

D. Մարդահամար (census)

Ժամանակի սահմանափակում: 1 վայրկյան

Հիշողության սահմանափակում: 128 MiB

Չեզեղատիկոյի մասին քիչ հայտնի մի փաստ էլ այն է, որ այնտեղ ապրում է N կին ինֆորմատիկներից բաղկացած մի գաղտնի հասարակություն: Այս հասարակությունը իսկապես շատ գաղտնի է. ոչ մի անդամ չգիտի մյուսների ինքնությունը:

Անդամների միջև միակ շփումն անուղղակի է՝ քաղաքի տարբեր վայրերում կալիճով գրված թվերի միջոցով: Յուրաքանչյուր 100 տարին մեկ հասարակությունը մարդահամար է անցկացնում՝ իր անդամների քանակը հաշվելու համար: Մարդահամարի ավարտից հետո յուրաքանչյուր անդամ պետք է իմանա հասարակության անդամների ընդհանուր թիվը:

Մարդահամարը տեղի է ունենում մի քանի օրերի ընթացքում: Ամեն օր այն անդամները, ովքեր դեռ մասնակցում են գործընթացին, ընտրում և կատարում են ճիշտ մեկ գործողություն՝ **կարդալ**, **գրել** կամ **դադարեցնել** մասնակցությունը:

- Եթե անդամն ընտրում է **կարդալ**, նա ընտրում է P վայր: Ցերեկը նա այցելում է P վայր և կարդում այնտեղ գրված թիվը:
- Եթե անդամն ընտրում է **գրել**, նա ընտրում է P վայր և V թիվ: Ուշ երեկոյան նա այցելում է P վայր և այնտեղ գրված թիվը փոխում V -ի: Քանի որ արդեն մուծ է, նա չի կարող կարդալ հին թիվը նորը գրելուց առաջ:
- Եթե անդամն ընտրում է **դադարեցնել**, նա այլևս ոչ մի գործողություն չի կատարում հաջորդ օրերին:

Եթե մեկ անդամը տեսնի, թե ինչպես է մեկ ուրիշը թիվ գրում, նա կարող է իմանալ նրա ինքնությունը: Հետևաբար, խստիվ արգելվում է, որ երկու կամ ավելի անդամներ նույն օրը նույն վայրում գրեն: (Կարդալու համար նման սահմանափակում չկա, քանի որ դա կարելի է անել գաղտնի:)

Եթե մեկ կամ մի քանի անդամներ կարդում են մի վայրից, որտեղ մեկ այլ անդամ ցանկանում է գրել նույն օրը, բոլոր կարդալու գործողությունները տեղի են ունենում գրելուց առաջ:

Ինչպե՞ս պետք է հասարակությունը պլանավորի իր մարդահամարի գործընթացը, որպեսզի նվազագույնի հասցնի օրերի քանակը մինչև բոլորը իմանան անդամների ճիշտ թիվը:

Իրականացում

Սա ինտերակտիվ խնդիր է, որտեղ ձեր ծրագրի անհայտ քանակությամբ պրոցեսներ $\Rightarrow (1 \leq N \leq 100)$ կաշխատեն միաժամանակ: Յուրաքանչյուր պրոցես սիմուլյացնում է հասարակության մեկ անդամի:

Կա 10^{18} վայր: Վայրի P համարը պետք է բավարարի $0 \leq P < 10^{18}$ պայմանին: Սկզբում բոլոր վայրերում գրված արժեքը $V = 0$ է:

Վայրում գրվող նոր V արժեքը միշտ պետք է լինի ամբողջ թիվ՝ $0 \leq V \leq 10^9$: Շատ եկթախնդիրներում V -ն կարող է լինել միայն 0 կամ 1: Մանրամասների համար տե՛ս «Գնահատում» բաժինը:

Երբ ձեր ծրագիրը սկսում է աշխատել, այն պետք է նախ կարդա մեկ տող՝ երկու ամբողջ թվով՝ I և M ($0 \leq I \leq M - 1$): սա տվյալ պրոցեսի (յուրաքանչյուր անդամի համար մեկ պրոցես) ներկայացրած անդամի յուրահատուկ ID-ն է և հնարավոր ID-ների ընդհանուր քանակը: Յուրաքանչյուր թեստի դեպքում բոլոր պրոցեսները կստանան M -ի նույն արժեքը և տարբեր I արժեքներ: Նկատի ունեցեք, որ կարող են լինել ID-ներ, որոնք չեն տրվել ոչ մի անդամի:

Այնուհետև, մարդահամարի յուրաքանչյուր օրվա համար ձեր ծրագիրը պետք է ընտրի կատարելիք գործողությունը և տպի համապատասխան տողը՝

