

D. מפקד אוכלוסין (census)

מגבלת זמן: 1 שניות

מגבלת זיכרון: 128MiB

עובדה פחות מוכרת על צ'זנטיקו היא שהיא ביתה של אגודת סתרים של N מדעניות מחשב. האגודה הזו אכן סודית מאוד; אף חברה לא מכירה שום חברה אחרת. לכל חברה יש מזהה ייחודי: מספר שלם אי-שלילי I .

התקשורת היחידה בין החברות היא עקיפה, דרך מספרים המקושקשים בגיר במיקומים שונים ברחבי העיר. כל 100 שנים, האגודה מבצעת מפקד אוכלוסין כדי לספור את חברותיה. לאחר השלמת המפקד, כל חברה צריכה לדעת מהו המספר הכולל של החברות באגודה.

המפקד נמשך על פני מספר ימים. בכל יום, כל חברה שעדיין לוקחת חלק בתהליך תבחר ותבצע בדיוק פעולה אחת: **לקרוא**, **לכתוב**, או **להפסיק** לקחת בו חלק.

- אם חברה בוחרת **לקרוא** (read), היא בוחרת מיקום P . במהלך היום, היא מבקרת במיקום P וקוראת את המספר שכתוב שם.

- אם חברה בוחרת **לכתוב** (write), היא בוחרת מיקום P ומספר V . בשעות הערב המאוחרות, היא מבקרת במיקום P ומשנה את המספר שהיה כתוב שם ל- V . מכיוון שכבר חשון, היא לא יכולה לקרוא את המספר הישן לפני שהיא כותבת את החדש.

- אם חברה בוחרת **להפסיק** (stop), היא לא תבצע יותר פעולות בימים הבאים.

אם חברה אחת תראה אחרת כותבת מספר, היא עלולה לזהות אותה. לכן, אסור בהחלט לשתי חברות או יותר לבחור לכתוב באותו מיקום באותו יום. (אין הגבלה כזו עבור קריאה, מכיוון שזה יכול להיעשות בדיסקרטיות.)

אם חברה אחת או יותר קוראות ממיקום שבו חברה אחרת רוצה לכתוב באותו יום, כל הקריאות מתבצעות לפני הכתיבה.

כיצד האגודה צריכה לתכנן את תהליך מפקד האוכלוסין שלה כדי למזער את מספר הימים עד שכולן ידעו את מספר החברות המדויק?

מימוש

זוהי בעיה אינטראקטיבית, שבה מספר לא ידוע של עותקים (instances) ($1 \leq N \leq 100$) של התוכנית שלך יורצו במקביל. כל עותק מדמה חברה אחת באגודה. ⇒

ישנם 10^{18} מיקומים. המספר P של מיקום חייב לקיים $0 \leq P < 10^{18}$. בתחילה, הערך שכתוב בכל המיקומים הוא $V = 0$. הערך החדש V שנכתב במיקום חייב להיות תמיד מספר שלם המקיים $0 \leq V \leq 10^9$. ברוב תתי-המשימות, V יכול להיות רק 0 או 1. ראי את חלק הניקוד לפרטים נוספים.

כאשר עותק של התוכנית שלך מתחיל לרוץ, הוא צריך קודם כל לקרוא שורה עם שני מספרים שלמים, I ו- M ($0 \leq I \leq M - 1$): המזהה הייחודי של חברת האגודה המיוצגת על ידי עותק זה והמספר הכולל של מזהים אפשריים. בכל טסט, כל העותקים יקבלו את אותו ערך של M וערכים שונים של I . שימי לב שייתכנו מזהים שלא הוקצו לאף חברה.

לאחר מכן, עבור כל יום בתהליך המפקד, התוכנית שלך צריכה לבחור את הפעולה שהיא רוצה לבצע ולהדפיס שורה בהתאם:

פעולה	משמעות
$r \ P$	לקרוא את מיקום P .
	לאחר הדפסת שורה זו, התוכנית שלך צריכה לקרוא שורה עם הערך הנוכחי שכתוב ב- P .

