

## D. Απογραφή (census)

Χρονικό όριο: 1 δευτερόλεπτο

Όριο μνήμης: 128 MiB

Μία ελάχιστη γνωστή πληροφορία για το Cesenatico είναι ότι αποτελεί έδρα μιας μυστικής αδελφότητας  $N$  προγραμματιστριών. Αυτή η αδελφότητα είναι εξαιρετικά μυστικοπαθής · κανένα μέλος δεν γνωρίζει κανένα άλλο. Κάθε μέλος έχει ένα μοναδικό ID: έναν μη αρνητικό ακέραιο  $I$ .

Η μόνη επικοινωνία μεταξύ των μελών είναι έμμεση, μέσω αριθμών χαραγμένων με κιμωλία σε διαφορετικές τοποθεσίες στην πόλη. Κάθε 100 χρόνια, η αδελφότητα πραγματοποιεί μια απογραφή για να μετρήσει τα μέλη της. Αφού ολοκληρωθεί η απογραφή, κάθε μέλος πρέπει να γνωρίζει τον συνολικό αριθμό των μελών της αδελφότητας.

Η απογραφή διαρκεί πολλές ημέρες. Κάθε ημέρα, κάθε μέλος που συμμετέχει ακόμα στη διαδικασία, επιλέγει και εκτελεί ακριβώς μία ενέργεια: **διαβάζει**, **γράφει** ή **σταματάει** να συμμετέχει στη διαδικασία.

- Αν ένα μέλος επιλέξει να **διαβάσει**, επιλέγει μια τοποθεσία  $P$ . Κατά τη διάρκεια της ημέρας, επισκέπτεται την τοποθεσία  $P$  και διαβάζει τον αριθμό που είναι γραμμένος εκεί.
- Αν ένα μέλος επιλέξει να **γράψει**, επιλέγει μια τοποθεσία  $P$  και έναν αριθμό  $V$ . Αργά το βράδυ, επισκέπτεται την τοποθεσία  $P$  και αλλάζει τον αριθμό που ήταν γραμμένος εκεί σε  $V$ . Επειδή έχει ήδη νυχτώσει, δεν μπορεί να διαβάσει τον παλιό αριθμό πριν γράψει τον καινούργιο.
- Αν ένα μέλος επιλέξει να **σταματήσει**, δεν εκτελεί καμία περαιτέρω ενέργεια τις επόμενες ημέρες.

Αν ένα μέλος δει ένα άλλο να γράφει έναν αριθμό, θα μπορούσε να την αναγνωρίσει. Επομένως, απαγορεύεται αυστηρά δύο ή περισσότερα μέλη να επιλέξουν να γράψουν στην ίδια τοποθεσία την ίδια ημέρα. (Δεν υπάρχει τέτοιος περιορισμός για το διάβασμα, καθώς μπορεί να γίνει διακριτικά.)

Αν ένα ή περισσότερα μέλη διαβάζουν από μια τοποθεσία όπου ένα άλλο μέλος θέλει να γράψει την ίδια ημέρα, όλες οι αναγνώσεις πραγματοποιούνται πριν από την εγγραφή.

Πώς πρέπει η αδελφότητα να σχεδιάσει τη διαδικασία της απογραφής της, ώστε να ελαχιστοποιήσει τον αριθμό των ημερών μέχρι όλοι να μάθουν τον σωστό αριθμό μελών;

### Υλοποίηση

⇒ Αυτό είναι ένα διαδραστικό πρόβλημα, στο οποίο ένας άγνωστος αριθμός στιγμιότυπων ( $1 \leq N \leq 100$ ) του προγράμματός σας, θα εκτελεστούν ταυτόχρονα. Κάθε στιγμιότυπο προσομοιώνει ένα μέλος της κοινωνίας.

Υπάρχουν  $10^{18}$  τοποθεσίες. Ο αριθμός  $P$  μιας τοποθεσίας πρέπει να ικανοποιεί  $0 \leq P < 10^{18}$ . Αρχικά, η τιμή που είναι γραμμένη σε όλες τις τοποθεσίες είναι  $V = 0$ .

Η νέα τιμή  $V$  που γράφεται σε μια τοποθεσία πρέπει πάντα να είναι ένας ακέραιος τέτοιος ώστε  $0 \leq V \leq 10^9$ . Στα περισσότερα υποπροβλήματα, το  $V$  μπορεί να είναι μόνο 0 ή 1. Δες την ενότητα **Βαθμολόγηση** για περισσότερες λεπτομέρειες.

