

## Ζυγαριές

Η Αριάδνη έχει έξι κέρματα, αριθμημένα από **1** έως **6**. Γνωρίζει ότι κάθε κέρμα έχει διαφορετικό βάρος από τα υπόλοιπα. Θέλει να ταξινομήσει τα κέρματα ανάλογα με το βάρος τους. Για το σκοπό αυτό χρησιμοποιεί ένα νέο είδος ζυγαριάς.

Η παραδοσιακή ζυγαριά έχει δύο δίσκους. Για να τη χρησιμοποιήσει κανείς, βάζει ένα κέρμα σε κάθε δίσκο και η ζυγαριά προσδιορίζει ποιο είναι το βαρύτερο.

Η ζυγαριά της Αριάδνης είναι πιο πολύπλοκη. Έχει τέσσερις δίσκους, που ονομάζονται **A**, **B**, **C**, και **D**. Η ζυγαριά έχει τέσσερις ρυθμίσεις, κάθε μία από τις οποίες απαντά μια διαφορετική ερώτηση σχετικά με τα κέρματα. Για να χρησιμοποιήσει τη ζυγαριά, η Αριάδνη πρέπει να βάλει ακριβώς ένα κέρμα σε καθέναν από τους δίσκους **A**, **B**, και **C**. Επιπλέον, αν έχει επιλέξει την τέταρτη ρύθμιση, πρέπει επίσης να βάλει ακριβώς ένα κέρμα στο δίσκο **D**.

Οι τέσσερις ρυθμίσεις οδηγούν τη ζυγαριά να μπορεί να απαντά στις ακόλουθες τέσσερις ερωτήσεις:

1. Ποιο από τα κέρματα στους δίσκους **A**, **B**, και **C** είναι το βαρύτερο;
2. Ποιο από τα κέρματα στους δίσκους **A**, **B**, και **C** είναι το ελαφρύτερο;
3. Ποιο από τα κέρματα στους δίσκους **A**, **B**, και **C** είναι το διάμεσο; (Δηλαδή το κέρμα που δεν είναι ούτε το ελαφρύτερο ούτε το βαρύτερο από τα τρία.)
4. Ανάμεσα στα κέρματα των δίσκων **A**, **B**, και **C**, εξέτασε μόνο αυτά που είναι βαρύτερα από το κέρμα στο δίσκο **D**. Αν υπάρχουν τέτοια κέρματα, ποιο από αυτά είναι το ελαφρύτερο; Διαφορετικά, αν δεν υπάρχουν τέτοια κέρματα, ποιο από τα κέρματα στους δίσκους **A**, **B**, και **C** είναι το ελαφρύτερο;

## Πρόβλημα

Γράψτε ένα πρόγραμμα που να ταξινομεί τα έξι κέρματα της Αριάδνης, σύμφωνα με το βάρος τους. Το πρόγραμμα μπορεί να κάνει ερωτήσεις στη ζυγαριά της Αριάδνης, συγκρίνοντας τα βάρη των κερμάτων. Το πρόγραμμά σας θα δοκιμαστεί με πολλές περιπτώσεις ελέγχου, κάθε μία από τις οποίες θα αντιστοιχεί σε ένα νέο σύνολο έξι κερμάτων.

Το πρόγραμμά σας πρέπει να υλοποιεί τις συναρτήσεις `init` και `orderCoins`. Κατά τη διάρκεια κάθε εκτέλεσης του προγράμματος, το πρόγραμμα βαθμολόγησης (`grader`) θα καλεί πρώτα τη συνάρτηση `init` ακριβώς μία φορά. Αυτό σας δίνει το πλήθος των περιπτώσεων ελέγχου και σας επιτρέπει να αρχικοποιήσετε μεταβλητές, αν το επιθυμείτε. Το πρόγραμμα βαθμολόγησης στη συνέχεια θα καλεί την `orderCoins()` μία φορά για κάθε περίπτωση ελέγχου.

- `init(T)`
  - `T`: Το πλήθος των περιπτώσεων ελέγχου που το πρόγραμμά σας θα πρέπει να λύσει κατά τη διάρκεια αυτής της εκτέλεσης. Το `T` είναι ένας ακέραιος στο διάστημα `1, ..., 18`.

