

## D. Decentralizované sčítanie ľudu (census)

Časový limit: 1 sekúnd

Pamäťový limit: 128 MiB

V Cesenaticu už dlho funguje Klub Samotárskeho Programátoriek, tak dlho, že ani nevedia, koľko ich je. Keďže sú samotárske, nestretávajú sa. Ale keďže sú aj *klub programátoriek*, naprogramovali si protokol samotárskej komunikácie, ktorý im pomôže zistiť niekoľko veľmi dôležitých informácií (najmä to, koľko ich je). Ale program zjedla M(i/y)ška, tak ho musíte nahradiť vy.

Klub Samotárskeho Programátoriek tvorí  $N$  členiek. Členky nepoznajú číslo  $N$ , ale vedia, že to je najviac  $M$ . Každá členka tiež pozná svoje unikátne ID: nezáporné celé číslo medzi 0 a  $M - 1$ .

V Cesenaticu existuje  $10^{18}$  tajných lokácií, očíslovaných od 0 po  $10^{18} - 1$  (vrátane). Každá obsahuje malú tabuľu, na ktorú sa zmestí práve jedno číslo v rozsahu od 0 po  $V$  (vrátane). Na začiatku je na každej číslo 0.

Klub Samotárskeho Programátoriek chce počas niekoľkých ďalších – najviac však 500 – dní použiť tieto tabule na zistenie počtu svojich členiek. Každý deň vie každá členka urobiť jednu z nasledujúcich troch akcií:

- Môže si ísť *ráno* **prečítať** číslo na tabuľi číslo  $0 \leq P < 10^{18}$ .
- Môže ísť *večer* **napísať** nové číslo na tabuľu číslo  $0 \leq P < 10^{18}$ . Keďže je už tma, staré číslo si prečítať nevie.
- Môže spokojne skonštatovať, že pozná počet členiek  $N$ , a akciu **ukončiť**. Ak raz akciu ukončí, už nemôže čítať ani písať ďalšie čísla.

Keďže (ako sme už viackrát spomenuli, ale ony sú na to naozaj citlivé) to je spolok *samotárskeho* programátoriek, nechcú sa navzájom stretnúť. Konkrétne, ak by dve členky išli písať číslo na rovnakú tabuľu v rovnaký večer, mohli by na seba naraziť, a to je príliš sociálnej interakcie! Preto je prísne zakázané, aby dve alebo viaceré členky písali na rovnakú tabuľu v ten istý deň. (Pre čítanie takéto obmedzenie neplatí – programátorky majú silné okuliare a číslo čítajú z bezpečnej vzdialenosti bez rizika sociálneho kontaktu).

Všimnite si tiež, že čítania prebehnú ráno a písania večer – každý deň všetky čítania prebehnú pred písaniami.

Zistite (naprogramujte), ako by mali programátorky postupovať, aby bez zbytočného sociálneho kontaktu zistili počet členiek v klube za čo najmenší počet dní.

### Implementácia

⇒ Toto je interaktívna úloha, v ktorej bude súčasne spustený neznámy počet inštancií ( $1 \leq N \leq 100$ ) vášho programu. Každá inštancia simuluje jednu členku spoločnosti.

Váš program má k dispozícii  $I$  a  $M$ : unikátne ID členky spolku a horný odhad na počet členiek klubu.

Počet členiek  $N$  ani najväčšie číslo, ktoré sa zmestí napísať na tabuľu, neviete. Rôzne testovacie sady (viď sekciu Bodovanie) dovoľujú rôzne rozsahy čísel na tabuliach, pričom vo všetkých platí, že

najväčšie číslo napísané na tabuli nemôže presiahnuť  $10^9$ . Vo väčšine sád platí, že na tabule sa dá napísať iba 0 alebo 1.

Propomínáme, že na začiatku je na každej z  $10^{18}$  tabúl napísaná hodnota 0.

Keď sa váš program spustí, dostane najskôr na vstupe čísla  $I$  a  $M$  oddelené medzerou. Je zaručené, že  $0 \leq I < M$ , a že skutočná hodnota  $N$  nepresiahne  $M$ . Všimnite si, že môže platiť  $M > N$  a teda budú existovať ID nepridelené žiadnej členke.

