

## D. Перепис (census)

Обмеження часу: 1 секунди

Обмеження пам'яті: 128 MiB

Маловідомий факт про Чезенатіко полягає в тому, що це дім для таємного товариства з  $N$  інформатикинь. Це товариство справді дуже таємне; жодна його членкиня не знає іншу. Кожна членкиня має унікальний ідентифікатор: ціле невід'ємне число  $I$ .

Єдиний зв'язок між членкинями здійснюється опосередковано через числа, записані крейдою в різних місцях по всьому місту. Кожні 100 років товариство проводить перепис, щоб підрахувати кількість своїх членкинь. Після завершення перепису кожна членкиня повинна знати загальну кількість членкинь у товаристві.

Перепис триває кілька днів. Щодня кожна членкиня, яка ще бере участь у процесі, обирає та виконує рівно одну дію: **прочитати**, **записати** або **припинити** участь.

- Якщо членкиня обирає **прочитати**, вона вибирає місце  $P$ . Вдень вона відвідує місце  $P$  і читає число, записане там.
- Якщо членкиня обирає **записати**, вона вибирає місце  $P$  та число  $V$ . Пізно ввечері вона відвідує місце  $P$  і змінює число, записане там, на  $V$ . Оскільки вже темно, вона не може прочитати старе число перед тим, як записати нове.
- Якщо членкиня обирає **припинити**, вона більше не виконує ніяких дій у наступні дні.

Якщо одна членкиня бачить, як інша записує число, вона може впізнати її. Тому суворо заборонено двом або більше членкиням записувати число у тому самому місці в один і той самий день. (Для читання такого обмеження немає, оскільки це можна зробити непомітно.)

Якщо одна або більше членкинь читають з місця, де інша членкиня хоче зробити запис того ж дня, всі операції читання відбуваються перед операцією запису.

Як товариству спланувати процес перепису, щоб мінімізувати кількість днів до того, як усі дізнаються правильну кількість членкинь?

### Реалізація

⇒ Це інтерактивна задача, у якій невідома кількість екземплярів ( $1 \leq N \leq 100$ ) вашої програми буде запущена одночасно. Кожен екземпляр моделює одну членкиню товариства.

Існує  $10^{18}$  місць. Номер  $P$  місця повинен задовольняти  $0 \leq P < 10^{18}$ . Початково значення, записане в усіх місцях, дорівнює  $V = 0$ .

Нове значення  $V$ , записане в місці, завжди має бути цілим числом, таким що  $0 \leq V \leq 10^9$ . У більшості підзадач  $V$  може бути лише 0 або 1. Див. розділ «Оцінювання» для отримання додаткових відомостей.

Коли екземпляр вашої програми запускається, він повинен спочатку зчитати рядок із двома цілими числами  $I$  та  $M$  ( $0 \leq I \leq M - 1$ ): унікальний ідентифікатор членкині товариства, представленого цим екземпляром, та загальну кількість можливих ідентифікаторів. У межах кожного

тесту всі екземпляри отримають однакове значення  $M$  та різні значення  $I$ . Зауважте, що можуть існувати ідентифікатори, які не призначені жодній членкині.

Потім, для кожного дня процесу перепису, ваша програма повинна обрати дію, яку вона хоче виконати, і вивести відповідний рядок:

Дія	Значення
$r\ P$	<b>Читати</b> (Read) з місця $P$ . Після виведення цього рядка ваша програма повинна зчитати рядок із поточним значенням, записаним у $P$ .
$w\ P\ V$	<b>Записати</b> (Write) у місці $P$ нове значення $V$ . Якщо кілька екземплярів записують у те саме місце $P$ в один і той же день, ви отримаєте вердикт <i>Неправильна відповідь</i> (Not correct). За винятком прикладів та підзадачі 3, ви повинні записувати $0 \leq V \leq 1$ ; див. розділ «Оцінювання».
$!\ N$	<b>Відповідь і зупинка</b> : повідомити, що є $N$ членкинь, і припинити участь у переписі. Після надання відповіді <b>ваша програма повинна нормально завершити роботу</b> . (Зауважте, що інші екземпляри вашої програми можуть продовжувати роботу протягом додаткових днів, перш ніж вони дадуть відповідь і завершать роботу.)