- Αυτή η συνάρτηση δεν έχει τιμή επιστροφής.
- `orderCoins()`
  - Αυτή η συνάρτηση καλείται ακριβώς μία φορά για κάθε περίπτωση ελέγχου.
  - Η συνάρτηση πρέπει να βρίσκει τη σωστή διάταξη των κερμάτων της Αριάδνης, καλώντας τις συναρτήσεις του προγράμματος βαθμολόγησης `getHeaviest()`, `getLightest()`, `getMedian()`, ή/και `getNextLightest()`.
  - Μόλις η συνάρτηση βρει τη σωστή διάταξη, πρέπει να την αναφέρει στο πρόγραμμα βαθμολόγησης καλώντας τη συνάρτηση `answer()`.
  - Μετά την κλήση της `answer()`, η συνάρτηση `orderCoins()` πρέπει να επιστρέψει. Δεν υπάρχει τιμή επιστροφής.

Μπορείτε να χρησιμοποιήσετε τις ακόλουθες συναρτήσεις του προγράμματος βαθμολόγησης στο πρόγραμμά σας:

- `answer(W)` — το πρόγραμμά σας πρέπει να χρησιμοποιεί αυτή τη συνάρτηση για να αναφέρει την απάντηση που βρήκε.
  - `W`: Ένας πίνακας μήκους 6 που περιέχει τη σωστή διάταξη των κερμάτων. Τα στοιχεία `W[0]` έως `W[5]` πρέπει να περιέχουν τους αριθμούς των κερμάτων (δηλαδή αριθμούς από το **1** έως το **6**) κατά σειρά από το ελαφρύτερο προς το βαρύτερο κέρμα.
  - Το πρόγραμμά σας πρέπει να καλεί αυτή τη συνάρτηση από την `orderCoins()`, μόνο μία φορά για κάθε περίπτωση ελέγχου.
  - Αυτή η συνάρτηση δεν έχει τιμή επιστροφής.
- `getHeaviest(A, B, C)`, `getLightest(A, B, C)`, `getMedian(A, B, C)` — αυτές αντιστοιχούν στις ρυθμίσεις 1, 2 και 3 αντίστοιχα της ζυγαριάς της Αριάδνης.
  - `A, B, C`: Τα κέρματα που τοποθετούνται στους δίσκους **A**, **B**, και **C**, αντίστοιχα. Τα `A`, `B`, και `C` πρέπει να είναι τρεις διαφορετικοί ακέραιοι αριθμοί, καθένας στο διάστημα από **1** έως **6** (συμπεριλαμβανομένων).
  - Κάθε συνάρτηση επιστρέφει έναν από τους αριθμούς `A`, `B`, και `C`: τον αριθμό του κατάλληλου κέρματος. Για παράδειγμα, η κλήση `getHeaviest(A, B, C)` επιστρέφει τον αριθμό του βαρύτερου από τα τρία δοθέντα κέρματα.
- `getNextLightest(A, B, C, D)` — αυτή αντιστοιχεί στην τέταρτη ρύθμιση της ζυγαριάς της Αριάδνης.
  - `A, B, C, D`: Τα κέρματα που τοποθετούνται στους δίσκους **A**, **B**, **C**, και **D**, αντίστοιχα. Τα `A`, `B`, `C`, και `D` πρέπει να είναι τέσσερις διαφορετικοί ακέραιοι αριθμοί, καθένας στο διάστημα από **1** έως **6** (συμπεριλαμβανομένων).
  - Η συνάρτηση επιστρέφει έναν από τους αριθμούς `A`, `B`, και `C`: τον αριθμό του κέρματος που επιλέγεται από τη ζυγαριά, σύμφωνα με τη διαδικασία που περιγράφηκε παραπάνω για τη ρύθμιση 4. Δηλαδή, το κέρμα που επιστρέφεται είναι το ελαφρύτερο μεταξύ των κερμάτων των δίσκων **A**, **B**, και **C** που είναι βαρύτερα από το κέρμα στο δίσκο **D**. Αν κανένα από αυτά δεν είναι βαρύτερο από το κέρμα στο δίσκο **D**, το κέρμα που επιστρέφεται είναι απλά το ελαφρύτερο από τα τρία κέρματα στους δίσκους **A**, **B**, και **C**.

## Βαθμολογία

Δεν υπάρχουν υποπροβλήματα σε αυτό το πρόβλημα. Αντίθετα, η βαθμολογία σας θα βασιστεί στο πλήθος των ζυγισμάτων (συνολικό πλήθος των κλήσεων στις συναρτήσεις `getLightest()`, `getHeaviest()`, `getMedian()` ή/και `getNextLightest()`) που το πρόγραμμά σας θα κάνει.

Το πρόγραμμά σας θα εκτελεστεί πολλές φορές με πολλές περιπτώσεις ελέγχου κάθε φορά. Έστω  $r$  το πλήθος των εκτελέσεων. Ο αριθμός αυτός είναι καθορισμένος στα δεδομένα ελέγχου. Αν το πρόγραμμά σας δεν ταξινομεί σωστά τα κέρματα σε οποιαδήποτε από τις εκτελέσεις, θα πάρετε 0 βαθμούς. Διαφορετικά, οι εκτελέσεις ταξινομούνται ανεξάρτητα η μία από την άλλη, ως εξής.