Následne váš program simuluje akcie programátorky s ID  $I$ . Môže postupne vypísať najviac 500 akcií. Akcie sa popisujú nasledovne:

Akcia	Význam
$r \ P$	Programátorka si ráno <b>prečíta</b> hodnotu na tabuli číslo $P$ Po vypísaní tohto riadku dostane váš program na vstupe riadok obsahujúci jediné celé číslo: hodnota na tabuli číslo $P$ v ráno toho dňa.
$w \ P \ V$	Programátorka <b>napíše</b> hodnotu $V$ na tabuľu číslo $P$ Ak by viacero programátoriek písalo na rovnaké $P$ v ten istý deň, dostanete <i>Not correct</i> (nesprávne). $V$ musí byť celé číslo nepresahujúce $10^9$ , pričom vo všetkých sádach okrem príkladov a sady 3 musí platiť $0 \leq V \leq 1$ ; viď sekciu Bodovanie.
$! \ N$	Programátorka <b>ukončí</b> akciu a vyhlási, že existuje $N$ členiek. Po odpovedi <b>by mal váš program normálne skončiť</b> . (Iné inštancie vášho programu zodpovedajúce iným členkám môžu pokračovať aj ďalšie dni.)

Pokiaľ ktorákoľvek inštancia vášho programu odpovie nesprávnu hodnotu  $N$ , použije nesprávnu formu inštrukcie, použije viac ako 500 dní alebo prekročí časový/pamäťový limit (v rámci jedného behu programu), vaše riešenie bude v danom teste vyhodnotené ako *Not correct* (nesprávne).

V opačnom prípade bude vaše riešenie v danom teste vyhodnotené ako *(Partially) correct* (čiastočne správne) a dostane body na základe  $D$  – maximálneho počtu dní, ktoré niektorá z členiek (inšancií programu) potrebovala na správnu odpoveď. Na plný počet bodov môžete na tabule písať iba hodnoty  $V \leq 1$  a použiť maximálne 61 dní, viď sekciu Bodovanie.

## Flushing

Ak nepoužívate poskytnuté šablóny, uistite sa, že po vypísaní každej inštrukcie flushnete výstup, inak môže byť váš program vyhodnotený ako *Not correct* (nesprávne). V Pythone sa to deje automaticky, ak na čítanie riadkov používate `input()`. V C++ `cout << endl;` robí flush automaticky. Ak používate `printf`, musíte ho nasledovať `fflush(stdout)`.

## Obmedzenia

- $1 \leq N \leq 100$ .
- $1 \leq M \leq 100\,000$ .
- Váš program môže použiť najviac 500 dní.

## Bodovanie

Váš program bude otestovaný na viacerých sádach vstupov, v ktorých existujú nasledujúce obmedzenia:

- **Podúloha 0 [ 0 bodov]:** Príklady (môžete napísať akékoľvek celé číslo  $0 \leq V \leq 1\,000\,000\,000$ ).
- **Podúloha 1 [11 bodov]:**  $M \leq 100$ , ID členiek sú  $0, 1, \dots, N - 1$

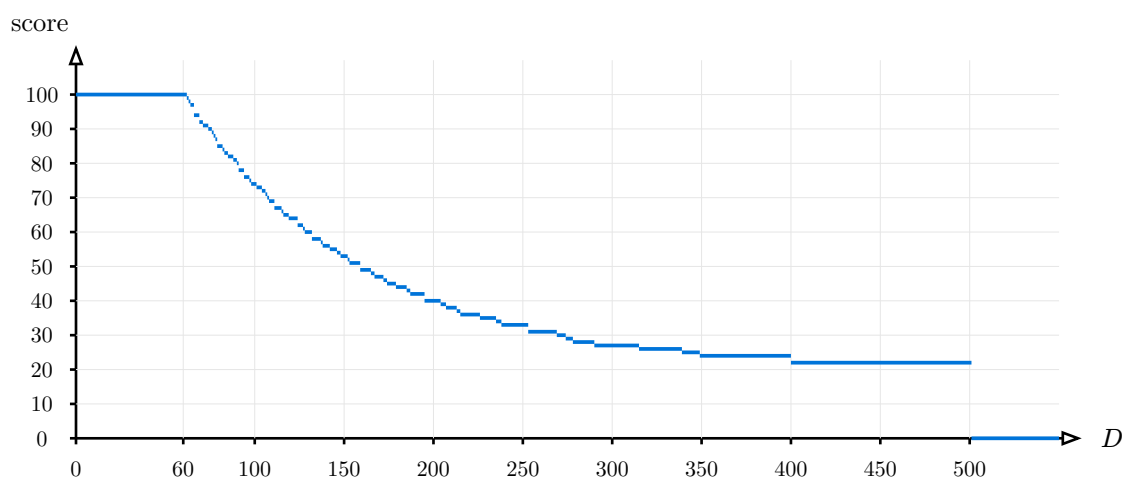
- **Podúloha 2 [12 bodov]:**  $1 \leq N \leq 2$ .
- **Podúloha 3 [22 bodov]:**  $M \leq 8000$  a môžete napísať akékoľvek celé číslo  $0 \leq V \leq 1\,000\,000\,000$ .
- **Podúloha 4 [55 bodov]:** Žiadne ďalšie obmedzenia.

V podúlohách 1, 2 a 4 zároveň môžu tabule obsahovať iba  $V = 0$  alebo  $V = 1$ .

