

D. Census (census)

Time limit: 1 seconds

Memory limit: 128 MiB

N эмэгтэй информатикчдаас бүрдсэн нууц нийгэмлэг Чезенатикод оршдог гэдэг нь төдийлөн мэддэггүй. Энэ нийгэмлэг үнэхээр маш нууцлаг тул гишүүдийн хэн нь ч бусад гишүүдийнхээ хэн болохыг мэддэггүй.

Гишүүд хоорондын цорын ганц харилцаа нь шууд бус хэлбэртэй: хотын янз бүрийн газарт шохойгоор бичсэн тоонуудаар дамжин явдаг.

Зуун жил тутамд нэг удаа нийгэмлэг гишүүдийнхээ тоог мэдэхийн тулд тооллого явуулдаг. Тооллого дууссаны дараа нийгэмлэг нийт хэдэн гишүүнтэй болохыг гишүүн бүр мэдсэн байх ёстой.

Тооллого хэд хэдэн өдөр үргэлжилнэ. Өдөр бүр, тооллогод оролцож байгаа гишүүн бүр дараах үйлдлээс сонгодог: “унших”, “бичих” эсвэл үйл явцад оролцохоо “болих”.

- Хэрэв гишүүн “унших” үйлдлийг сонговол тэрээр P нэг байршлыг сонгоно. Өдрийн цагаар тэр P байршилд очиж, тэнд бичсэн тоог уншина.
- Хэрэв гишүүн “бичих” үйлдлийг сонговол тэрээр P байршил болон V тоог сонгоно. Үдэш тэрээр P байршилд очиж, тэнд бичсэн тоог V болгон өөрчилнө. Нэгэнт харанхуй болсон тул тэрээр шинэ тоог бичихээсээ өмнө хуучин тоог уншиж чадахгүй.
- Хэрэв гишүүн “болих” үйлдлийг сонговол дараагийн өдрүүдэд ямар ч үйлдэл хийхгүй.

Хэрэв нэг гишүүн өөр нэг гишүүнийг тоо бичиж байхыг харвал түүний хэн болохыг мэдэж болзошгүй. Иймээс нэг өдөр хоёр буюу түүнээс олон гишүүн нэг ижил байршилд бичихийг сонгохыг хатуу хориглоно. (Унших үйлдлийн хувьд ийм хязгаарлалт байхгүй, учир нь үүнийг анзаарагдахгүйгээр хийж болно.)

Хэрэв нэг буюу түүнээс олон гишүүн тухайн өдөр өөр нэг гишүүн бичихээр төлөвлөсөн байршлаас унших бол унших бүх үйлдэл бичихээс өмнө явагдана.

Бүх хүн зөв гишүүдийн тоог мэдэхэд шаардагдах өдрийн тоог хамгийн бага байлгахын тулд нийгэмлэг тооллогын үйл явцаа хэрхэн төлөвлөх вэ?

Implementation

⇒ Энэ нь интерактив бодлого бөгөөд таны програмын үл мэдэгдэх тооны хувилбарууд ($1 \leq N \leq 100$)-тай харилцан ажиллана. Хувилбар бүрийг нийгэмлэгийн нэг гишүүнийг дуурайлгасан байна.

Нийт 10^{18} байршил байна. Байршлын дугаар P нь $0 \leq P < 10^{18}$ нөхцөлийг хангана. Эхэндээ бүх байршилд бичигдсэн утга $V = 0$ байна.

Байршил дээр шинээр бичих утга V нь үргэлж $0 \leq V \leq 10^9$ байх бүхэл тоо байна. Ихэнх дэд бодлогод V нь зөвхөн 0 эсвэл 1 байж болно. Дэлгэрэнгүйг Scoring хэсгээс харна уу.

Таны програм нэг хувилбартай ажиллаж эхлэхдээ эхлээд I болон M гэсэн хоёр бүхэл тоо агуулсан мөр унших ёстой ($0 \leq I \leq M - 1$): энэ нь хувилбараар төлөөлүүлж буй нийгэмлэгийн гишүүний давтагдашгүй ID дугаар болон боломжит ID-нуудын нийт тоо байна.

