

D. 人口調査 (census)

実行時間制限: 1 秒

メモリ制限: 128 MiB

チェゼナティコに関するあまり知られていない事実として、 N 名の女性情報科学者からなる秘密の社会がある、という点が挙げられる。この社会は極めて秘密であり、どのメンバーも互いに知らない。各メンバーは相異なる ID を持っており、非負整数 I で表される。

メンバー同士の唯一のやり取りの方法は、町の様々な場所で、チョークを用いて数字を落書きするという、間接的な方法である。この社会では 100 年に一度、総メンバー数を調べる人口調査を行う。人口調査が完了したら、すべてのメンバーが総メンバー数を知っている必要がある。

人口調査は複数の日にまたがる。各日には、まだ人口調査に参加しているメンバーが、**読み込み**、**書き込み**、**終了**のいずれか 1 つの行動を選択し、実行する。

- もし、あるメンバーが**読み込み**を選択した場合、場所 P を選択する。日中に場所 P を訪問し、その場所の落書きに書かれている数字を読む。
- もし、あるメンバーが**書き込み**を選択した場合、場所 P と整数 V を選択する。夜のはじめ頃に場所 P を訪問し、その場所の落書きに書かれている整数を V に変更する。既に外が暗いため、そのメンバーは元々書かれていた整数を読むことはできない。
- もし、あるメンバーが**終了**を選択した場合、そのメンバーは今後、行動をすることはできない。

もし、あるメンバーが「別のメンバーの書き込みの様子」を目撃した場合、書き込んだメンバーのことを認識してしまう。そのため、複数のメンバーが、同じ日・同じ場所で書き込みを行うことは禁止されている (ただし、読み込みに関しては秘密裏に行うことができるため、同様の制限はない)。

もし、1 人または複数のメンバーが読み込みを行い、別のメンバーが同じ日・同じ場所で書き込みを行った場合、読み込みは書き込みの前に発生する。

すべてのメンバーが、総メンバー数を特定するまでにかかる時間をできるだけ短くするには、社会は果たしてどのような人口調査の計画を立てればよいのだろうか？

実装方式

⇒ この問題はインタラクティブ問題である。 N 個 ($1 \leq N \leq 100$) のあなたのプログラムのインスタンスが同時に実行されるが、その数 N は未知である。それぞれのインスタンスは、1 人の社会のメンバーの行動をシミュレーションする。

全部の 10^{18} 個の場所がある。場所を表す整数 P は $0 \leq P < 10^{18}$ を満たす必要がある。ここで、最初すべての場所に整数 $V = 0$ が書かれている。

書き込みの際に、新しい整数 V は $0 \leq V \leq 10^9$ を満たす必要がある。大半の小課題について、 V の値は 0 か 1 である必要がある。詳細は採点方式の項を参照すること。

プログラムの各インスタンスでは、まず開始時に 2 つの整数 I と M ($0 \leq I \leq M - 1$) を読み込む必要がある。それぞれ、当該インスタンスで扱われているメンバーの ID、および利用され得る ID の個数を表す。ここで各メンバーの ID は相異なる。1 つのテストケースでは、各インスタンスは同

じ整数 M を受け取り、相異なる整数 I を受け取る。ここで、どのメンバーにも割り当てられていない ID があるかもしれない。

続いて、人口調査の各日に、プログラムは実行したい行動を選択し、以下の形式で 1 行を出力をする必要がある。

行動	意味
$r\ P$	場所 P での読み込み。 その行を出力した後、あなたのプログラムは、場所 P に書かれている現在の整数からなる 1 行を入力しなければならない。
$w\ P\ V$	場所 P での整数 V の書き込み。 もし複数のインスタンスが同じ場所 P で書き込んだ場合、不正解と判定される。 サンプルケースおよび小課題 3 を除き、あなたは $0 \leq V \leq 1$ を書き込む必要がある。詳細は採点方式の項を参照せよ。
$!\ N$	解答して終了: 総メンバー数が N であることを報告し、人口調査への参加を終了する。 解答後、あなたのプログラムは通常通り終了する必要がある。(なお、他のインスタンスは解答を報告するまでの追加の何日分か、実行を継続するかもしれない。)