Όταν ξεκινά ένα στιγμιότυπο του προγράμματός σας, θα πρέπει πρώτα να διαβάσει μια γραμμή με δύο ακέραιους,  $I$  και  $M$  ( $0 \leq I \leq M - 1$ ): το μοναδικό ID του μέλους της κοινωνίας που αντιπροσωπεύεται

από αυτό το στιγμιότυπο και τον συνολικό αριθμό των πιθανών IDs. Μέσα σε κάθε αρχείο δοκιμής (test case), όλα τα στιγμιότυπα, θα λάβουν την ίδια τιμή  $M$  και διακριτές τιμές  $I$ . Σημειώσε ότι μπορεί να υπάρχουν IDs που δεν έχουν ανατεθεί σε κανένα μέλος.

Στη συνέχεια, για κάθε ημέρα στη διαδικασία απογραφής, το πρόγραμμά σας θα πρέπει να επιλέγει την ενέργεια που θέλει να εκτελέσει και να εκτυπώνει μια γραμμή ανάλογα:

Ενέργεια	Σημασία
$r \ P$	<b>Ανάγνωση</b> από την τοποθεσία $P$ . Μετά την εκτύπωση αυτής της γραμμής, το πρόγραμμά σας πρέπει να διαβάσει μια γραμμή με την τρέχουσα τιμή που είναι γραμμένη στο $P$ .
$w \ P \ V$	<b>Εγγραφή</b> στην τοποθεσία $P$ της νέας τιμής $V$ . Αν πολλαπλά στιγμιότυπα γράψουν στο ίδιο $P$ την ίδια ημέρα, θα λάβεις την ετυμηγορία <i>Not correct</i> . Εκτός από τα παραδείγματα και το υποπρόβλημα 3, πρέπει να γράφεις $0 \leq V \leq 1$ . Δες την ενότητα Βαθμολόγηση.
$! \ N$	<b>Απάντηση και σταμάτημα</b> : ανέφερε ότι υπάρχουν $N$ μέλη και σταμάτα να συμμετέχεις στην απογραφή. Μετά την απάντηση, <b>το πρόγραμμά σας πρέπει να τερματίσει κανονικά</b> . (Σημείωσε ότι άλλα στιγμιότυπα του προγράμματός σας μπορεί να συνεχίσουν να τρέχουν για επιπλέον ημέρες πριν απαντήσουν και τερματίσουν και αυτά.)

Αν κάποιο στιγμιότυπο του προγράμματός σας απαντήσει με λάθος τιμή  $N$ , παραβιάσει το πρωτόκολλο, χρησιμοποιήσει περισσότερες από 500 ημέρες ή υπερβεί το όριο χρόνου/μήνης (ανά διεργασία), η υποβολή σας θα κριθεί ως *Not correct* για το συγκεκριμένο αρχείο δοκιμής.

Διαφορετικά, το πρόγραμμά σας θα είναι *(Partially) Correct* στο αρχείο δοκιμής και θα βαθμολογηθεί με βάση την τιμή  $D$ : τον μέγιστο αριθμό ημερών που χρειάστηκε οποιοδήποτε από τα στιγμιότυπα για να απαντήσει. Για τη πλήρη βαθμολογία, πρέπει να επιλύσετε κάθε αρχείο δοκιμής με  $D \leq 61$  και  $V \leq 1$ . Δες την ενότητα **Βαθμολόγηση** για λεπτομέρειες.

**Flushing.** Αν δεν χρησιμοποιείς τα παρεχόμενα πρότυπα (templates), βεβαιωθείτε ότι κάνετε flush την τυπική έξοδο (standard output) μετά την εκτύπωση κάθε γραμμής, αλλιώς το πρόγραμμά σας μπορεί να κριθεί ως *Not correct*. Στην Python, αυτό συμβαίνει αυτόματα αν χρησιμοποιείς `input()` για να διαβάσετε γραμμές. Στην C++, το `cout << endl`; κάνει flush εκτός από την εκτύπωση μιας νέας γραμμής · αν χρησιμοποιείς `printf`, χρησιμοποίησε `fflush(stdout)`.

## Περιορισμοί

- $1 \leq N \leq 100$ .
- $1 \leq M \leq 100\,000$ .
- Μπορείς να χρησιμοποιήσετε το πολύ 500 ημέρες.