$w P V$	<p>לכתוב במיקום P את הערך החדש V.</p> <p>אם מספר עותקים כותבים באותו P באותו יום, תקבלי את המשוב <i>Not correct</i>. מלבד הדוגמאות ותת-משימה 3, את חייבת לכתוב $0 \leq V \leq 1$; ראי את סעיף הניקוד.</p>
N !	<p>לענות ולהפסיק: לדווח שיש N חברות ולהפסיק לקחת חלק במפקד.</p> <p>לאחר התשובה, על התוכנית שלך לסיים את הריצה כרגיל. (שימי לב שעותקים אחרים של התוכנית שלך עשויים להמשיך לרוץ במשך ימים נוספים לפני שיענו ויסיימו את הריצה.)</p>

אם עותק כלשהו של התוכנית שלך יענה את הערך הלא נכון של N , יפר את הפרוטוקול, ישתמש ביותר מ-500 ימים, או יחרוג ממגבלת הזמן/זיכרון (עבור כל תהליך בנפרד), ההגשה שלך תשפט כ-*Not correct* עבור הטסט הנוכחי.

אחרת, התוכנית שלך תהיה *Correct (Partially)* על הטסט ותנוקד על בסיס הערך D : המספר המקסימלי של ימים שלקח לעותק כלשהו לענות. לניקוד מלא, עליך לפתור כל טסט עם $D \leq 61$ ו- $V \leq 1$. ראי את חלק הניקוד לפרטים.

Flushing. אם את לא משתמשת בתבניות שסופקו, ודאי שאת מרוקנת (flush) את הפלט הסטנדרטי (stdout) אחרי הדפסת כל שורה, אחרת התוכנית שלך עלולה להישפט כ-*Not correct*. בפיתון, זה קורה אוטומטית אם את משתמשת ב-`input()` כדי לקרוא שורות. ב-`C++`, שימוש ב-`cout << endl; flush` בנוסף להדפסת שורה חדשה; אם את משתמשת ב-`printf`, השתמשי ב-`fflush(stdout)`.

אילוצים

- $1 \leq N \leq 100$
- $1 \leq M \leq 100\,000$
- את יכולה להשתמש ב-500 ימים לכל היותר.

ניקוד

התוכנית שלך תיבדק על מספר טסטים המקובצים לתת-משימות. כדי לקבל את הניקוד עבור תת-משימה, עליך לפתור נכון את כל הטסטים שהיא מכילה.

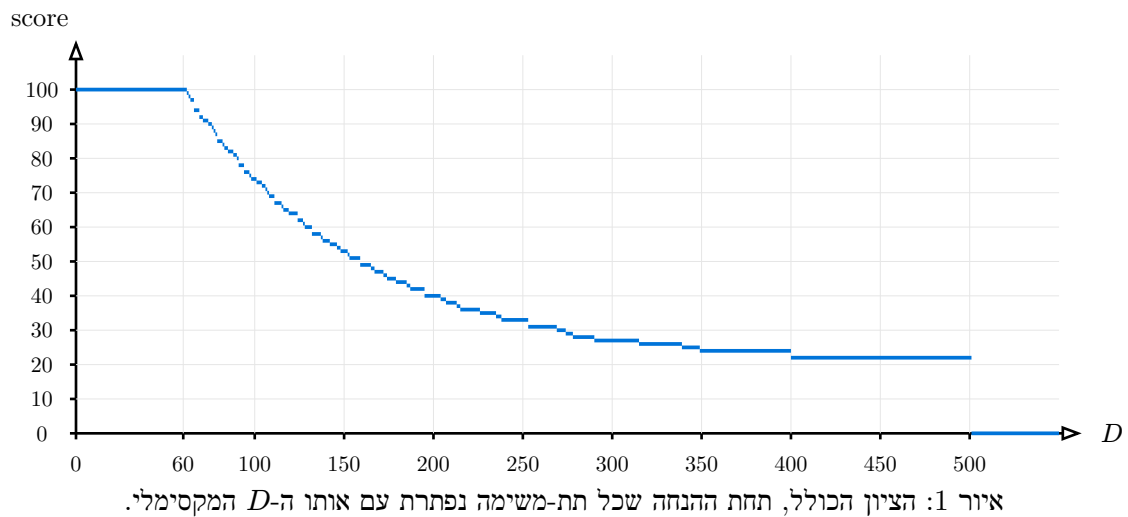
- תת-משימה 0 [0 נקודות]: דוגמאות (את יכולה לכתוב כל מספר שלם $0 \leq V \leq 1\,000\,000\,000$).
- תת-משימה 1 [11 נקודות]: $M \leq 100$, ול- N החברות יש מזהים $0, 1, \dots, N-1$.
- תת-משימה 2 [12 נקודות]: $1 \leq N \leq 2$.
- תת-משימה 3 [22 נקודות]: $M \leq 8000$, ואת יכולה לכתוב כל מספר שלם $0 \leq V \leq 1\,000\,000\,000$.
- תת-משימה 4 [55 נקודות]: ללא מגבלות נוספות.

