



Задатак Prison

 2 секунде  1024 MB

Алиса и Боб су неправедно осуђени на затвор са максималним обезбеђењем. Сада морају да испланирају бекство. Да би то урадили, морају бити у могућности да комуницирају што је могуће ефикасније (конкретно, Алиса мора да шаље информације свакодневно Бобу). Међутим, не могу се састати уживо. Једини начин на који могу размењивати информације је путем белешки написаних на салветама. Сваког дана Алиса жели да пошаље Бобу нову информацију - број између 0 и $N - 1$. За сваки ручак, Алиса добија три салвете и на свакој салвети записује број између 0 и $M - 1$ (могуће је да неки број напише више пута) и оставља их на својој столици. Затим, њихов непријатељ, Чарли, уништава једну од салвета и измеша преостале две. Коначно, Боб проналази преостале две салвете и чита бројеве на њима. Мора тачно декодирати оригинални број који је Алиса желела да му пошаље. На салветама је ограничен простор, тако да је M фиксиран. Међутим, циљ Алисе и Боба је да максимизују проток информација, тако да имају право да изаберу N колико год желе велико. Помозите Алиси и Бобу тако што ћете за сваког од њих применити стратегију, покушавајући да максимизујете број N .



Детаљи имплементације

Пошто је ово комуникациони задатак, ваш програм ће се покретати у два одвојена извршавања (једно за Алису и једно за Боба) која не могу делити податке или комуницирати на било који други начин осим оног описаног овде. Потребно је да имплементирате три функције:

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

Она ће бити позвана једном на почетку Алисиног извршавања вашег програма и једном на почетку Бобовог извршавања. Додељује јој се вредност M и мора вратити жељени број N . Оба позива функције `setup` морају вратити исти N .

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

Она имплементира Алисину стратегију. Биће позвана са бројем за енкодирање A ($0 \leq A < N$) и мора вратити три броја W_1, W_2, W_3 ($0 \leq W_i < M$) у које је A енкодиран. Ова функција ће бити позвана укупно T пута - једном дневно (вредности A се могу понављати кроз дане).

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

Она имплементира Бобову стратегију. Биће позвана са два од три броја које враћа `encode` у неком редоследом. Мора вратити исту вредност A коју је `encode` примио.



Ова функција ће такође бити позвана T пута - што одговара T позива функције `encode`; биће истим редоследом. Сви позиви функције `encode` ће се догодити пре свих позива функције `decode`.



Ограничења

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



Бодовање

За одређени подзадатак, део поена S које добијете зависи од најмањег N који враћа `setup` на било ком тест примеру у том подзадатку. Такође зависи од N^* , што је циљна вредност N која вам је потребна да бисте добили пун број поена за подзадатак:

- Ако решење даје нетачан одговор на било којем тест примеру, онда је $S = 0$.
- Ако $N \geq N^*$, онда је $S = 1.0$.
- Ако $N < N^*$, онда је $S = \max\left(0.35 \max\left(\frac{\log(N) - 0.985 \log(M)}{\log(N^*) - 0.985 \log(M)}, 0.0\right)^{0.3} + 0.65 \left(\frac{N}{N^*}\right)^{2.4}, 0.01\right)$.



Подзадаци

Подзадатак	Поени	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



Пример

Размотримо следећи пример у ком је $T = 5$. Посматрамо шему енкодирања где Алиса шаље три једнака броја да енкодира 0 или три различита броја да енкодира 1.



Приметимо да Боб може да декодира оригинални број из било која два од три броја које је Алиса послала.

Извршање	Позив функције	Повратна вредност
Alice	<code>setup(10)</code>	2
Bob	<code>setup(10)</code>	2
Alice	<code>encode(0)</code>	{5, 5, 5}
Alice	<code>encode(1)</code>	{8, 3, 7}
Alice	<code>encode(1)</code>	{0, 3, 1}
Alice	<code>encode(0)</code>	{7, 7, 7}
Alice	<code>encode(1)</code>	{6, 2, 0}
Bob	<code>decode(5, 5)</code>	0
Bob	<code>decode(8, 7)</code>	1
Bob	<code>decode(3, 0)</code>	1
Bob	<code>decode(7, 7)</code>	0
Bob	<code>decode(2, 0)</code>	1



Пример грејдера

За пример оцењивача, сви позиви `encode` и `decode` биће у истом извршавању вашег програма. Додатно, функција `setup` ће бити позвана само једном по извршавању (за разлику од два пута као у систему за оцењивање).

Улаз је само један цео број – M . Затим ће исписати N које је ваш `setup` вратио. Затим ће позвати функције `encode` и `decode` (у том редоследу) T пута са насумично генерисаним бројевима од 0 до $N - 1$ и насумично генерисаним изборима два од три броја из `encode` који ће се проследити у `decode` (и њихов редослед). Исписаће поруку о грешци ако ваше решење није исправно.