

## D. Väestönlaskenta (census)

Aikaraja: 1 sekuntia

Muistiraja: 128 MiB

Eräs vähemmän tunnettu fakta Cesenaticosta on, että siellä asustaa salainen  $N$  naispuolisen tietojenkäsittelijän muodostama seura. Seura on todellakin hyvin salainen; kukaan jäsenistä ei tiedä muiden jäsenten henkilöllisyyttä. Jokaisella jäsenellä on uniikki ID: ei-negatiivinen kokonaisluku  $I$ .

Jäsenten ainoa kommunikaatio on epäsuoraa, liidulla eri puolille kaupunkia raapustettujen numeroiden välityksellä. Joka 100. vuosi seura suorittaa väestönlaskennan jäsentensä määrän selvittämiseksi. Kun väestönlaskenta on ohi, jokaisen jäsenen tulisi tietää seuran jäsenten kokonaismäärä.

Väestönlaskenta kestää useita päiviä. Joka päivä jokainen prosessiin vielä osallistuva jäsen valitsee ja suorittaa täsmälleen yhden toiminnon: **lukeminen**, **kirjoittaminen** tai **lopettaminen**.

- Jos jäsen valitsee **lukemisen**, hän valitsee sijainnin  $P$ . Päiväsaikaan hän vierailee sijainnissa  $P$  ja lukee siellä olevan numeron.
- Jos jäsen valitsee **kirjoittamisen**, hän valitsee sijainnin  $P$  ja numeron  $V$ . Myöhään illalla hän vierailee sijainnissa  $P$  ja muuttaa siellä olevan numeron arvoon  $V$ . Koska on jo pimeää, hän ei voi lukea vanhaa numeroa ennen uuden kirjoittamista.
- Jos jäsen valitsee **lopettamisen**, hän ei enää suorita toimintoja tulevinä päivinä.

Jos yksi jäsen näkee toisen kirjoittavan numeron, hän saa selville tämän henkilöllisyyden. Tämän vuoksi on ankarasti kiellettyä, että kaksi tai useampi jäsen valitsee kirjoittamisen samaan sijaintiin samana päivänä. (Lukemiselle ei ole vastaavaa rajoitusta, sillä se voidaan tehdä huomaamattomasti.)

Jos yksi tai useampi jäsen lukee sijainnista, johon toinen jäsen haluaa kirjoittaa samana päivänä, kaikki lukemiset tapahtuvat ennen kirjoittamista.

Miten seuran tulisi suunnitella väestönlaskentaprosessi siten, että minimoidaan päivien määrä, jonka jälkeen kaikki tietävät oikean jäsenmäärän?

### Toteutus

⇒ Tämä on interaktiivinen ongelma, jossa suoritetaan samanaikaisesti tuntematon määrä ( $1 \leq N \leq 100$ ) ohjelmasi tapahtumia. Jokainen tapahtuma simuloi yhtä seuran jäsentä.

Sijainteja on  $10^{18}$  kappaletta. Sijainnin  $P$  täytyy toteuttaa  $0 \leq P < 10^{18}$ . Aluksi kaikissa sijainneissa oleva arvo on  $V = 0$ .

Sijaintiin kirjoitettavan uuden arvon  $V$  on aina oltava kokonaisluku, siten että  $0 \leq V \leq 10^9$ . Useimmissa osatehtävissä  $V$  voi olla vain 0 tai 1. Katso Pisteytys-osio saadaksesi lisätietoja.

Kun ohjelmasi tapahtuma käynnistyy, sen tulee ensin lukea rivi, jossa on kaksi kokonaislukua,  $I$  ja  $M$  ( $0 \leq I \leq M - 1$ ): tämän tapauksen edustaman seuran jäsenen uniikki ID sekä mahdollisten ID:iden kokonaismäärä. Kunkin testitapauksen sisällä kaikki tapaukset saavat saman  $M$ :n arvon ja uniikit  $I$ :n arvot. Huomaa, että voi olla ID:itä, joita ei ole jaettu yhdellekään jäsenelle.

Sitten, jokaisena väestönlaskennan päivänä, ohjelmasi tulee valita suoritettava toiminto ja tulostaa vastaava rivi:

Toiminto	Merkitys
$r \ P$	<b>Lue</b> sijainti $P$ . Tämän rivin tulostamisen jälkeen ohjelmasi tulee lukea rivi, jossa on sijaintiin $P$ kirjoitettu nykyinen arvo.
$w \ P \ V$	<b>Kirjoita</b> sijaintiin $P$ uusi arvo $V$ . Jos usea tapahtuma kirjoittaa samaan $P$ :hen samana päivänä, saat ilmoituksen <i>Not correct</i> . Lukuun ottamatta esimerkkejä ja osatehtävää 3, sinun tulee kirjoittaa $0 \leq V \leq 1$ ; katso Pisteytys-osio.
$! \ N$	<b>Vastaa ja lopeta:</b> ilmoita, että jäseniä on $N$ kappaletta ja lopeta osallistuminen väestönlaskentaan. Vastaamisen jälkeen <b>ohjelmasi tulee sulkeutua normaalisti</b> . (Huomaa, että muut tapaukset ohjelmastasi saattavat jatkaa toimintaansa vielä useita päiviä ennen kuin ne vastaavat ja sulkeutuvat.)