Գործողություն	Իմաստ
$r \ P$	Կարդալ P վայրը: Այս տողը տպելուց հետո ձեր ծրագիրը պետք է կարդա P -ում գրված ընթացիկ արժեքը:
$w \ P \ V$	Գրել P վայրում V նոր արժեքը: Եթե մի քանի պրոցեսներ նույն օրը գրում են նույն P վայրում, դուք կստանաք <i>Սխալ</i> : Բացի օրինակներից և եկթախնդիր 3-ից, պետք է գրեք $0 \leq V \leq 1$; տե՛ս «Գնահատում» բաժինը:
$! \ N$	Պատասխանել և ավարտել : հայտնել, որ կա N անդամ և դադարեցնել մարդահամարին մասնակցությունը: Պատասխանելուց հետո ձեր ծրագիրը պետք է ավարտի աշխատանքը : (Նկատի ունեցեք, որ ձեր ծրագրի մյուս պրոցեսները կարող են շարունակել աշխատել լրացուցիչ օրեր, մինչև նրանք նույնպես պատասխանեն և ավարտեն:)

Եթե ձեր ծրագրի որևէ պրոցես պատասխանի N -ի սխալ արժեք, խախտի նշված պահանջները, օգտագործի 500-ից ավելի օր, կամ գերազանցի ժամանակի/հիշողության սահմանաչափը (մեկ պրոցեսի համար), ձեր լուծումը տվյալ թեստի համար կգնահատվի որպես *Սխալ* :

Հակառակ դեպքում՝ ձեր ծրագիրը կհամարվի (*Մասնակի*) *Ճիշտ* և կգնահատվի՝ հիմնվելով D արժեքի վրա՝ առավելագույն օրեր օգտագործած անդամի օգտագործած օրերի քանակը: Ամբողջական միավոր ստանալու համար դուք պետք է լուծեք յուրաքանչյուր թեստ $D \leq 61$ և $V \leq 1$ պայմաններով: Մանրամասների համար տե՛ս «Գնահատում» բաժինը:

Մաքրում (Flushing): Եթե չեք օգտագործում տրամադրված ձևանմուշները, համոզվեք, որ յուրաքանչյուր տող տպելուց հետո մաքրում եք ստանդարտ ելքը, հակառակ դեպքում ձեր ծրագիրը կարող է գնահատվել որպես *Սխալ* : Python-ում դա տեղի է ունենում ավտոմատ՝ `input()` օգտագործելիս: C++-ում `cout << endl`; տպումից հետո ելքը ելքը մաքրվում է; `printf`-ի դեպքում օգտագործեք `fflush(stdout);`:

Սահմանափակումներ

- $1 \leq N \leq 100$.
- $1 \leq M \leq 100\,000$.
- Կարող եք օգտագործել առավելագույնը 500 օր:

Գնահատում

Ձեր ծրագիրը կստուգվի մի քանի թեստերի վրա, որոնք խմբավորված են ենթախնդիրների մեջ: Ենթախնդրի համար միավորներ ստանալու համար դուք պետք է ճիշտ լուծեք դրանում պարունակվող բոլոր թեստերը:

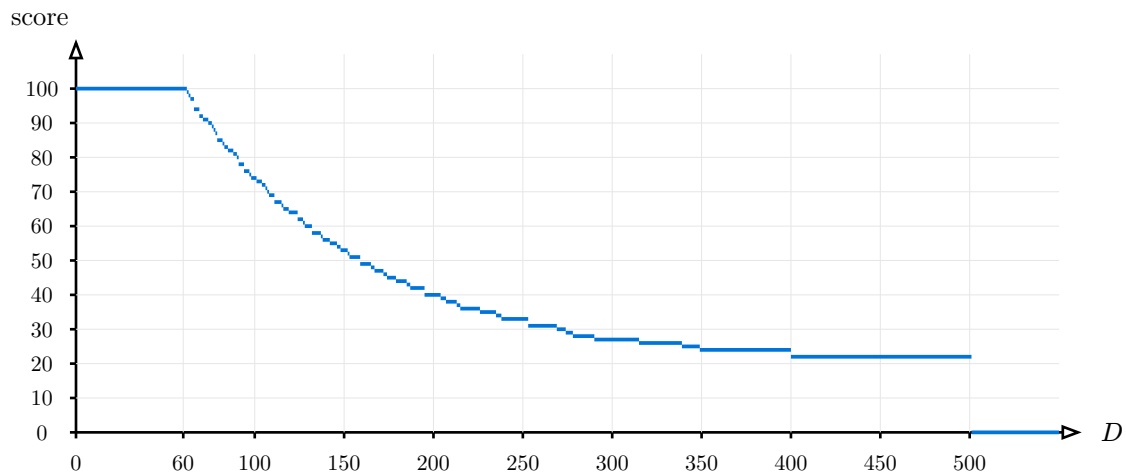
- **Ենթախնդիր 0 [0 միավոր]:** Օրինակներ (կարող եք գրել ցանկացած ամբողջ թիվ $0 \leq V \leq 1000000000$).
- **Ենթախնդիր 1 [11 միավոր]:** $M \leq 100$, և N անդամները ունեն $0, 1, \dots, N - 1$ ID-ներ:
- **Ենթախնդիր 2 [12 միավոր]:** $1 \leq N \leq 2$.
- **Ենթախնդիր 3 [22 միավոր]:** $M \leq 8000$, և կարող եք գրել ցանկացած ամբողջ թիվ $0 \leq V \leq 1000000000$.
- **Ենթախնդիր 4 [55 միավոր]:** Լրացուցիչ սահմանափակումներ չկան:

