



Φυλακή

⌚ 2 δευτ. 💾 1024 MB

Η Αλίκη και ο Βασίλης έχουν καταδικαστεί άδικα σε φυλακή υψίστης ασφαλείας. Τώρα πρέπει να σχεδιάσουν την απόδρασή τους. Για να το πετύχουν αυτό, πρέπει να είναι σε θέση να επικοινωνούν όσο το δυνατόν πιο αποτελεσματικά (συγκεκριμένα, η Αλίκη πρέπει να στέλνει καθημερινά πληροφορίες στον Βασίλη). Ωστόσο, δεν μπορούν να συναντηθούν και μπορούν να ανταλλάξουν πληροφορίες μόνο μέσω σημειωμάτων γραμμένων σε χαρτοπετσέτες. Κάθε μέρα η Αλίκη θέλει να στείλει μια νέα πληροφορία στον Βασίλη - έναν αριθμό μεταξύ 0 και $N - 1$. Σε κάθε γεύμα, η Αλίκη παίρνει τρεις χαρτοπετσέτες και γράφει έναν αριθμό μεταξύ 0 και $M - 1$ σε κάθε χαρτοπετσέτα (μπορεί να υπάρχουν επαναλήψεις) και τις αφήνει στη θέση της. Στη συνέχεια, η εχθρός τους, η Τσάρλι, καταστρέφει μία από τις χαρτοπετσέτες και ανακατεύει τις άλλες δύο. Τέλος, ο Βασίλης βρίσκει τις δύο υπόλοιπες χαρτοπετσέτες και διαβάζει τους αριθμούς πάνω τους. Πρέπει να αποκωδικοποιήσει με ακρίβεια τον αρχικό αριθμό που ήθελε να του στείλει η Αλίκη. Υπάρχει περιορισμένος χώρος στις χαρτοπετσέτες, επομένως το M είναι σταθερό. Ωστόσο, ο στόχος της Αλίκης και του Βασίλη είναι να μεγιστοποιήσουν τη ροή πληροφοριών, ώστε να είναι ελεύθεροι να επιλέξουν το μεγαλύτερο δυνατό N . Βοηθήστε την Αλίκη και τον Βασίλη εφαρμόζοντας μια στρατηγική για τον καθένα από αυτούς, προσπαθώντας να μεγιστοποιήσετε την τιμή του N .



Λεπτομέρειες υλοποίησης

Δεδομένου ότι πρόκειται για πρόβλημα επικοινωνίας, το πρόγραμμά σας θα τρέξει σε δύο ξεχωριστές εκτελέσεις (μία για την Αλίκη και μία για τον Βασίλη) που δεν μπορούν να μοιραστούν δεδομένα ή να επικοινωνήσουν με οποιονδήποτε άλλο τρόπο εκτός από αυτόν που περιγράφεται εδώ. Πρέπει να υλοποιήσετε τρεις συναρτήσεις:

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

Αυτή η συνάρτηση θα κληθεί μία φορά στην αρχή της εκτέλεσης του προγράμματός σας για την Αλίκη και μία φορά στην αρχή της εκτέλεσης για τον Βασίλη. Της δίνεται το M και πρέπει να επιστρέψει το επιθυμητό N . Και οι δύο κλήσεις στο `setup` πρέπει να επιστρέψουν το ίδιο N .

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

Αυτή η συνάρτηση υλοποιεί τη στρατηγική της Αλίκης. Θα κληθεί με τον αριθμό A ($0 \leq A < N$) και πρέπει να επιστρέψει τρεις αριθμούς W_1, W_2, W_3 ($0 \leq W_i < M$) που κωδικοποιούν το A . Η συνάρτηση θα κληθεί συνολικά T φορές - μία φορά για κάθε ημέρα (οι τιμές του A μπορεί να επαναληφθούν μεταξύ των ημερών).



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

Αυτή η συνάρτηση υλοποιεί τη στρατηγική του Βασίλη. Θα κληθεί με δύο από τους τρεις αριθμούς που επιστρέφονται από το `encode` με κάποια σειρά. Πρέπει να επιστρέψει την ίδια τιμή A που έλαβε το `encode`. Θα κληθεί επίσης T φορές - που αντιστοιχούν στις T κλήσεις του `encode` και θα είναι με την ίδια σειρά. Όλες οι κλήσεις στο `encode` θα πραγματοποιηθούν πριν από όλες τις κλήσεις στο `decode`.



Περιορισμοί

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



Βαθμολόγηση

Για ένα συγκεκριμένο υποπρόβλημα, το κλάσμα S των μονάδων που λαμβάνετε εξαρτάται από το μικρότερο N που επιστρέφεται από το `setup` σε οποιαδήποτε περίπτωση ελέγχου (test case) του υποπροβλήματος. Εξαρτάται επίσης από το N^* , το οποίο είναι η τιμή του N που χρειάζεται για να λάβετε όλες τις μονάδες του υποπροβλήματος:

- Αν η λύση σας αποτύχει σε οποιαδήποτε περίπτωση ελέγχου, τότε $S = 0$.
- Αν $N \geq N^*$, τότε $S = 1.0$.
- Αν $N < N^*$, τότε $S = \max \left(0.35 \max \left(\frac{\log(N) - 0.985 \log(M)}{\log(N^*) - 0.985 \log(M)}, 0.0 \right)^{0.3} + 0.65 \left(\frac{N}{N^*} \right)^{2.4}, 0.01 \right)$.



Υποπροβλήματα

Υποπρόβλημα	Μονάδες	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

Παράδειγμα

Εξετάστε το ακόλουθο παράδειγμα με $T = 5$. Εδώ έχουμε μια στρατηγική κωδικοποίησης όπου η Αλίκη στέλνει τρεις ίσους αριθμούς για να κωδικοποιήσει τον αριθμό 0, ή τρεις διακριτούς αριθμούς για να κωδικοποιήσει 1. Παρατηρήστε ότι ο Βασίλης μπορεί να αποκωδικοποιήσει τον αρχικό αριθμό από οποιουδήποτε δύο από τους τρεις αριθμούς που έστειλε η Αλίκη.

Εκτέλεση	Κλήση συνάρτησης	Επιστρεφόμενη τιμή
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

Ενδεικτικός Βαθμολογητής (Sample grader)

Για τον ενδεικτικό βαθμολογητή, όλες οι κλήσεις στις συναρτήσεις `encode` και `decode` θα πραγματοποιούνται στην ίδια εκτέλεση του προγράμματός σας. Επιπλέον, η `setup` θα καλείται μόνο μία φορά (σε αντίθεση με δύο φορές, μία φορά ανά εκτέλεση, όπως στο σύστημα βαθμολόγησης).

Η είσοδος είναι ένας ακέραιος αριθμός $-M$. Στη συνέχεια, θα εκτυπώσει το N που επέστρεψε το `setup` που υλοποιήσατε. Στη συνέχεια, θα καλέσει τις συναρτήσεις `encode` και `decode`, με τυχαίους αριθμούς από 0 έως $N-1$, T φορές. Θα επιλέξει τυχαία δύο από τους τρεις αριθμούς που επιστρέφει το `encode` και θα τους δώσει στο `decode`. Θα εκτυπώσει ένα μήνυμα σφάλματος εάν η λύση σας αποτύχει.