

D. Spis powszechny (census)

Limit czasu: 1 sekundy

Limit pamięci: 128 MiB

Mało znanym faktem o Cesenatico jest to, że jest ono domem tajnego stowarzyszenia N informatyczek. To stowarzyszenie jest niezwykle tajne; żadna z członkiń nie zna tożsamości żadnej innej. Każda członkini ma unikalny indentyfikator; całkowitą liczbę nieujemną I .

Członkinie nie komunikują się bezpośrednio. Jedyna dozwolona komunikacja odbywa się za pomocą liczb zapisywanych kredą w różnych lokalizacjach w mieście. Co 100 lat stowarzyszenie przeprowadza spis powszechny, aby policzyć swoje członkinie. Po zakończeniu spisu każda członkini powinna znać całkowitą liczbę członkiń stowarzyszenia.

Spis odbywa się na przestrzeni wielu dni. Codziennie każda członkini, która nadal bierze udział w procesie, wybiera i wykonuje **dokładnie jedną** akcję: **read**, **write** albo **stop**.

- Jeśli członkini wykonuje **read**, wybiera lokalizację P . W ciągu dnia odwiedza lokalizację P i odczytuje liczbę, która jest tam zapisana.
- Jeśli członkini wybierze **write**, wybiera lokalizację P oraz liczbę V . Późnym wieczorem odwiedza lokalizację P i zmienia liczbę, która była tam zapisana, na V . Ponieważ jest już ciemno, nie może odczytać starej liczby przed zapisaniem nowej.
- Jeśli członkini wybierze **stop**, nie podejmuje już żadnych działań w kolejnych dniach.

Jeśli jedna członkini zobaczy, jak inna zapisuje liczbę, mogłaby ją rozpoznać. Dlatego surowo zabrania się, aby dwie lub więcej członkiń wykonało operację **write** w tej samej lokalizacji tego samego dnia. (Nie ma takiego ograniczenia dla operacji **read**, ponieważ można ją wykonać dyskretnie.)

Jeśli w tej samej lokalizacji tego samego dnia akcje wykona więcej niż jedna członkini, to wszystkie operacje **read** odbywają się przed operacją **write**.

W jaki sposób stowarzyszenie powinno zaplanować swój proces spisu, aby zminimalizować liczbę dni do momentu, w którym każda członkini pozna prawidłową liczbę członkiń stowarzyszenia?

Implementacja

⇒ Jest to zadanie interaktywne, w którym jednocześnie uruchamiana jest nieznana liczba instancji ($1 \leq N \leq 100$) twojego programu. Każda instancja symuluje jedną członkinie stowarzyszenia.

Istnieje dokładnie 10^{18} lokalizacji. Każda z nich jest opisana liczbą P spełniającą $0 \leq P < 10^{18}$. Początkowo w każdej lokalizacji zapisane jest $V = 0$.

Nowa wartość V zapisana w lokalizacji musi być liczbą całkowitą taką, że $0 \leq V \leq 10^9$. W większości podzadań V może wynosić tylko 0 lub 1. Zobacz sekcję Punktacja, aby uzyskać więcej szczegółów.

Kiedy instancja twojego programu się uruchamia, powinna najpierw wczytać jedną linię z dwiema liczbami całkowitymi, I oraz M ($0 \leq I \leq M - 1$): unikalny identyfikator członkini stowarzyszenia reprezentowanej przez tę instancję oraz maksymalna możliwa liczba różnych identyfikatorów. W

każdym przypadku testowym wszystkie instancje otrzymają tę samą wartość M oraz różne wartości I . Zauważ, że mogą istnieć identyfikatory, które nie są przypisane do żadnej członkini.

Następnie, Twój program dla każdego dnia, w którym uczestniczy w spisie, powinien wybrać akcję, którą chce wykonać, i odpowiednio wypisać linię:

Akcja	Znaczenie
$r\ P$	Read w lokalizacji P . Po wypisaniu tej linii, Twój program powinien wczytać linię z bieżącą wartością zapisaną w P .
$w\ P\ V$	Write : zapisuje wartość V w lokalizacji P . Jeśli wiele instancji zapisze w tej samej lokalizacji P tego samego dnia, otrzymasz werdykt <i>Błędna odpowiedź</i> (Not correct). Z wyjątkiem przykładów i podzadania 3, musisz zapisać $0 \leq V \leq 1$; zobacz sekcję Punktacja.
$!\ N$	Odpowiedz i stop : zgłoś, że istnieje N członkiń i zakończ udział w spisie. Po udzieleniu odpowiedzi, twój program powinien zakończyć działanie w normalny sposób . (Zauważ, że inne instancje twojego programu mogą kontynuować działanie przez kolejne dni, zanim udzielą odpowiedzi i zakończą pracę.)