Ենթախնդիր 1, 2 և 4-ում «Գրել» գործողության ժամանակ կարող եք գրել միայն $V = 0$ կամ $V = 1$:

Թող M_s -ը լինի s ենթախնդրի առավելագույն միավորը (ցույց է տրված վերևում), իսկ D_s -ը՝ s ենթախնդրի բոլոր թեստերում ձեր ծրագրերից որևէ մեկի օգտագործած օրերի առավելագույն քանակը: Այդ դեպքում՝

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

score_s -ի արժեքը կլորացվում է մինչև ամենամոտ ամբողջ թիվը՝ յուրաքանչյուր ենթախնդրի համար, և ձեր ընդհանուր միավորը դրանց գումարն է: Ամբողջական միավոր ստանալու համար դուք պետք է ունենաք $D \leq 61$ և $V \leq 1$ յուրաքանչյուր թեստի համար:



Նկար 1: Ընդհանուր միավոր, ենթադրելով, որ յուրաքանչյուր ենթախնդիր լուծված է նույն առավելագույն D -ով:

Օրինակներ

First example: 5 members, 100 possible IDs

Gra.	Sol. 0	Gra.	Sol. 1	Gra.	Sol. 2	Gra.	Sol. 3	Gra.	Sol. 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				

Second example: 2 members, 8000 possible IDs

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

Բացատրություն

Առաջին օրինակ: Մենք ունենք $N = 5$ անդամ՝ հաջորդական 0, 1, 2, 3, 4 ID-ներով և $M = 100$ (վավեր է 1, 3 և 4 ենթախնդիրների համար): i լուծումը համապատասխանում է i ID ունեցող անդամին: Վերոնշյալ փոխադրեցությունը գործողությունների ընդամենը մեկ հնարավոր օրինական հաջորդականություն է և նախատեսված չէ որպես արդյունավետ կամ խելամիտ ռազմավարություն. այն ցույց է տրված միայն արձանագրության աշխատանքը պատկերելու համար:

Երկրորդ օրինակ: Մենք ունենք $N = 2$ անդամ՝ 0 և 3 ID-ներով, և $M = 8000$ (վավեր է 2, 3 և 4 ենթախնդիրների համար): Առաջին օրը 0 ID ունեցող անդամը 0 է գրում 0 վայրում (ոչ մի փոփոխություն), իսկ 3 ID ունեցող անդամը 1 է գրում 2 վայրում:

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

Երկրորդ օրը, 0 ID-ն 1 է գրում 1 վայրում, իսկ 3 ID-ն կարդում է նույն վայրը: Նկատի ունեցեք, որ կարդալը տեղի է ունենում ցերեկը՝ երեկոյան գրելուց առաջ: Հետևաբար, 3 ID-ն դեռ տեսնում է 0:

location	0	1	2	3	4	...
----------	---	---	---	---	---	-----

value	0	1	1	0	0	...
-------	---	---	---	---	---	-----

Երրորդ օրը, երկուսն էլ կարդում են 2 վայրից, որտեղ գրված է 1:

Չորրորդ օրը, 0 ID-ն պատասխանում է, որ կա 2 անդամ (ճիշտ է), իսկ 3 ID-ն կարդում է 1 վայրում գրված 1-ը: 0 ID-ն անմիջապես ավարտում է աշխատանքը և հաջորդ օրերին չի մասնակցում:

Վերջապես, $D = 5$ օրը, մնացած անդամը նույնպես ճիշտ պատասխանում է՝ $N = 2$:

Թեստավորման գործիք

Ձեր լուծումը թեստավորելը հեշտացնելու համար մենք տրամադրում ենք պարզ գործիք, որը կարող էք ներբեռնել CMS-ից: Գործիքը կամընտիր է: Նկատի ունեցեք, որ CMS-ի պաշտոնական ստուգիչը (grader) տարբերվում է թեստավորման գործիքից:

Գործիքն օգտագործելու համար ձեզ անհրաժեշտ է մուտքային ֆայլ: Կարող եք օգտագործել տրամադրված մուտքային ֆայլերը՝ `census.input0.txt` և `census.input1.txt`, կամ ստեղծել ձերը: Մուտքային ֆայլը պետք է սկսվի անդամների N քանակով և հնարավոր M ID-ներով, որին հաջորդում է N թվերով տող, որը նշում է հասարակության անդամների ID-ները:

Python ծրագրերի համար, օրինակ՝ `census.py` (սովորաբար աշխատում է որպես `pypy3 census.py`), թեստավորման գործիքը գործարկեք հետևյալ կերպ՝

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

C++ ծրագրերի համար նախ կոմպիլացրեք ձեր լուծումը (`census.cpp`).

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

և հետո գործարկեք թեստավորման գործիքը.

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