Έστω  $Q$  ο ελάχιστος αριθμός τέτοιος ώστε να είναι δυνατό να ταξινομηθεί κάθε σύνολο έξι κερμάτων χρησιμοποιώντας  $Q$  ζυγίσματα με τη ζυγαριά της Αριάδνης. Για να κάνουμε το πρόβλημα πιο δύσκολο, δεν αποκαλύπτουμε εδώ την τιμή του  $Q$ .

Υποθέστε ότι το μέγιστο πλήθος ζυγισμάτων ανάμεσα σε όλες τις περιπτώσεις ελέγχου όλων των εκτελέσεων είναι  $Q + y$  για κάποιον ακέραιο  $y$ . Στη συνέχεια, έστω μία συγκεκριμένη εκτέλεση του προγράμματός σας. Έστω ότι το μέγιστο πλήθος ζυγισμάτων μεταξύ όλων των  $T$  περιπτώσεων ελέγχου αυτής της εκτέλεσης είναι  $Q + x$  για κάποιο μη αρνητικό ακέραιο  $x$ . (Αν χρησιμοποιείτε λιγότερα από  $Q$  ζυγίσματα για κάθε περίπτωση ελέγχου, τότε  $x = 0$ .) Τότε, η βαθμολογία σας για αυτή την εκτέλεση θα είναι  $\frac{100}{r((x+y)/5+1)}$ , στρογγυλευμένη προς τα κάτω με δύο δεκαδικά ψηφία.

Ειδικότερα, αν το πρόγραμμά σας κάνει το πολύ  $Q$  ζυγίσματα σε κάθε περίπτωση ελέγχου κάθε εκτέλεσης, θα πάρετε 100 βαθμούς.

## Παράδειγμα

Υποθέστε ότι τα κέρματα είναι ταξινομημένα **3 4 6 2 1 5** από το ελαφρύτερο προς το βαρύτερο.

Κλήση	Επιστρέφει	Εξήγηση
<code>getMedian(4, 5, 6)</code>	6	Το κέρμα 6 είναι το διάμεσο των 4, 5, και 6.
<code>getHeaviest(3, 1, 2)</code>	1	Το κέρμα 1 είναι το βαρύτερο των 1, 2, και 3.
<code>getNextLightest(2, 3, 4, 5)</code>	3	Τα κέρματα 2, 3, και 4 είναι όλα ελαφρύτερα από το κέρμα 5, επομένως επιστρέφεται το ελαφρύτερο από αυτά (3).
<code>getNextLightest(1, 6, 3, 4)</code>	6	Τα κέρματα 1 και 6 είναι και τα δύο βαρύτερα από το κέρμα 4. Μεταξύ των κερμάτων 1 και 6, το κέρμα 6 είναι το ελαφρύτερο.
<code>getHeaviest(3, 5, 6)</code>	5	Το κέρμα 5 είναι το βαρύτερο των coins 3, 5 και 6.
<code>getMedian(1, 5, 6)</code>	1	Το κέρμα 1 είναι το διάμεσο των 1, 5 και 6.
<code>getMedian(2, 4, 6)</code>	6	Το κέρμα 6 είναι το διάμεσο των 2, 4 και 6.
<code>answer([3, 4, 6, 2, 1, 5])</code>		Το πρόγραμμα βρήκε τη σωστή διάταξη για αυτή την περίπτωση ελέγχου.

## Υπόδειγμα βαθμολογητή (Sample grader)

Ο βαθμολογητής που σας δίνεται ως υπόδειγμα διαβάζει την είσοδό του με την εξής μορφή:

- γραμμή **1**:  $T$  — το πλήθος των περιπτώσεων ελέγχου
- κάθε μία από τις γραμμές **2** έως  $T + 1$ : μία ακολουθία **6** διακριτών αριθμών από το **1** έως το **6** : η σειρά των κερμάτων από το ελαφρύτερο προς το βαρύτερο.

Για παράδειγμα, η είσοδος που αποτελείται από δύο περιπτώσεις ελέγχου, όπου τα κέρματα είναι διατεταγμένα **1 2 3 4 5 6** και **3 4 6 2 1 5** είναι η εξής:

```

2
1 2 3 4 5 6
3 4 6 2 1 5

```

Το υπόδειγμα βαθμολογητή τυπώνει τον πίνακα που περνά ως παράμετρος στη συνάρτηση `answer()`.