Нэг тестийн тохиолдлын дотор бүх хувилбарууд ижил M утгыг авах бөгөөд I утгууд нь хоорондоо ялгаатай байна. Зарим ID ямар ч гишүүнд оноогдоогүй байж болохыг анхаарна уу.

Дараа нь тооллогын үйл явцын өдөр бүр таны програм гүйцэтгэхийг хүссэн үйлдлээ сонгож, түүнд тохирох нэг мөр хэвлэх ёстой:

Үйлдэл	Утга
$r\ P$	P байршлыг унших . Энэ мөрийг хэвлэсний дараа таны програм P -д одоо бичигдсэн байгаа утгыг агуулсан нэг мөр унших ёстой.
$w\ P\ V$	P байршил дээр шинэ V утгыг бичих . Хэрэв нэг өдөр олон хувилбар ижил P байршилд бичвэл <i>Not correct</i> гэж мэдээлнэ. Жишээнүүд болон 3-р дэд бодлогоос бусад тохиолдолд та $0 \leq V \leq 1$ утга бичих ёстой; Scoring хэсгийг харна уу.
$!\ N$	Хариулаад болих : нийт N гишүүн байна гэж мэдээлээд тооллогод оролцохоо зогсооно. Хариулсны дараа таны програм хэвийн дуусах ёстой . (Таны програмын бусад хувилбарууд хариугаа өгч дуусаад гарахаасаа өмнө нэмэлт хэдэн өдрүүд дээр үргэлжлэн ажиллаж байж болно гэдгийг анхаарна уу.)

Хэрэв таны програмын аль нэг хувилбар N -ийн буруу утгыг хариулах, протокол зөрчих, 500-аас олон өдөр ашиглах, эсвэл (процесс бүрийн) хугацаа/санах ойн хязгаарыг хэтрүүлэх тохиолдолд таны илгээлт тухайн тестийн хувьд *Not correct* гэсэн мэдээлэл гаргах болно.

Үгүй бол таны програм тухайн тест дээр (*Partially*) *Correct* гэж үнэлэгдэж, D утгаар оноо авна: D нь аль нэг хувилбар хариулах хүртлээ зарцуулсан өдрийн хамгийн их тоо юм.

Бүтэн оноо авахын тулд та бүх тестийг $D \leq 61$ ба $V \leq 1$ нөхцөлтэйгээр бодох шаардлагатай. Дэлгэрэнгүйг Scoring хэсгээс харна уу.

Гаралтыг flush хийх. Хэрэв та өгөгдсөн загваруудыг ашиглахгүй байгаа бол мөр бүрийг хэвлэсний дараа стандарт гаралтыг flush хийхээ мартуузай. Үгүй бол таны програм *Not correct* гэсэн мэдээлэл өгч болзошгүй.

Python хэлэнд мөр уншихдаа `input()` ашиглавал энэ нь автоматаар хийгддэг.

C++ хэлэнд `cout << endl;` нь шинэ мөр хэвлэхийн зэрэгцээ flush хийдэг; хэрэв `printf` ашиглаж байгаа бол `fflush(stdout)`-ийг ашиглана.

Constraints

- $1 \leq N \leq 100$.
- $1 \leq M \leq 100\,000$.
- Та хамгийн ихдээ 500 өдөр ашиглаж болно.

Scoring

Таны программыг хэд хэдэн тестийн тохиолдол дээр үнэлэх бөгөөд тэдгээрийг дэд бодлогууд болгон бүлэглэсэн байна. Дэд бодлогын оноог авахын тулд та түүнд багтсан бүх тестийг зөв бодсон байх ёстой.

