

D. Sensus (census)

Time limit: 1 detik

Memory limit: 128 MiB

Fakta yang jarang diketahui mengenai Cesenatico: kota ini merupakan rumah perkumpulan rahasia yang beranggotakan N informatikawan perempuan. Perkumpulan ini sangat rahasia sampai-sampai tidak ada anggota yang mengetahui identitas anggota lainnya. Setiap anggota memiliki ID unik: sebuah bilangan bulat non-negatif I .

Komunikasi dilakukan melalui penulisan angka-angka di berbagai lokasi. Setiap 100 tahun, perkumpulan ini melakukan sebuah sensus untuk menghitung jumlah anggotanya. Setelah selesai, setiap anggota harus mengetahui total anggota di dalam perkumpulan tersebut.

Sensus berlangsung selama beberapa hari. Setiap harinya, semua anggota yang masih lanjut akan melakukan tepat satu tindakan dari membaca (**read**), menulis (**write**), atau berhenti (**stop**).

- Apabila seorang anggota memilih untuk membaca (**read**), maka ia perlu memilih lokasi P . Pada siang hari, ia akan mengunjungi lokasi P dan membaca angka yang tertulis di sana.
- Apabila seorang anggota memilih untuk menulis (**write**), maka ia perlu memilih lokasi P dan angka V . Pada malam hari, ia akan mengunjungi lokasi P dan mengubah angka yang tertulis di sana menjadi V . Karena sudah gelap, ia tidak bisa membaca angka sebelumnya.
- Apabila seorang anggota memilih untuk berhenti (**stop**), ia tidak akan melakukan tindakan apa pun lagi di hari-hari berikutnya

Seorang anggota dapat mengetahui identitas anggota lain jika ia melihat anggota lain tersebut menulis sebuah angka. Oleh karena demikian, mereka tidak diperbolehkan untuk menulis di lokasi yang sama pada hari yang sama. (Hal ini tidak berlaku untuk membaca karena bisa dilakukan secara diam-diam.)

Dalam suatu hari, semua pembacaan terjadi sebelum penulisan.

Bagaimana cara perkumpulan ini merencanakan proses sensusnya demi meminimalkan jumlah hari sampai semua orang mengetahui jumlah anggota yang benar?

Implementasi

⇒ Permasalahan ini bersifat interaktif, sejumlah instans ($1 \leq N \leq 100$) dari program Anda akan dijalankan secara bersamaan. Setiap instans mensimulasikan satu anggota dari perkumpulan tersebut.

Terdapat 10^{18} lokasi. Nomor P dari suatu lokasi harus memenuhi $0 \leq P < 10^{18}$. Awalnya, nilai yang tertulis di semua lokasi adalah $V = 0$.

Nilai baru V yang ditulis di suatu lokasi harus berupa bilangan bulat sedemikian sehingga $0 \leq V \leq 10^9$. Di sebagian besar subsoal, V hanya bisa bernilai 0 atau 1. Lihat bagian Penilaian untuk detail lebih lanjut.

Saat instans program Anda dimulai, program Anda pertama-tama harus membaca dua bilangan bulat, I dan M ($0 \leq I \leq M - 1$): ID unik yang diwakili oleh instans ini dan total jumlah ID yang mungkin. Di setiap kasus uji, semua instans akan mendapatkan nilai M yang sama dan nilai I

yang berbeda-beda. Perhatikan bahwa mungkin saja ada ID yang tidak diberikan kepada anggota mana pun.

Kemudian, untuk setiap hari dalam proses sensus, program Anda harus memilih tindakan yang ingin dilakukan dan mencetak baris yang sesuai dengan:

Tindakan	Makna
$r\ P$	Membaca (read) lokasi P . Setelah mencetak baris ini, program Anda harus membaca sebuah baris dengan nilai yang menyatakan angka yang tertulis di lokasi P .
$w\ P\ V$	Menulis (write) di lokasi P dengan nilai V . Apabila beberapa instans menulis di lokasi P di hari yang sama, Anda akan mendapatkan <i>Not correct</i> . Kecuali untuk contoh dan subsoal 3, Anda harus menulis $0 \leq V \leq 1$ (lihat bagian Penilaian).
$!\ N$	Menjawab dan berhenti (answer and stop): melapor bahwa terdapat N anggota dan berhenti. Setelah menjawab, program Anda harus berhenti . (Perhatikan bahwa instans lainnya di program Anda masih mungkin lanjut berjalan untuk beberapa hari sebelum mereka menjawab dan berhenti.)