もし、いずれかのインスタンスが、間違った N の値を解答した場合、上記のプロトコルに違反した場合、500 日を超えた場合、または (各プロセス毎の) 実行時間制限・メモリ制限を超過した場合、あなたのプログラムは当該テストケースにおいて不正解と採点される。

そうでない場合、あなたのプログラムは、最も長いインスタンスが使った日数 D の値に応じて、正解 (または部分的に正解) と判定される。満点を得るためには、 $D \leq 61$ かつ $V \leq 1$ ですべてのテストケースに正解する必要がある。詳細は採点方式の項を参照すること。

フラッシュ. 提供されたテンプレートを使用しない場合、各行の出力の後に、標準出力をフラッシュしなければならない。さもなくば、あなたのプログラムは不正解と判定される。Python では、入力時に `input()` を用いた場合、自動的にフラッシュが発生する。C++ では、改行の出力時に `cout << endl`; を用いた場合、自動的にフラッシュが発生する。printf を用いる場合、`fflush(stdout)` を使用しなければならない。

制約

- $1 \leq N \leq 100$.
- $1 \leq M \leq 100\,000$.
- 最大で 500 日を使うことができる。

採点方式

あなたの解答は各小課題ごとに評価される。各小課題は複数のテストケースからなる。各小課題について得点を得るためには、その小課題に含まれるすべてのテストケースに正解する必要がある。

- 小課題 0 [0 点]: サンプルケース ($0 \leq V \leq 1\,000\,000\,000$ の範囲の整数を書ける)。
- 小課題 1 [11 点]: $M \leq 100$, N 名のメンバーの ID は $0, 1, \dots, N-1$ である。
- 小課題 2 [12 点]: $1 \leq N \leq 2$.
- 小課題 3 [22 点]: $M \leq 8000$, $0 \leq V \leq 1\,000\,000\,000$ の範囲の整数を書ける。
- 小課題 4 [55 点]: 追加の制約はない。

小課題 1, 2, 4 では、書き込みの際に $V = 0$ か $V = 1$ しか書けないことに注意せよ。

小課題 s の配点を X_s 点とし (前項を参照), 小課題 s のテストケースであなたのプログラムが使用した最大の日数を D_s とする。このとき、得点は以下ようになる。

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

ここで, score_s は各小課題について, 最も近い整数に四捨五入され, あなたの解答の得点はすべての小課題の合計となる. 満点を獲得するには, すべてのテストケースで $D \leq 61$ および $V \leq 1$ を満たす必要がある.