Якщо будь-який екземпляр вашої програми дає неправильне значення  $N$ , порушує протокол, використовує більше 500 днів або перевищує ліміт часу/пам'яті (на процес), ваша відповідь отримає вердикт *Неправильна відповідь* (Not correct) для даного тесту.

В іншому випадку ваша програма отримає вердикт *Частково зараховано* (Partially Correct) на тесті й отримає оцінку на основі значення  $D$ : максимальної кількості днів, яку витратив будь-який екземпляр для відповіді. Для отримання повної оцінки вам потрібно розв'язати кожен тест із  $D \leq 61$  та  $V \leq 1$ . Див. розділ «Оцінювання» для отримання додаткових відомостей.

**Буферизація (Flushing).** Якщо ви не використовуєте надані шаблони, переконайтеся, що ви очищаєте стандартний вивід після друку кожного рядка, інакше ваша програма може отримати вердикт *Неправильна відповідь* (Not correct). У Python це відбувається автоматично, якщо ви використовуєте `input()` для читання рядків. У C++ `cout << endl`; очищає буфер на додаток до друку символу нового рядка; якщо використовуєте `printf`, використовуйте `fflush(stdout)`.

## Обмеження

- $1 \leq N \leq 100$ .
- $1 \leq M \leq 100\,000$ .
- Ви можете використовувати щонайбільше 500 днів.

## Оцінювання

Ваша програма буде протестована на кількох тестах, згрупованих у підзадачі. Щоб отримати оцінку за підзадачу, ви повинні правильно розв'язати всі тести, що містяться в ній.

- **Підзадача 0 [0 балів]:** Приклади (ви можете записувати будь-яке ціле число  $0 \leq V \leq 1\,000\,000\,000$ ).
- **Підзадача 1 [11 балів]:**  $M \leq 100$ , і  $N$  членкинь мають ідентифікатори  $0, 1, \dots, N - 1$ .
- **Підзадача 2 [12 балів]:**  $1 \leq N \leq 2$ .
- **Підзадача 3 [22 балів]:**  $M \leq 8000$ , і ви можете записувати будь-яке ціле число  $0 \leq V \leq 1\,000\,000\,000$ .
- **Підзадача 4 [55 балів]:** Без додаткових обмежень.

У підзадачах 1, 2 та 4 ви можете записувати лише  $V = 0$  або  $V = 1$  у кожній дії Записати (Write).

Нехай  $X_s$  — максимальна кількість балів за підзадачу  $s$  (наведена вище), а  $D_s$  — найбільша кількість днів, використана будь-якою з ваших програм на тесті в підзадачі  $s$ . Тоді:

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

Значення  $\text{score}_s$  заокруглюється до найближчого цілого числа для кожної підзадачі, а ваша загальна оцінка є сумою цих значень. Для отримання повної оцінки за задачу вам потрібно  $D \leq 61$  та  $V \leq 1$  для кожного тесту.

