



Task Prison

 2 sec.  1024 MB

Алису и Боба несправедливо приговорили к тюремному заключению строгого режима. Теперь им предстоит спланировать побег. Для этого им необходимо общаться максимально эффективно (в частности, Алисе необходимо ежедневно отправлять Бобу информацию). Однако они не могут встречаться и могут обмениваться информацией только посредством записей, написанных на салфетках. Каждый день Алиса хочет отправлять Бобу новую информацию — число от 0 до $N - 1$. За каждым обедом Алиса берёт три салфетки, пишет на каждой из них число от 0 до $M - 1$ (могут быть повторения) и оставляет их на своём стуле. Затем их враг, Чарли, уничтожает одну из салфеток и перемешивает две другие. Наконец, Боб находит две оставшиеся салфетки и читает на них числа. Он должен точно расшифровать исходное число, которое Алиса хотела ему отправить. Место на салфетках ограничено, поэтому M фиксировано. Однако цель Алисы и Боба — максимизировать пропускную способность, поэтому они могут выбрать N как можно большего значения. Помогите Алисе и Бобу, реализовать стратегию для каждого из них, чтобы максимизировать значение N .



Implementation details

Поскольку это задача коммуникации, ваша программа будет запущена в двух отдельных запусках (одно для Алисы и одно для Боба), которые не смогут обмениваться данными или взаимодействовать каким-либо другим способом, кроме описанного здесь. Вам необходимо реализовать три функции:

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

Эта функция будет вызвана один раз при запуске вашей программы Алисой и один раз при запуске программы Бобом. Функция получает M и должна вернуть желаемое N . Оба вызова `setup` должны возвращать одинаковое N .

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

Это реализует стратегию Алисы. Функция будет вызвана с числом, закодированным в A ($0 \leq A < N$), и должна вернуть три числа W_1, W_2, W_3 ($0 \leq W_i < M$), закодировавших A . Эта функция будет вызвана всего T раз — один раз в день (значения A могут повторяться в разные дни).

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

Это реализует стратегию Боба. Функция будет вызвана с двумя из трёх чисел, возвращаемых функцией `encode`, в определённом порядке. Она должна вернуть то



же значение A , что и функция `encode`. Эта функция также будет вызвана T раз — что соответствует T вызовам функции `encode`; они будут выполнены в том же порядке. Все вызовы `encode` будут выполняться перед вызовами `decode`.



Constraints

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



Scoring

Для конкретной подзадачи доля S полученных баллов зависит от наименьшего N , возвращаемого `setup` для любого теста в этой подзадаче. Она также зависит от N^* — целевого значения N , необходимого для получения полного количества баллов за подзадачу:

- Если ваше решение не проходит хотя бы один тест, то $S = 0$.
- Если $N \geq N^*$, то $S = 1, 0$.
- Если $N < N^*$, то $S = \max \left(0.35 \max \left(\frac{\log(N) - 0.985 \log(M)}{\log(N^*) - 0.985 \log(M)}, 0.0 \right)^{0.3} + 0.65 \left(\frac{N}{N^*} \right)^{2.4}, 0.01 \right)$.



Subtasks

Подзадача	Баллы	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



Example

Рассмотрим следующий пример с $T = 5$. Здесь мы имеем схему кодирования, в которой Алиса отправляет три одинаковых числа для кодирования 0 или три



различных числа для кодирования 1. Обратите внимание, что Боб может декодировать исходное число из любых двух из трёх чисел, отправленных Алисой.

Запуск	Вызов функции	Возвращаемое значение
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

Sample grader

В примере оценщика все вызовы `encode` и `decode` будут происходить в одном и том же выполнении программы. Кроме того, `setup` будет вызван только один раз (а не дважды, по одному разу за выполнение, как в системе оценивания).

На вход поступает одно целое число — M . Затем программа выведет N , которые вернули в вашей программе `setup`. Затем она вызовет функции `encode` и `decode` в указанном порядке T раз со случайно сгенерированными числами от 0 до $N - 1$ и случайно сгенерированными вариантами выбора двух из трёх чисел из `encode` для функции `decode` (и в каком порядке). Если решение не удалось, программа выведет сообщение об ошибке.