2, 3 и 4). Първия ден членът с ID 0 записва 0 на място 0 (няма промяна), а членът с ID 3 записва 1 на място 2.

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

На втория ден ID 0 записва 1 на място 1, а ID 3 чете от същото място. Забележете, че четенето се случва през деня, преди писането вечерта. Следователно ID 3 все още вижда 0.

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

На третия ден и двамата четат от място 2, където е записана 1.

На четвъртия ден ID 0 отговаря, че има 2 членове (правилно), докато ID 3 чете 1 от място 1. ID 0 веднага след това приключва и не участва в следващите дни.

Накрая, на ден $D = 5$, останалият член също правилно отговаря, че $N = 2$.

Инструмент за тестване

За улеснение при тестването на Вашето решение, предоставяме прост инструмент, който можете да изтеглите от CMS. Използването на инструмента е по желание. Забележете, че официалният оценител в CMS се различава от инструмента за тестване.

За да използвате инструмента, ви е необходим входен файл. Можете да използвате предоставените примерни входове `census.input0.txt` и `census.input1.txt` или да създадете свои собствени. Входният файл трябва да започва с броя на членовете N и възможните ID-та M, последван от ред с N числа, указващи ID-тата на членовете на обществото.

За Python програми, примерно `census.py` (обикновено се изпълнява като `pypy3 census.py`), стартирайте инструмента за тестване по следния начин:

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

За C++ програми, първо компилирайте Вашето решение:

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

и след това стартирайте инструмента за тестване:

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

Забележете, че в тази задача стандартния изход се използва за комуникация с инструмента за тестване, така че не трябва да бъде използван за контролни печати. Вместо това може да ползвате стандартния поток за грешки ('stderr'). В C++ може да ползвате `cerr << msg << endl;`. В Python може да ползвате `print(msg, file=sys.stderr)`.

Инструментът за тестване ще чете тези контролни печати заедно със заявките, изпълнени от всичките инстанции на Вашата програма. Имайте предвид, че поради технически причини те може да се печатат несинхронизирано помежду си.