- Subtask 0 [0 points]:** Бодлогын өгүүлбэрт өгсөн жишээнүүд ($0 \leq V \leq 1\,000\,000\,000$ байх дурын бүхэл тоо бичиж болно).
- Subtask 1 [11 points]:** $M \leq 100$ ба N гишүүдийн ID нь $0, 1, \dots, N - 1$ байна.
- Subtask 2 [12 points]:** $1 \leq N \leq 2$.

- **Subtask 3 [22 points]:** $M \leq 8000$ ба $0 \leq V \leq 1000000000$ байх дурын бүхэл тоо бичиж болно.
- **Subtask 4 [55 points]:** Нэмэлт хязгаарлалт байхгүй.

1, 2, 4-р дэд бодлогууд дээр бичих үйлдэл бүрт зөвхөн $V = 0$ эсвэл $V = 1$ утга бичиж болно.

M_s нь s -р дэд бодлогын хамгийн их оноо (дээр харуулсан), харин D_s нь s -р дэд бодлогын аль нэг тест дээр таны програмуудын аль нэгний ашигласан өдрийн хамгийн их тоо байг. Тэгвэл:

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

score_s утгыг дэд бодлого бүрийн хувьд хамгийн ойр бүхэл тоо руу тоймлох бөгөөд таны нийт оноо эдгээрийн нийлбэр байна.

Бүтэн оноо авахын тулд тест бүр дээр $D \leq 61$ ба $V \leq 1$ байх шаардлагатай.

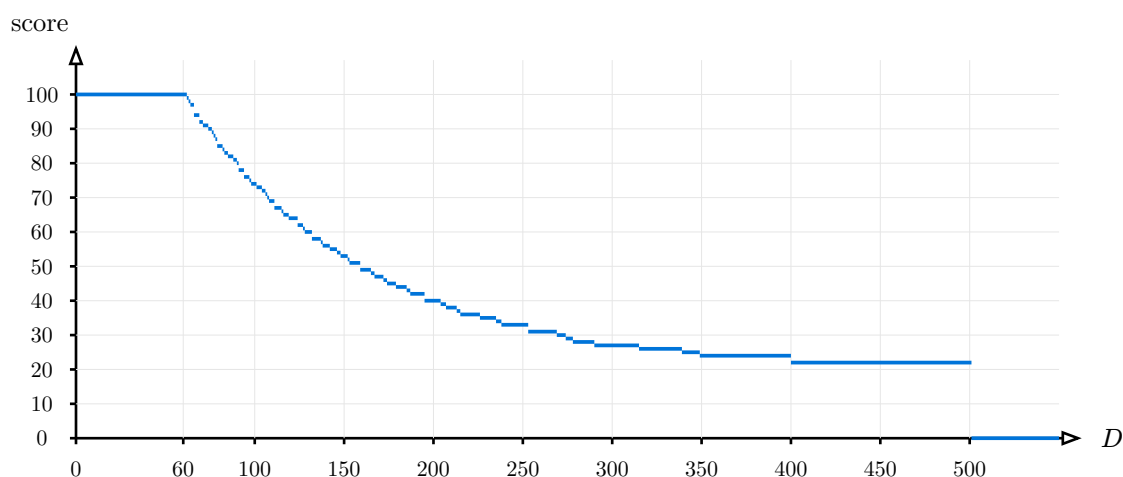


Figure 1: Бүх дэд бодлогыг ижил хамгийн их D утгатайгаар бодсон гэж үзсэн үеийн нийт оноо.

Examples

First example: 5 members, 100 possible IDs

Шалг. Бод. 0	Шалг. Бод. 1	Шалг. Бод. 2	Шалг. Бод. 3	Шалг. Бод. 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

Шалгагч	Бодолт 0
0 8000	
	w 0 0
	w 1 1
	r 2
1	
	! 2

Шалгагч	Бодолт 1
3 8000	
	w 2 1
	r 1
0	
	r 2
1	
	r 1
1	
	! 2

Explanation

Эхний жишээ. Бидэнд дараалсан 0, 1, 2, 3, 4 ID-тай $N = 5$ гишүүн болон $M = 100$ байна (энэ нь 1, 3, 4-р дэд бодлогод тохирно). i -р тохиолдол нь ID нь i байх гишүүнд харгалзана. Дээрх харилцан ажиллагаа нь зөвшөөрөгдөх үйлдлүүдийн нэг боломжит дараалал төдий бөгөөд **үр ашигтай эсвэл оновчтой стратеги байх албагүй**; зөвхөн протокол хэрхэн ажилладгийг харуулах зорилготой.