## Βαθμολογία

Το πρόγραμμά σας θα ελεγχθεί σε αρκετά αρχεία δοκιμών ομαδοποιημένα σε υποπροβλήματα. Για να λάβετε τη βαθμολογία για ένα υποπρόβλημα, πρέπει να επιλύσετε σωστά όλα τα αρχεία δοκιμών που περιέχει.

- **Υποπρόβλημα 0 [ 0 πόντοι]**: Παραδείγματα (μπορείτε να γράψετε οποιονδήποτε ακέραιο  $0 \leq V \leq 1\,000\,000\,000$ ).
- **Υποπρόβλημα 1 [11 πόντοι]**:  $M \leq 100$ , και τα  $N$  μέλη έχουν IDs  $0, 1, \dots, N - 1$ .
- **Υποπρόβλημα 2 [12 πόντοι]**:  $1 \leq N \leq 2$ .

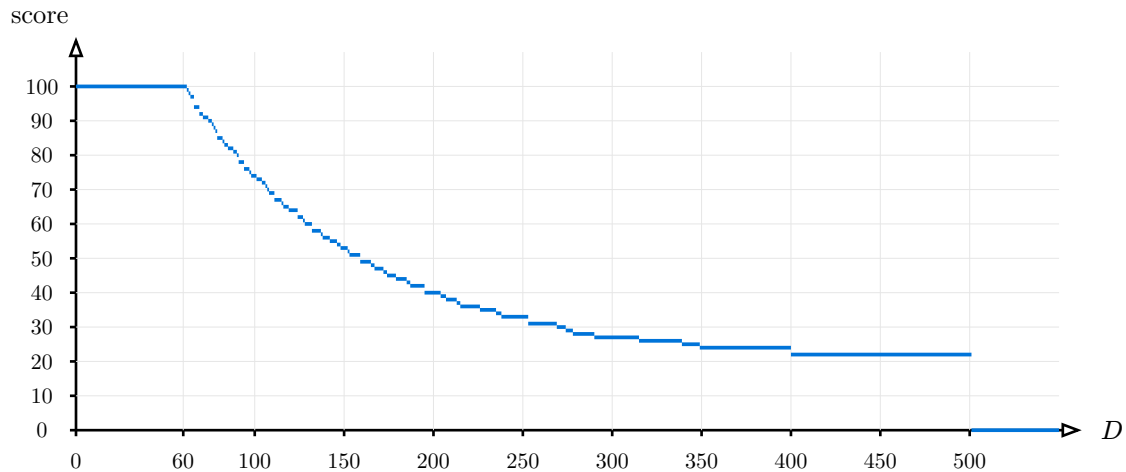
- **Υποπρόβλημα 3 [22 πόντοι]:**  $M \leq 8000$ , και μπορείτε να γράψετε οποιονδήποτε ακέραιο  $0 \leq V \leq 1000000000$ .
- **Υποπρόβλημα 4 [55 πόντοι]:** Χωρίς επιπλέον περιορισμούς.

Στα υποπρόβληματα 1, 2, και 4, μπορείτε να γράψετε μόνο  $V = 0$  ή  $V = 1$  σε κάθε ενέργεια εγγραφής (Write).

Εστω  $X_s$  η μέγιστη βαθμολογία για το υποπρόβλημα  $s$  (που εμφανίζονται παραπάνω), και  $D_s$  ο μεγαλύτερος αριθμός ημερών που χρησιμοποιεί κάποιο από τα προγράμματά σας σε ένα αρχείο δοκιμής στο υποπρόβλημα  $s$ . Τότε:

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

Η τιμή του  $\text{score}_s$  στρογγυλοποιείται στον πλησιέστερο ακέραιο ανά υποπρόβλημα, και το συνολικό σας σκορ είναι το άθροισμα αυτών. Για να πάρετε τη πλήρη βαθμολογία για την άσκηση, χρειάζεστε  $D \leq 61$  και  $V \leq 1$  σε κάθε αρχείο δοκιμής.



Σχήμα 1: Συνολικό σκορ, υποθέτοντας ότι κάθε υποπρόβλημα λύνεται με το ίδιο μέγιστο  $D$ .

## Παραδείγματα εισόδου/εξόδου

Πρώτο παράδειγμα. Κάθε ζεύγος στηλών δείχνει την επικοινωνία μεταξύ του grader και ενός στιγμιότυπου.

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				

Δεύτερο παράδειγμα.