Jeśli jakakolwiek instancja twojego programu udzieli błędnej odpowiedzi N , naruszy protokół, użyje więcej niż 500 dni lub przekroczy limit czasu/pamięci (oddzielny na każdy proces), twoje zgłoszenie zostanie ocenione jako *Not correct* (Błędna odpowiedź) dla danego przypadku testowego.

W przeciwnym razie twój program otrzyma ocenę *(Partially) Correct* ((Częściowo) Poprawny) dla danego przypadku testowego i zostanie oceniony na podstawie wartości D : maksymalnej liczby dni, jakiej potrzebowała jakakolwiek instancja, aby udzielić odpowiedzi. Aby uzyskać pełną liczbę punktów, musisz rozwiązać każdy przypadek testowy z $D \leq 61$ i $V \leq 1$. Zobacz sekcję Punktacja, aby uzyskać szczegóły.

Czyszczenie bufora (Flushing). Jeśli nie używasz dostarczonych szablonów, upewnij się, że czyścisz standardowe wyjście po wypisaniu każdej linii, w przeciwnym razie twój program może zostać oceniony jako *Błędna odpowiedź*. W języku Python dzieje się to automatycznie, jeśli używasz `input()` do odczytu linii. W C++, `cout << endl;` poza wypisaniem nowej linii czyści bufor; jeśli używasz `printf`, użyj `fflush(stdout)`.

Ograniczenia

- $1 \leq N \leq 100$.
- $1 \leq M \leq 100\,000$.
- Możesz użyć maksymalnie 500 dni.

Punktacja

Zestaw testów dzieli się na następujące podzadania. Testy do każdego podzadania składają się z jednej lub większej liczby testów. Żeby zdobyć punkty za podzadanie, Twój program musi poprawnie przejść wszystkie testy, które do niego należą.

- **Podzadanie 0 [0 punktów]**: Przykłady (w operacji write możesz zapisać dowolną liczbę całkowitą $0 \leq V \leq 1\,000\,000\,000$).
- **Podzadanie 1 [11 punktów]**: $M \leq 100$, a N członkiń ma identyfikatory $0, 1, \dots, N - 1$.
- **Podzadanie 2 [12 punktów]**: $1 \leq N \leq 2$.

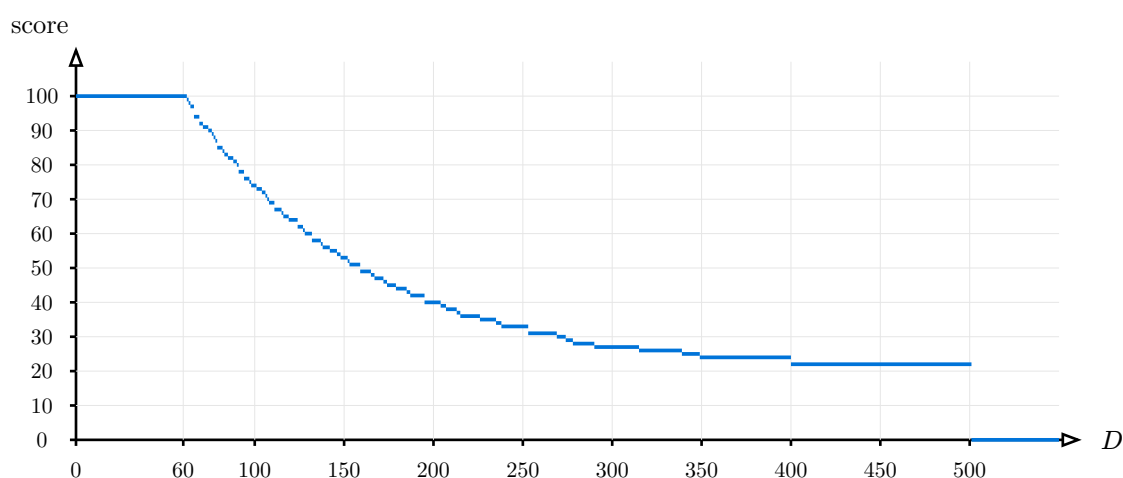
- **Podzadanie 3 [22 punkty]:** $M \leq 8000$, a ty w operacji write możesz zapisać dowolną liczbę całkowitą $0 \leq V \leq 1000000000$.
- **Podzadanie 4 [55 punktów]:** Brak dodatkowych ograniczeń.

W podzadaniach 1, 2 i 4 w każdej akcji write możesz zapisać tylko $V = 0$ lub $V = 1$.

