



Zadatak Zatvor

 2 sek.  1024 MB

Alice i Bob su nepravedno osuđeni na zatvor s maksimalnim osiguranjem. Sada moraju isplanirati bijeg. Da bi to učinili, moraju moći komunicirati što učinkovitije (posebno Alice mora slati dnevne informacije Bobu). Međutim, ne mogu se sastati i mogu razmjenjivati informacije samo putem bilješki napisanih na salvetama. Svaki dan Alice želi poslati novu informaciju Bobu - broj između 0 i $N - 1$. Za svaki ručak Alice dobiva tri salvete i na svaku salvetu napiše broj između 0 i $M - 1$ (mogu se ponavljati) i ostavlja ih na svom sjedalu. Zatim njihov neprijatelj, Charly, uništava jednu od salveta i pomiješa preostale dvije. Konačno, Bob pronalazi dvije preostale salvete i čita brojeve na njima. Mora točno dekodirati izvorni broj koji mu je Alice htjela poslati. Na salvetama je ograničen prostor, pa je M fiksiran. Međutim, Alicein i Bobov cilj je maksimizirati protok informacija, tako da mogu slobodno odabrati što veći broj N . Pomozi Alici i Bobu tako što ćeš za svakog od njih primijeniti strategiju, pokušavajući maksimizirati vrijednost od N .



Implementacijski detalji

Budući da se radi o komunikacijskom problemu, vaš će se program izvršavati u dva odvojena izvršavanja (jedno za Alice i jedno za Boba) koja ne mogu dijeliti podatke ili komunicirati na bilo koji drugi način osim onog opisanog ovdje. Potrebno je implementirati tri funkcije:

```
int setup(int M);
```

Ova funkcija će biti pozvana jednom na početku Alicinog izvršavanja vašeg programa i jednom na početku Bobovog izvršavanja. Dodijeljen je M i funkcija mora vratiti željeni N . Oba poziva funkcije `setup` moraju vratiti isti N .

```
std::vector<int> encode(int A);
```

Ova funkcija implementira Alisinu strategiju. Bit će pozvana s brojem za kodiranje A ($0 \leq A < N$) i mora vratiti tri broja W_1, W_2, W_3 ($0 \leq W_i < M$) koji kodiraju A . Ova funkcija bit će pozvana ukupno T puta - jednom dnevno (vrijednosti A mogu se ponavljati između dana).

```
int decode(int X, int Y);
```

Ova funkcija implementira Bobovu strategiju. Bit će pozvana s dva od tri broja koje vraća `encode` nekim redoslijedom. Mora vratiti istu vrijednost A koju je `encode` primio. Ova će se funkcija također pozivati T puta - što odgovara T pozivima funkcije `encode`; bit će istim redoslijedom. Svi pozivi funkcije `encode` dogodit će se prije svih poziva funkcije `decode`.



Ograničenja

- $M \leq 4300$
- $T = 5000$



Bodovanje

Za određeni podzadatak, udio S bodova koje dobijete ovisi o najmanjem N koji vraća setup na bilo kojem testu u tom podzadatku. Također ovisi o N^* , što je ciljana vrijednost od N koja vam je potrebna da biste dobili puni broj bodova za podzadatak:

- Ako je vaše rješenje krivo na bilo kojem primjeru, tada je $S = 0$.
- Ako je $N \geq N^*$, tada je $S = 1.0$.
- Ako je $N < N^*$, tada je $S = \max\left(0.35 \max\left(\frac{\log(N) - 0.985 \log(M)}{\log(N^*) - 0.985 \log(M)}, 0.0\right)^{0.3} + 0.65 \left(\frac{N}{N^*}\right)^{2.4}, 0.01\right)$.



Podzadaci

| Podzadatak | Bodovi | M | N^* |
|------------|--------|------|---------|
| 1 | 10 | 700 | 82017 |
| 2 | 10 | 1100 | 202217 |
| 3 | 10 | 1500 | 375751 |
| 4 | 10 | 1900 | 602617 |
| 5 | 10 | 2300 | 882817 |
| 6 | 10 | 2700 | 1216351 |
| 7 | 10 | 3100 | 1603217 |
| 8 | 10 | 3500 | 2043417 |
| 9 | 10 | 3900 | 2536951 |
| 10 | 10 | 4300 | 3083817 |



Primjer

Razmotrimo sljedeći primjer s $T = 5$. Ovdje imamo shemu kodiranja gdje Alice šalje tri jednaka broja za kodiranje 0 ili tri različita broja za kodiranje 1. Primijetite da Bob može dekodirati izvorni broj iz bilo koja dva od tri broja koje je Alice poslala.



| Osoba | Poziv funkcije | Povratna vrijednost |
|-------|----------------|---------------------|
| Alice | setup(10) | 2 |
| Bob | setup(10) | 2 |
| Alice | encode(0) | {5, 5, 5} |
| Alice | encode(1) | {8, 3, 7} |
| Alice | encode(1) | {0, 3, 1} |
| Alice | encode(0) | {7, 7, 7} |
| Alice | encode(1) | {6, 2, 0} |
| Bob | decode(5, 5) | 0 |
| Bob | decode(8, 7) | 1 |
| Bob | decode(3, 0) | 1 |
| Bob | decode(7, 7) | 0 |
| Bob | decode(2, 0) | 1 |



Ocjenjivač probnih primjera

Za primjer ocjenjivača, svi pozivi na `encode` i `decode` bit će u istom izvršavanju vašeg programa. Osim toga, `setup` će se pozvati samo jednom (za razliku od dva puta, jednom po izvršavanju, kao u sustavu ocjenjivanja).

Ulaz je samo jedan cijeli broj – M . Zatim će ispisati N koje je vaš `setup` vratio. Zatim će pozvati funkcije `encode` i `decode` ovim redoslijedom T puta sa slučajno generiranim brojevima od 0 do $N - 1$ i slučajno generiranim izborima koja dva od tri broja iz `encode` dati `decode` (i kojim redoslijedom). Ispisat će poruku o pogrešci ako vaše rješenje nije uspjelo.