Jika ada instans di program Anda yang salah menjawab nilai N , melanggar protokol yang diberikan di atas, menggunakan lebih dari 500 hari, ataupun melebihi batas waktu/memori (per proses), Anda akan menerima *Not correct*.

Jika tidak, program Anda akan diberikan (*Partially*) *Correct* dan dinilai berdasarkan D : jumlah hari maksimum yang dibutuhkan oleh instans mana pun untuk menjawab. Untuk nilai penuh, kamu harus menyelesaikan setiap kasus uji dengan $D \leq 61$ dan $V \leq 1$. Lihat bagian Subsoal dan Penilaian untuk detailnya.

Flushing. Jika kamu tidak menggunakan *template* yang disediakan, pastikan untuk melakukan **flush** pada keluaran setelah mencetak setiap baris, atau program Anda berisiko dinilai *Not correct*. Pada Python, ini dilakukan secara otomatis jika kamu menggunakan `input()` untuk membaca baris. Di C++, `cout << endl;` juga melakukan **flush** selain mencetak baris baru; jika menggunakan `printf`, gunakan `fflush(stdout)`.

Batasan

- $1 \leq N \leq 100$.
- $1 \leq M \leq 100\,000$.
- Anda dapat menggunakan paling banyak 500 hari.

Subsoal dan Penilaian

Program Anda akan diuji ke beberapa kasus uji yang dikelompokkan ke dalam subsoal. Untuk mendapatkan nilai untuk suatu subsoal, Anda harus menyelesaikan semua tes di dalamnya dengan benar.

- Subsoal 0 [0 poin]:** Contoh (Anda dapat menulis bilangan bulat manapun selama $0 \leq V \leq 1\,000\,000\,000$).
- Subsoal 1 [11 poin]:** $M \leq 100$, dan N anggota memiliki ID $0, 1, \dots, N - 1$.
- Subsoal 2 [12 poin]:** $1 \leq N \leq 2$.

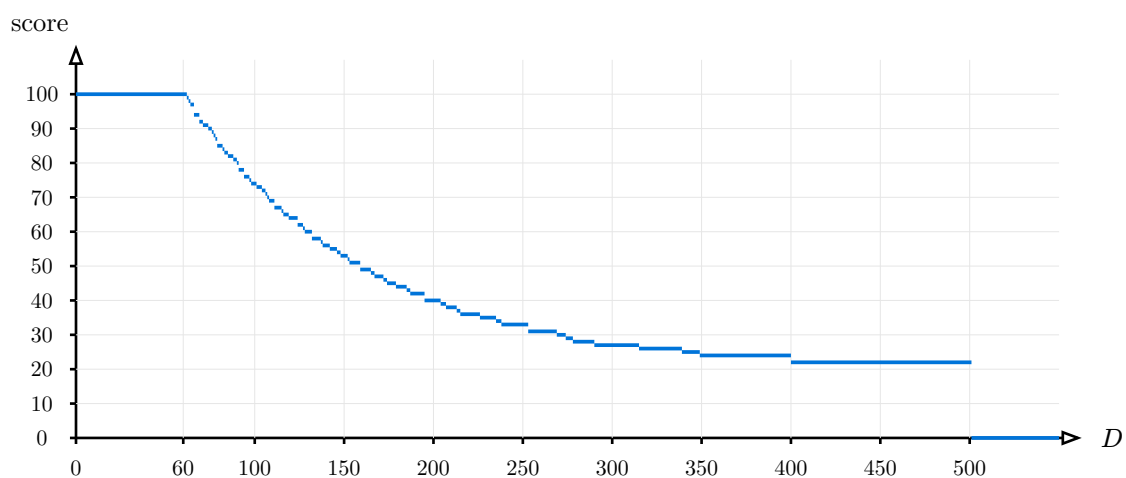
- **Subsoal 3 [22 poin]:** $M \leq 8000$, and Anda dapat menulis bilangan bulat manapun selama $0 \leq V \leq 1000000000$.
- **Subsoal 4 [55 poin]:** Tidak ada batasan tambahan.