Jos joku ohjelmasi tapahtumista antaa väärän  $N$ :n arvon, rikkoo protokollaa, käyttää yli 500 päivää tai ylittää (prosessikohtaisen) aikarajan/muistirajan, palautuksesi tulkitaan *Not correct* kyseisessä testitapauksessa.

Muussa tapauksessa ohjelmasi on (*Partially*) *Correct* kyseisessä testitapauksessa ja se pisteytetään arvon  $D$  perusteella: se on maksimimäärä päiviä, jonka yksikään tapahtuma käytti vastataksaan. Täysiä pisteitä varten sinun on ratkaistava jokainen testitapausta siten, että  $D \leq 61$  ja  $V \leq 1$ . Katso Pisteytys-osio saadaksesi lisätietoja.

**Tulosteen tyhjennys (Flushing).** Jos et käytä tarjottuja pohjia, varmista, että tyhjennät vakiotulosteen jokaisen rivin tulostamisen jälkeen, muuten ohjelmasi saatetaan tulkita *Not correct*. Pythonissa tämä tapahtuu automaattisesti, jos käytät `input()`-funktioita lukemiseen. C++:ssa `cout << endl`; tyhjentää tulosteen rivinvaihdon lisäksi; jos käytät `printf`-funktioita, käytä `fflush(stdout)`.

## Rajoitukset

- $1 \leq N \leq 100$ .
- $1 \leq M \leq 100\,000$ .
- Voit käyttää enintään 500 päivää.

## Pisteytys

Ohjelmasi testataan useilla testitapauksilla, jotka on ryhmitelty osatehtäviin. Saadaksesi pisteet osatehtävästä, sinun on ratkaistava oikein kaikki siihen kuuluvat testit.

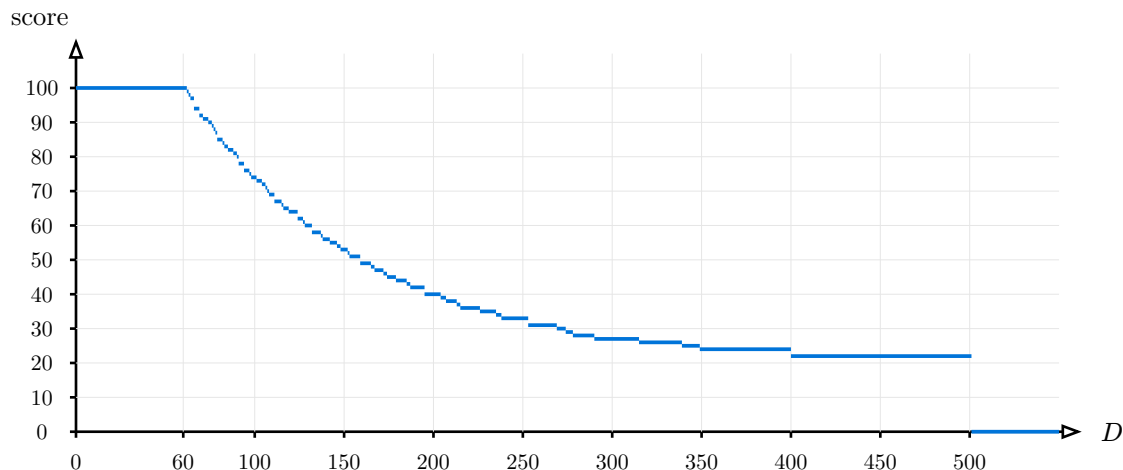
- **Osatehtävä 0 [ 0 pistettä]:** Esimerkit (voit kirjoittaa minkä tahansa kokonaisluvun  $0 \leq V \leq 1\,000\,000\,000$ ).
- **Osatehtävä 1 [11 pistettä]:**  $M \leq 100$ , ja  $N$  jäsenellä on ID:t  $0, 1, \dots, N - 1$ .
- **Osatehtävä 2 [12 pistettä]:**  $1 \leq N \leq 2$ .
- **Osatehtävä 3 [22 pistettä]:**  $M \leq 8000$ , ja voit kirjoittaa minkä tahansa kokonaisluvun  $0 \leq V \leq 1\,000\,000\,000$ .
- **Osatehtävä 4 [55 pistettä]:** Ei lisärajoituksia.

