Task Prison

⏳ 2 sec. 💾 1024 MB

حكم على أليس وبوب ظلما بالسجن شديد الحراسة. والآن عليهما التخطيط لهروبهما. وللقيام بذلك، يجب أن يكونا قادرين على التواصل بأكبر قدر ممكن من الكفاءة (على وجه الخصوص، تحتاج أليس إلى إرسال معلومات يومية إلى بوب). ومع ذلك، لا يمكنهما الالتقاء ويمكنهما فقط تبادل المعلومات من خلال ملاحظات مكتوبة على المناديل.

تريد أليس كل يوم إرسال معلومة جديدة إلى بوب - رقم بين 0 و $N-1$. في كل غداء، تحصل أليس على ثلاثة مناديل وتكتب رقما بين 0 و $M-1$ على كل منديل (قد يكون هناك تكرار) وتتركها على مقعدها. ثم يقوم عدوهما، تشارلي، بتدمير أحد المناديل ويخلط الاثنين الآخرين. أخيرا، يجد بوب المنديلين المتبقيين ويقرأ الأرقام الموجودة عليهما. يجب عليه فك شفرة الرقم الأصلي الذي أرادت أليس إرساله إليه بدقة. هناك مساحة محدودة على المناديل، لذا فإن M ثابت. اوجد استراتيجية بين اليس وبوب تعمل لجميع الارقام من 0 الي $N-1$ لاكبر قيمة ممكنة ل N.

🐾 Implementation details

Since this is a communication problem, your program will be run in two separate executions (one for Alice and one for Bob) that cannot share data or communicate in any way other than the one described here. You need to implement three functions:

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

This will be called once at the start of Alice's execution of your program and once at the start of Bob's execution. It is given M and must return the desired N. Both calls to setup must return the same N.

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

This implements Alice's strategy. It will be called with the number to encode A ($0 \le A < N$) and must return three numbers W_1, W_2, W_3 ($0 \le W_i < M$) that encode A. This function will be called a total of T times – once per day (values of A may repeat between days).

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

This implements Bob's strategy. It will be called with two of the three numbers returned by encode in some order. It must return the same value A that encode received. This function will also be called T times – corresponding to the T calls to encode; they will be in the same order. All calls to encode will happen before all calls to decode.

🐾 Constraints

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

EJOI 2025 Day 1
Task Prison
🌐 **Arabic**

**European Junior Olympiad
in Informatics 2025**
Shumen, Bulgaria | 29 August – 04 September

🐯 Scoring

For a particular subtask, the fraction S of the points you get depends on the smallest N returned by `setup` on any test in that subtask. It also depends on N^*, which is the target value of N that you need to get the full points for the subtask:

- If your solution fails on any test, then $S = 0$.
- If $N \geq N^*$, then $S = 1.0$.
- If $N < N^*$, then $S = \max\left(0.35 \max\left(\frac{\log(N) - 0.985\log(M)}{\log(N^*) - 0.985\log(M)}, 0.0\right)^{0.3} + 0.65\left(\frac{N}{N^*}\right)^{2.4}, 0.01\right)$.

🐯 Subtasks

Subtask	Points	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

Consider the following example with $T = 5$. Here we have an encoding scheme where Alice sends three equal numbers to encode 0 or three distinct numbers to encode 1. Notice that Bob can decode the original number from any two of the three numbers Alice sent.

EJOI 2025 Day 1
Task Prison
🌐 **Arabic**

European Junior Olympiad
in Informatics 2025
Shumen, Bulgaria | 29 August – 04 September

Execution	Function call	Return value
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

🖼 Sample grader

For the sample grader, all calls to `encode` and `decode` will be in the same execution of your program. Additionally, `setup` will be called only once (as opposed to twice, once per execution, as in the grading system).

The input is just a single integer – M. Then it will print out the N your `setup` returned. It will then call functions `encode` and `decode` in this order T times with randomly generated numbers from 0 to $N-1$ and randomly generated choices of which two of the three numbers from `encode` to give to `decode` (and in what order). It will print out an error message if your solution failed.