Pada subsoal 1, 2, dan 4, Anda hanya dapat menulis $V = 0$ atau $V = 1$ dalam setiap tindakan menulis.

Misalkan X_s adalah poin maksimum untuk subsoal s (sesuai yang di atas), dan D_s adalah jumlah hari terbanyak yang digunakan oleh salah satu program Anda pada tes di subsoal s . Maka:

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

Nilai dari skor_s dibulatkan ke bilangan bulat terdekat, dan total skor Anda adalah jumlah dari skor-skor tersebut. Untuk mendapatkan nilai penuh, Anda perlu $D \leq 61$ dan $V \leq 1$ pada setiap kasus uji.



Figur 1 : Total skor, dengan asumsi setiap subsoal diselesaikan dengan D maksimum yang sama.

Contoh

Contoh pertama. Setiap pasang kolom menunjukkan komunikasi antara grader dan suatu instans

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				

Contoh kedua.

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

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

Penjelasan

Contoh Pertama. Kita memiliki $N = 5$ anggota dengan ID berurutan 0, 1, 2, 3, 4 dan $M = 100$ (berlaku untuk subsoal 1, 3, dan 4). Solusi i sesuai dengan anggota dengan ID i . Interaksi di atas hanya ditunjukkan sebagai kemungkinan urutan operasi yang legal dan **tidak** dimaksudkan sebagai strategi yang efisien maupun masuk akal; ini hanya ditampilkan untuk mengilustrasikan cara kerja protokolnya.

Contoh Kedua. Kita memiliki $N = 2$ anggota, dengan ID 0 dan 3, serta $M = 8000$ (berlaku untuk subsoal 2, 3, dan 4). Di hari pertama, anggota dengan ID 0 menulis 0 di lokasi 0 (tidak ada perubahan), dan anggota dengan ID 3 menulis 1 di lokasi 2.

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

Di hari kedua, ID 0 menulis 1 di lokasi 1, dan ID 3 membaca di lokasi yang sama. Perhatikan bahwa pembacaan terjadi pada siang hari, sebelum penulisan di malam hari. Jadi ID 3 masih melihat angka 0.

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

Di hari ketiga, mereka berdua membaca lokasi 2, di mana angka 1 tertulis.

Di hari keempat, ID 0 menjawab bahwa ada 2 anggota (benar), sementara ID 3 membaca angka 1 di lokasi 1. ID 0 segera keluar dan tidak lanjut lagi.

Di hari $D = 5$, anggota yang tersisa menjawab $N = 2$ dengan benar.

Uji

Untuk memfasilitasi pengujian solusi Anda, kami menyediakan alat sederhana yang dapat Anda unduh dari CMS. Alat ini bersifat opsional. Catat bahwa **grader** resmi di CMS berbeda dengan alat uji ini.

Untuk menggunakan alat ini, Anda memerlukan berkas masukan. Anda dapat menggunakan contoh masukan yang disediakan `census.input0.txt` dan `census.input1.txt`, atau membuat sendiri. Berkas masukan harus dimulai dengan jumlah anggota N dan kemungkinan ID M yang diikuti oleh satu baris dengan N angka yang menentukan ID dari anggota perkumpulan.

Untuk program Python, misalnya `census.py` (biasanya dijalankan dengan `pypy3 census.py`), jalankan alat uji sebagai berikut:

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

Untuk program C++, *compile* solusi Anda:

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

dan kemudian jalankan alat uji:

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

Catat bahwa di permasalahan ini keluaran digunakan untuk berkomunikasi dengan *grader*, maka keluaran jangan digunakan untuk *debugging*. Anda dapat menggunakan keluaran error (`stderr`). Di C++ Anda dapat menggunakan `cerr << msg << endl;`. Di Python Anda dapat menggunakan `print(msg, file=sys.stderr)`.

Alat uji akan membaca dan menyajikan pesan-pesan `stderr` untuk semua instans program Anda. Catat bahwa untuk alasan teknis hasil dapat tercetak secara tidak sinkron dengan yang lainnya.