**Osatehtävissä 1, 2 ja 4 voit kirjoittaa vain  $V = 0$  tai  $V = 1$  kussakin Kirjoita-toiminnossa.**

Olkoon  $X_s$  osatehtävän  $s$  maksimipistemäärä (näkyä yllä) ja  $D_s$  suurin päivien määrä, jonka mikään ohjelmistasi käyttää testissä osatehtävässä  $s$ . Tällöin:

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

$\text{score}_s$ :n arvo pyöristetään lähimpään kokonaislukuun per osatehtävä, ja kokonaispisteesi on näiden summa. Täysiä pisteitä varten tarvitset  $D \leq 61$  ja  $V \leq 1$  jokaisessa testitapauksessa.



Kuva 1: Kokonaispisteet olettaen, että jokainen osatehtävä ratkaistaan samalla maksimi- $D$ :llä.

## Esimerkit

Ensimmäinen esimerkki. Jokainen sarakepari näyttää kommunikaation pisteyttäjän ja tapahtuman välillä.

Pis.	Tap. 0	Pis.	Tap. 1	Pis.	Tap. 2	Pis.	Tap. 3	Pis.	Tap. 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				

Toinen esimerkki.

Pisteyttäjä	Tapahtuma 0
0 8000	
	w 0 0
	w 1 1
	r 2
1	
	! 2

Pisteyttäjä	Tapahtuma 1
3 8000	
	w 2 1
	r 1
0	
	r 2
1	
	r 1
1	
	! 2

## Selitys

**Ensimmäinen esimerkki.** Meillä on  $N = 5$  jäsentä peräkkäisillä ID:illä 0, 1, 2, 3, 4 ja  $M = 100$  (pätee osatehtäville 1, 3 ja 4). Ratkaisu  $i$  vastaa jäsentä, jolla on ID  $i$ . Yllä oleva vuorovaikutus on vain yksi mahdollinen laillinen toimintosekvenssi, eikä sen ole tarkoitus olla tehokas tai järkevä strategia; se on näytetty vain havainnollistamaan, miten protokolla toimii.

**Toinen esimerkki.** Meillä on  $N = 2$  jäsentä, ID:t 0 ja 3, ja  $M = 8000$  (pätee osatehtäville 2, 3 ja 4). Ensimmäisenä päivänä ID 0 kirjoittaa 0 sijaintiin 0 (ei muutosta), ja ID 3 kirjoittaa 1 sijaintiin 2.

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

Toisena päivänä ID 0 kirjoittaa 1 sijaintiin 1, ja ID 3 lukee saman sijainnin. Huomaa, että lukeminen tapahtuu päiväsaikaan, ennen illan kirjoitusta. Siten ID 3 näkee edelleen 0:n.

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

Kolmantena päivänä he molemmat lukevat sijainnin 2, johon on kirjoitettu 1.

Neljäntenä päivänä ID 0 vastaa, että jäseniä on 2 (oikein), kun taas ID 3 lukee 1:n sijainnista 1. ID 0 poistuu heti tämän jälkeen eikä osallistu tulevinä päivinä.

Lopuksi, päivänä  $D = 5$ , jäljellä oleva jäsen vastaa myös oikein  $N = 2$ .

## Testaustyökalu

Testauksen helpottamiseksi tarjoamme yksinkertaisen työkalun, jonka voit ladata CMS:stä. Työkalun käyttö on vapaaehtoista. Huomaa, että CMS:n virallinen pisteyttäjä on erilainen kuin tämä testi-työkalu.

Työkalun käyttöön tarvitset syötetiedoston. Voit käyttää annettuja esimerkkisyötteitä `census.input0.txt` ja `census.input1.txt`, tai tehdä oman. Syötetiedoston tulee alkaa jäsenten määrällä  $N$  ja mahdollisilla ID:illä  $M$ , jota seuraa rivi  $N$  numeroa, jotka määrittävät seuran jäsenten ID:t.

Python-ohjelmille, sanotaan `census.py` (ajetaan normaalisti komennolla `pypy3 census.py`), aja testityökalu seuraavasti:

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

C++-ohjelmille, käännä ensin ratkaisusi:

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

ja aja sitten testityökalu:

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

Huomaa, että tämän ohjelman standarditulostetta käytetään kommunikointiin pisteyttäjän kanssa, joten sitä ei pidä käyttää debukkaukseen. Sen sijasta, voit käyttää standardi virhetulostetta (stderr). C++:ssa voit käyttää use `cerr << msg << endl`; Pythonissa voit käyttää `print(msg, file=sys.stderr)`.

Testaustyökalu lukee ja esittää nämä stderr viestit yhdessä kaikkien ohjelmatapaustesi pyyntöjen kanssa. Huomaa, että teknisistä syistä ne saattavat näyttää hieman epäsynkassa olevilta keskenään.