Aby ste získali body za sadu, váš program musí byť na každom vstupe v tejto sade ohodnotený ako (čiasť) správny, pričom počet bodov závisí na maximálnom počte dní, ktoré váš program použil vo vstupoch tejto sade. Ak sa v sade  $s$  dá získať maximálne  $X_s$  bodov a maximum dní, ktoré váš program v sade  $s$  použil, je  $D_s$ , potom počet bodov za túto sadu je nasledovná hodnota zaokrúhlená na najbližšie celé číslo.

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

Vaše celkové skóre je súčtom týchto hodnôt pre všetky sady. Aby ste získali plný počet bodov za úlohu, potrebujete  $D \leq 61$  a  $V \leq 1$  na každom vstupe.



Obr. 1: Celkové skóre za predpokladu, že každá sada  $s$  je vyriešená s rovnakým  $D_s = D$ .

## Príklady

Prvý príklad. Každý pár stĺpcov zobrazuje komunikáciu medzi testovačom a jednou inštanciou.

Testo.	Inšt. 0	Testo.	Inšt. 1	Testo.	Inšt. 2	Testo.	Inšt. 3	Testo.	Inšt. 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				

Druhý príklad.

Testovač	Inšancia 0
0 8000	
	w 0 0
	w 1 1
	r 2
1	
	! 2

Testovač	Inšancia 1
3 8000	
	w 2 1
	r 1
0	
	r 2
1	
	r 1
1	
	! 2

## Vysvetlenie

**Prvý príklad.** Máme  $N = 5$  členiek, ktorých ID sú 0, 1, 2, 3, 4, a  $M = 100$  (mohol by sa objaviť v sadách 1, 3 a 4). Inšancia  $i$  zodpovedá členke s ID  $i$ . Interakcia vyššie je len jedna možná legálna postupnosť operácií a **nie** je mienená ako efektívna alebo rozumná stratégia; je zobrazená len na ilustráciu toho, ako protokol funguje.

**Druhý príklad.** Máme  $N = 2$ , členky s ID 0 a 3, a  $M = 8000$  (mohol by sa objaviť v sadách 2, 3 a 4). V prvý deň členka s ID 0 zapíše 0 na tabuľu číslo 0 (nič tak nezmení) a členka s ID 3 zapíše 1 na tabuľu číslo 2. Tabuľe tak večer vyzerajú nasledovne:

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

V druhý deň členka s ID 0 zapíše 1 na tabuľu číslo 1 a ID 3 si prečíta hodnotu z tej istej tabuľe. Všimnite si, že čítanie prebehne cez deň, pred zápisom večer. Preto programátorka s ID 3 stále vidí 0.

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

V tretí deň obidve prečítajú číslo 1 z tabuľe 2.

V štvrtý deň programátorka s ID 0 (správne) odpovie, že existujú 2 členky, kým tá s ID 3 prečíta čísla 1 z tabuľe 1.

Programátorka s ID 0 na to končí, a v nasledujúcich dňoch sa už komunikácie nezúčastňuje. Sčítanie je dokončené, keď v deň  $D = 5$  zostávajúca členka taktiež správne odpovie, že  $N = 2$ .

## Testovanie

Na uľahčenie testovania vášho riešenia poskytujeme jednoduchý nástroj, ktorý si môžete stiahnuť z CMS. Použitie tohto nástroja je voliteľné. Oficiálny testovač na CMS sa od testovacieho nástroja líši.

Na použitie tohto nástroja potrebujete súbor so vstupom, pričom môžete použiť vstupy s príkladmi `census.input0.txt` a `census.input1.txt`, alebo si vytvoriť vlastné. Vstupný súbor by mal začínať riadkom s počtom členiek  $N$  a počtom možných ID  $M$ , oddelenými medzerou, nasledovaný riadkom s  $N$  číslami špecifikujúcimi ID členiek spoločnosti. Tieto ID by mali byť odlišné a v rozsahu od 0 po  $M - 1$ .

Pre programy v Pythone, povedzme `census.py` (zvyčajne spúšťané ako `pypy3 census.py`), spustite testovací nástroj nasledovne:

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

Pre programy v C++, najprv skompilujte svoje riešenie:

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

a potom spustite testovací nástroj:

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

Všimnite si, že v tejto úlohe sa štandardný výstup používa na komunikáciu s testovačom, preto by sa nemal používať na debugovanie. Namiesto toho môžete použiť štandardný chybový výstup (`stderr`). V C++ môžete použiť `cerr << msg << endl;`. V Pythone môžete použiť `print(msg, file=sys.stderr)`.

Testovací nástroj prečíta a zobrazí tieto správy zo `stderr` spolu s akciami vykonanými všetkými inštan-  
ciami vášho programu. Všimnite si, že z technických príčin sa môžu zobrazovať mierne asynchrónne.