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

## Εξήγηση

**Πρώτο Παράδειγμα.** Έχουμε  $N = 5$  μέλη με συνεχόμενα IDs 0, 1, 2, 3, 4 και  $M = 100$  (έγκυρο για τα υποπροβλήματα 1, 3, και 4). Το στιγμιότυπο  $i$  αντιστοιχεί στο μέλος με ID  $i$ . Η αλληλεπίδραση παραπάνω είναι απλώς μια πιθανή νόμιμη ακολουθία ενεργειών και **δεν** προορίζεται να είναι μια αποτελεσματική ή λογική στρατηγική · εμφανίζεται μόνο για να δείξει πώς λειτουργεί το πρωτόκολλο.

**Δεύτερο Παράδειγμα.** Έχουμε  $N = 2$  μέλη, με IDs 0 και 3, και  $M = 8000$  (έγκυρο για τα υποπροβλήματα 2, 3, και 4). Την πρώτη ημέρα, το μέλος με ID 0 γράφει ένα 0 στην τοποθεσία 0 (καμία αλλαγή), και το μέλος με ID 3 γράφει ένα 1 στην τοποθεσία 2.

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

Τη δεύτερη ημέρα, το ID 0 γράφει ένα 1 στην τοποθεσία 1, και το ID 3 διαβάζει από την ίδια τοποθεσία. Σημείωσε ότι η ανάγνωση συμβαίνει κατά τη διάρκεια της ημέρας, πριν από την εγγραφή το βράδυ. Επομένως, το ID 3 βλέπει ακόμα ένα 0.

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

Την τρίτη ημέρα, και οι δύο διαβάζουν την τοποθεσία 2, όπου είναι γραμμένο ένα 1.

Την τέταρτη ημέρα, το ID 0 απαντά ότι υπάρχουν 2 μέλη (σωστό), ενώ το ID 3 διαβάζει το 1 στην τοποθεσία 1. Το ID 0 τερματίζει αμέσως μετά από αυτό και δεν συμμετέχει τις επόμενες ημέρες.

Τελικά, την ημέρα  $D = 5$ , το εναπομείναν μέλος απαντά επίσης σωστά ότι  $N = 2$ .

## Ελεγχος (Testing)

Για να διευκολύνουμε τον έλεγχο της λύσης σας, παρέχουμε ένα απλό εργαλείο που μπορείτε να κατεβάσετε από το CMS. Η χρήση του εργαλείου είναι προαιρετική. Σημείωσε ότι ο επίσημος grader στο CMS είναι διαφορετικός από αυτό το εργαλείο ελέγχου.

Για να χρησιμοποιήσετε το εργαλείο, χρειάζεστε ένα αρχείο εισόδου. Μπορείτε να χρησιμοποιήσετε τα παρεχόμενα παραδείγματα εισόδου `census.input0.txt` και `census.input1.txt`, ή να φτιάξετε τα δικά σας. Το αρχείο εισόδου πρέπει να ξεκινά με τον αριθμό των μελών  $N$  και τα πιθανά IDs  $M$ , ακολουθούμενο από μια γραμμή με  $N$  αριθμούς που καθορίζουν τα IDs των μελών της κοινωνίας.

Για προγράμματα Python, έστω `census.py` (που κανονικά εκτελείται ως `python3 census.py`) τρέξε το εργαλείο ελέγχου ως εξής:

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

Για προγράμματα C++, πρώτα κάντε compile τη λύση σας:

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

και μετά τρέξε το εργαλείο ελέγχου:

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

Σημειώστε ότι σε αυτό το πρόβλημα η τυπική έξοδος (standard output) χρησιμοποιείται για την επικοινωνία με τον grader, οπότε δεν πρέπει να χρησιμοποιείται για debugging. Αντίθετα, μπορείτε να χρησιμοποιήσετε την standard error output (stderr). Στην C++ μπορείτε να χρησιμοποιήσετε `cerr << msg << endl;`. Στην Python μπορείτε να χρησιμοποιήσετε `print(msg, file=sys.stderr)`.

Το εργαλείο ελέγχου θα διαβάσει και θα παρουσιάσει αυτά τα μηνύματα stderr μαζί με τα ερωτήματα που εκτελούνται από όλα τα στιγμιότυπα του προγράμματός σας. Σημειώστε ότι για τεχνικούς λόγους μπορεί να εμφανίζονται ελαφρώς εκτός συγχρονισμού μεταξύ τους.