Niech X_s oznacza maksymalną liczbę punktów za podzadanie s (pokazane powyżej), a D_s największą liczbę dni, jakiej użył którykolwiek z twoich programów w teście w podzadaniu s . Wtedy:

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

Wartość wynik_s jest zaokrąglana do najbliższej liczby całkowitej dla każdego podzadania, a twój łączny wynik jest sumą tych wartości. Aby uzyskać pełną liczbę punktów, musisz uzyskać $D \leq 61$ oraz $V \leq 1$ w każdym przypadku testowym.



Rysunek 1: Łączny wynik, przy założeniu, że każde podzadanie jest rozwiązane z tym samym maksymalnym D .

Przykłady

Pierwszy przykład. Każda para kolumn pokazuje komunikację pomiędzy sprawdzaczką, a instancją.

Spr.	Inst. 0	Spr.	Inst. 1	Spr.	Inst. 2	Spr.	Inst. 3	Spr.	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 przykład.

Sprawdzaczka	Instancja 0	Sprawdzaczka	Instancja 1
0 8000		3 8000	
	w 0 0		w 2 1
	w 1 1		r 1
		0	
	r 2		r 2
1		1	
	! 2		r 1
		1	
			! 2

Wyjaśnienie

Pierwszy przykład. Mamy $N = 5$ członkiń z kolejnymi identyfikatorami 0, 1, 2, 3, 4 oraz $M = 100$ (poprawne dla podzadań 1, 3 i 4). Instancja i odpowiada członkini o identyfikatorze i . Interakcja powyżej jest tylko jedną z możliwych legalnych sekwencji operacji i **nie** ma na celu bycia wydajną lub sensowną strategią; jest pokazana tylko w celu zilustrowania, jak działa protokół.

Drugi przykład. Mamy $N = 2$ członkinie, o identyfikatorach 0 i 3, oraz $M = 8000$ (poprawne dla podzadań 2, 3 i 4). Pierwszego dnia członkini o identyfikatorze 0 zapisuje 0 w lokalizacji 0 (brak zmiany), a członkini o identyfikatorze 3 zapisuje 1 w lokalizacji 2.

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

Drugiego dnia identyfikator 0 zapisuje 1 w lokalizacji 1, a identyfikator 3 odczytuje tę samą lokalizację. Zauważ, że odczyt następuje w ciągu dnia, przed wieczornym zapisem. Dlatego identyfikator 3 nadal widzi 0.

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

Trzeciego dnia obie odczytują lokalizację 2, w której zapisana jest 1.

Czwartego dnia identyfikator 0 odpowiada, że są 2 członkinie (poprawnie), podczas gdy identyfikator 3 odczytuje 1 w lokalizacji 1. Identyfikator 0 natychmiast po tym kończy pracę i nie bierze udziału w kolejnych dniach.

Ostatecznie, w dniu $D = 5$, pozostała członkini również poprawnie odpowiada $N = 2$.

Narzędzie do testowania

Aby ułatwić testowanie twojego rozwiązania, udostępniamy proste narzędzie, które możesz pobrać z CMS. Użycie narzędzia jest opcjonalne. Zauważ, że oficjalna sprawdzaczka na CMS różni się od narzędzia testowego.

Aby użyć narzędzia, potrzebujesz pliku wejściowego. Możesz użyć dostarczonych przykładowych danych wejściowych `census.input0.txt` i `census.input1.txt`, lub stworzyć własne. Plik wejściowy powinien zaczynać się od liczby członkiń N i możliwych identyfikatorów M , a następnie zawierać linię z N liczbami określającymi identyfikatory członkiń stowarzyszenia.

Dla programów w Pythonie, załóżmy `census.py` (zazwyczaj uruchamiany jako `pypy3 census.py`), uruchom narzędzie testowe w następujący sposób:

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

Dla programów w C++, najpierw skompiluj swoje rozwiązanie:

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

a następnie uruchom narzędzie testowe:

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

Zauważ, że w tym zadaniu standardowego wyjścia używa się do komunikacji ze sprawdzaczką, więc nie powinno być używane do debugowania. W zamian możesz używać standardowego wyjścia błędów (`stderr`). W C++ możesz użyć `cerr << msg << endl; .` W Pythonie możesz użyć `print(msg, file=sys.stderr)`.

Narzędzie do testowania przeczyta i wyświetli wiadomości wypisane na `stderr` razem z zapytaniami wykonanymi przez wszystkie instancje Twojego programu. Zauważ, że z powodów technicznych te wiadomości mogą pojawić się w różniącej się kolejności.