Хоёр дахь жишээ. Бидэнд 0 ба 3 ID-тай $N = 2$ гишүүн, мөн $M = 8000$ байна (энэ нь 2, 3, 4-р дэд бодлогод тохирно). Эхний өдөр 0 ID-тай гишүүн 0 байршилд 0 бичнэ (өөрчлөлт хийхгүй), харин 3 ID-тай гишүүн 2 байршилд 1 бичнэ.

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

Хоёр дахь өдөр 0 ID-тай гишүүн 1 байршилд 1 бичих бөгөөд 3 ID-тай гишүүн мөн тэр байршилыг уншина. Унших үйлдэл өдрийн цагаар, оройн бичих үйлдлээс өмнө явагддагийг анхаарна уу. Иймээс 3 ID-тай гишүүн 0-г харна.

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

Гурав дахь өдөр тэд хоёулаа 1 гэж бичсэн байгаа 2 байршилыг уншина.

Дөрөв дэх өдөр 0 ID-тай гишүүн нийт 2 гишүүн байна гэж хариулна (зөв), харин 3 ID-тай гишүүн 1 байршил дахь 1-ийг уншина. Үүний дараа 0 ID-тай гишүүн больж, дараагийн өдрүүдэд оролцохгүй.

Эцэст нь, $D = 5$ дахь өдөр үлдсэн гишүүн мөн $N = 2$ гэж зөв хариулна.

Testing Tool

Таны шийдлийг тестлэхэд зориулан CMS-ээс татаж авч болох энгийн хэрэгсэл (tool)-ийг өгсөн. Энэ хэрэгслийг ашиглах нь заавал биш. CMS дээрх албан ёсны grader нь энэхүү testing tool-оос өөр гэдгийг анхаарна уу.

Tool-ийг ашиглахын тулд оролтын файл хэрэгтэй. Та өгөгдсөн `census.input0.txt` болон `census.input1.txt` жишээ файлуудыг ашиглаж болно, эсвэл өөрийн файл үүсгэж болно.

Оролтын файл нь эхлээд гишүүдийн тоо N болон боломжит ID-ны тоо M -оос эхэлнэ. Үүний дараа N ширхэг тоо бүхий мөр байх бөгөөд энэ нь нийгэмлэгийн гишүүдийн ID-нуудыг заана.

Python програмын хувьд, жишээ нь census.py (pypy3 census.py гэж ажиллуулдаг), testing tool-ийг дараах байдлаар ажиллуулна:

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

C++ программын кодоо анхны удаад хөрвүүлэхдээ:

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

дараа нь тестийн хэрэгслийг дараах байдлаар ажиллуулна:

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

Энэ бодлогод стандарт гаралтыг зэрэглэгчтэй харилцахад ашигладаг тул дибаг хийхэд ашиглах ёсгүй гэдгийг анхаарна уу. Үүний оронд та стандарт алдааны гаралтыг (stderr) ашиглаж болно. C++ хэл дээр та cerr << msg << endl; ашиглаж болно. Python хэл дээр та print(msg, file=sys.stderr) ашиглаж болно.

Тестийн хэрэгсэл нь эдгээр stderr мессежийг таны програмын бүх тохиолдлоор гүйцэтгэсэн асуулгын хамт уншиж, харуулах болно. Техникийн шалтгааны улмаас тэдгээр нь хоорондоо бүрэн хамааралгүй байж магадгүйг анхаарна уу.