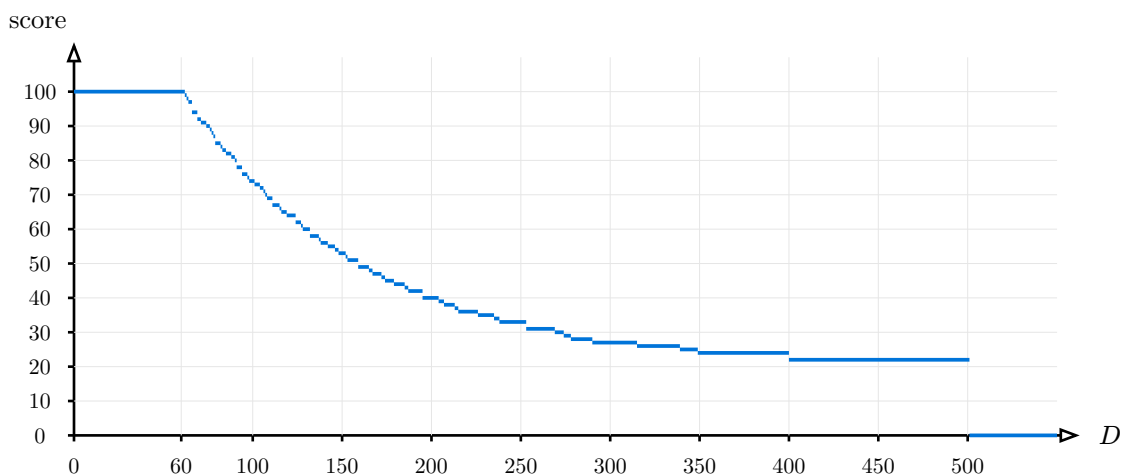


Рисунок 1: Загальна оцінка, припускаючи, що кожна підзадача розв'язана з однаковим максимальним значенням  $D$ .

## Приклади вводу/виводу

Перший приклад. Кожна пара стовпців показує взаємодію між перевіряючою системою (grader) та одним екземпляром (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				

Другий приклад.

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

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

## Пояснення

**Перший приклад.** У нас є  $N = 5$  членкинь із послідовними ідентифікаторами 0, 1, 2, 3, 4 та  $M = 100$  (виконується для підзадач 1, 3 та 4). Екземпляр  $i$  відповідає членкині з ідентифікатором  $i$ . Наведена вище взаємодія є лише однією з можливих допустимих послідовностей операцій і **не** є ефективною чи розумною стратегією; вона наведена лише для ілюстрації того, як працює протокол.

**Другий приклад.** У нас є  $N = 2$  членкині з ідентифікаторами 0 та 3, і  $M = 8000$  (виконується для підзадач 2, 3 та 4). У перший день членкиня з ідентифікатором 0 записує 0 у місці 0 (без змін), а членкиня з ідентифікатором 3 записує 1 у місці 2.

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

На другий день членкиня з ідентифікатором 0 записує 1 у місці 1, а членкиня з ідентифікатором 3 читає те саме місце. Зауважте, що читання відбувається вдень, перед записом увечері. Отже, членкиня з ідентифікатором 3 все ще бачить 0.

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

На третій день вони обидві читають місце 2, де записана 1.

На четвертий день членкиня з ідентифікатором 0 відповідає, що є 2 членкині (правильно), тоді як членкиня з ідентифікатором 3 читає 1 у місці 1. Членкиня з ідентифікатором 0 негайно завершує роботу після цього і не бере участі в наступні дні.

Нарешті, на день  $D = 5$ , членкиня, що залишилась, також правильно відповідає  $N = 2$ .

## Тестування

Для полегшення тестування вашого розв'язку ми надаємо простий інструмент, який ви можете завантажити з CMS. Використання інструменту є необов'язковим. Зауважте, що офіційна перевіряюча система (grader) на CMS відрізняється від інструменту тестування.

Щоб використовувати цей інструмент, вам потрібен вхідний файл. Ви можете використовувати надані приклади вхідних даних `census.input0.txt` та `census.input1.txt` або створити власні. Вхідний файл повинен починатися з кількості членкинь  $N$  та можливих ідентифікаторів  $M$ , за якими слідує рядок із  $N$  числами, що вказують ідентифікатори членкинь товариства.

Для програм на Python, скажімо `census.py` (зазвичай запускається як `pyru3 census.py`), запустіть інструмент тестування наступним чином:

```
python3 testing_tool.py pyru3 census.py < census.input0.txt
```

Для програм на C++, спочатку скомпілюйте ваш розв'язок:

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

а потім запустіть інструмент тестування:

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

Зауважте, що в цій задачі стандартний вивід використовується для спілкування з перевіряючою системою, тому його не слід використовувати для налагодження. Натомість ви можете використовувати стандартний потік помилок (`stderr`). У C++ ви можете використовувати `cerr << msg << endl`; У Python ви можете використовувати `print(msg, file=sys.stderr)`.

Інструмент тестування зчитуватиме та відображатиме ці повідомлення зі стандартного потоку помилок (`stderr`) разом із запитами, виконаними всіма екземплярами вашої програми. Зауважте, що з технічних причин вони можуть відображатися з невеликою розсинхронізацією.