בתת-משימות 1, 2 ו-4, תוכלי לכתוב רק $V = 0$ או $V = 1$ בכל פעולת כתיבה.

יהי X_s הניקוד המקסימלי עבור תת-משימה s (כמוצג למעלה), ו- D_s המספר הגדול ביותר של ימים בו תוכנית כלשהי שלך השתמשה בטסט בתת-משימה s . אזי:

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

הערך של score_s יעוגל למספר השלם הקרוב ביותר בכל תת-משימה, והציון הכולל שלך הוא סכום הציונים הללו. כדי לקבל את מלוא הניקוד על המשימה, את צריכה שיתקיים $D \leq 61$ ו- $V \leq 1$ בכל טסט.



Examples

First example. Each pair of columns shows the communication between the grader and one instance.

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				

Second example.

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

הסבר

דוגמה ראשונה. יש לנו $N = 5$ חברות בעלות מזהים עוקבים 0, 1, 2, 3, 4 ו- $M = 100$ (תקף לתתי-משימות 1, 3 ו-4). עותק i מתאים לחברה בעלת המזהה i . האינטראקציה למעלה היא רק רצף פעולות חוקי אפשרי אחד והיא אינה נועדה להיות אסטרטגיה יעילה או הגיונית; היא מוצגת רק כדי להמחיש כיצד הפרוטוקול עובד.

דוגמה שנייה. יש לנו $N = 2$ חברות, עם המזהים 0 ו-3, ו- $M = 8000$ (תקף לתתי-משימות 2, 3 ו-4). ביום הראשון, החברה עם המזהה 0 כותבת 0 במיקום 0 (ללא שינוי), והחברה עם המזהה 3 כותבת 1 במיקום 2.

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

ביום השני, החברה עם המזהה 0 כותבת 1 במיקום 1, והחברה עם המזהה 3 קוראת מאותו המיקום. שימי לב שהקריאה מתבצעת בשעות היום, לפני הכתיבה בערב. לכן, החברה עם המזהה 3 עדיין רואה 0.

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

ביום השלישי, שתיהן קוראות את מיקום 2, שם כתוב 1.

ביום הרביעי, החברה עם המזהה 0 עונה שיש 2 חברות (נכון), בעוד החברה עם המזהה 3 קוראת את ה-1 במיקום 1. החברה עם המזהה 0 מיד מסיימת את ריצתה לאחר מכן ולא תשתתף בימים הבאים.

לבסוף, ביום $D = 5$, החברה הנותרת גם כן עונה נכונה שיש $N = 2$.

בדיקות

כדי להקל עליך בבדיקת הפתרון שלך, אנו מספקים כלי פשוט שתוכלי להוריד מה-CMS. השימוש בכלי נתון לבחירתך. שימי לב שהגריידר הרשמי ב-CMS שונה מכלי הבדיקה.

כדי להשתמש בכלי את זקוקה לקובץ קלט. תוכלי להשתמש בדוגמאות הקלט שסופקו `census.input0.txt` ו-`census.input1.txt`, או ליצור כאלה משלך. קובץ הקלט צריך להתחיל במספר החברות N ומספר המזהים האפשריים M , ואז שורה עם N מספרים המציגים את המזהים של חברות האגודה.

עבור תוכניות בפייתון, למשל `census.py` (בדרך כלל מורצות בעזרת `python3 census.py`) הריצי את כלי הבדיקה כך:

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

עבור תוכניות ב-C++, תחילה קמפלי את הפתרון שלך:

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

ואז הריצי את כלי הבדיקה:

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

שימי לב שבבעיה זו פלט הפקודה הרגיל (standard output) משמש לתקשורת עם הגריידר, ולכן אין להשתמש בו לדיבוג. במקום זאת, תוכלי להשתמש בפלט השגיאות (stderr). ב-C++ את יכולה להשתמש ב-`cerr << msg << endl;` בפייתון את יכולה להשתמש ב-`print(msg, file=sys.stderr)`.

כלי הבדיקה יקרא ויצג את הודעות ה-stderr הללו יחד עם השאלות שבוצעו על ידי כל עותקי התוכנית שלך. שימי לב שמסיבות טכניות הן עלולות להופיע בחוסר סנכרון קל אחת עם השנייה.