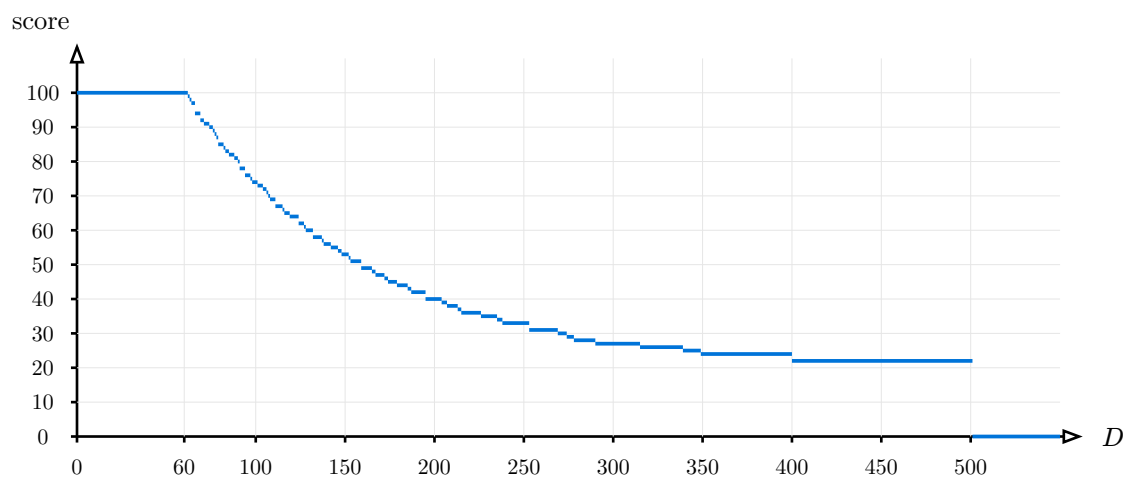


図 1: すべての小課題で, 同じ最大日数 D で解答した場合の合計得点.

入出力例

First example: 5 members, 100 possible IDs

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: 2 members, 8000 possible IDs

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

解説

入出力例 1. この例では，社会に $N = 5$ 名のメンバーがおり，連続する ID 0,1,2,3,4 を持ち，また $M = 100$ である（小課題 1,3,4 の制約を満たす）．Instance i は，ID i を持つメンバーに対応する．上記の入出力例は，正当な行動の例の 1 つであり，最適な行動であるとは限らない．どのようにプロトコルが動くかのみを表している．

入出力例 2. この例では，社会に $N = 2$ 名のメンバーがおり，ID 0,3 を持ち，また $M = 8000$ である（小課題 2,3,4 の制約を満たす）．1 日目には，ID 0 を持つメンバーが場所 0 に整数 0 を書き込み（したがって整数は変更されない），ID 3 を持つメンバーが場所 2 に整数 1 を書き込む．

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

2 日目には，ID 0 を持つメンバーが場所 1 に整数 1 を書き込み，ID 3 が同じ場所で読み込みを行う．ここで読み込みは日中に，書き込みは夜のはじめ頃に起こるため，ID 3 は元の整数 0 を読む．

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

3 日目には、両者が場所 2 から読み込む。整数 1 が書かれている。

4 日目には、ID 0 が「総メンバー数 2 名」と解答し（※正解である）、ID 3 が場所 1 で整数 1 を読み込む。ID 0 のプログラムは終了し、今後の人口調査には参加しない。

最後に 5 日目には、残りのメンバーが「総メンバー数 $N = 2$ 名」と正しく解答する。

テストのためのツール

あなたが解法をテストすることを容易にするため、簡単なツールが CMS からダウンロードできるようになっている。ここで、本ツールを必ずしも使う必要はない。なお、CMS で使用される実際の採点プログラムは、本ツールとは異なる。

本ツールを使うには、まず入力ファイルを指定しなければならない。あなたは、提供されたサンプルテストケースの入力 `census.input0.txt` および `census.input1.txt` を使用しても良く、独自のテストケースを使用しても良い。入力ファイルは総メンバー数 N およびあり得る ID の個数 M から始まり、その後 N 行にわたり、各メンバーの ID を表す整数が続く必要がある。

Python の場合、たとえば `census.py` の場合（通常 `pypy3 census.py` とすれば実行されるが）、以下を実行しなければならない。

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

C++ の場合、まずコンパイルを行わなければならない。

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

続いて、以下のような形式で、テストのためのツールを実行しなければならない。

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

本問題では、採点プログラムとのコミュニケーションで、標準出力が用いられるため、標準出力をデバッグのために使用してはならない。その代わりに、標準エラー出力 (stderr) を用いることができる。C++ では `cerr << msg << endl;` という形式を使うことができる。Python では `print(msg, file=sys.stderr)` という形式を使うことができる。

テストのためのツールでは、stderr の出力も読み、各インスタンスのクエリの出力と一緒に表示する。技術的な理由により、stderr の出力は他のクエリの出力と比べて、少し同期がずれる場合がある。