

Digger

Αυτό είναι ένα διαδραστικό (interactive) πρόβλημα

Περιγραφή προβλήματος

Κατά τις ανασκαφές στα ερείπια της αρχαίας πόλης της Σαλαμίνας, οι αρχαιολόγοι ανακάλυψαν έναν κρυμμένο βασιλικό τάφο, θαμμένο σε άγνωστο βάθος n μέτρων κάτω από την επιφάνεια. Η αποστολή σας είναι να βρείτε τον αριθμό n εκτελώντας πειράματα με ένα ρομπότ εκσκαφέα.

Σε κάθε πείραμα, δίνετε στο ρομπότ ένα σύνολο ακεραίων a_1, a_2, \dots, a_k και του ζητάτε να σκάψει από την επιφάνεια (βάθος 0), στο βάθος n , σκάβοντας μόνο σε κινήσεις των a_i μέτρων τη φορά. Αυτό σημαίνει ότι αν το ρομπότ βρίσκεται στο βάθος x , μπορεί να προχωρήσει μόνο σε κάποιο από τα βάθη $x + a_1, x + a_2, \dots$, or $x + a_k$. Όταν το ρομπότ φτάσει στο βάθος n , σας αναφέρει τον αριθμό των κινήσεων που έκανε. Το ρομπότ είναι αρκετά έξυπνο και πάντα επιλέγει τον μικρότερο αριθμό κινήσεων για να φτάσει στο βάθος n .

Για παράδειγμα, έστω ότι το βάθος που ψάχνουμε είναι $n = 23$. Αν δώσετε στο ρομπότ το σύνολο $a = [1, 3, 8]$, θα φτάσει στο βάθος n σε 5 κινήσεις (π.χ. $0 \rightarrow 3 \rightarrow 4 \rightarrow 7 \rightarrow 15 \rightarrow 23$) και αν του δώσετε το σύνολο $a = [9, 5]$, θα φτάσει το βάθος n σε 3 κινήσεις ($0 \rightarrow 9 \rightarrow 18 \rightarrow 23$).

Η δουλειά σας είναι να ανακαλύψετε το βάθος n αφού κάνετε μερικά πειράματα με τους παρακάτω περιορισμούς:

- Το βάθος n βρίσκεται στο διάστημα 1 έως 30000 (συμπεριλαμβανομένων).
- Ο συνολικός αριθμός των πειραμάτων είναι το πολύ 20.
- Το συνολικό πλήθος στοιχείων των συνόλων a για όλα τα πειράματα πρέπει να είναι το πολύ 250.
- Σε κάθε πείραμα, πρέπει να είναι εφικτό το ρομπότ να φτάσει στο βάθος n .

Αλληλεπίδραση (Interaction)

Για να κάνετε ένα πείραμα, το πρόγραμμά σας πρέπει να τυπώσει μια γραμμή της μορφής: $? \quad k \quad a_1 \quad a_2 \quad \dots \quad a_k$, όπου k ($1 \leq k \leq 250$) είναι το πλήθος στοιχείων του συνόλου και τα a_i ($1 \leq a_i \leq 30000$) είναι τα στοιχεία του συνόλου. Κατόπιν, το πρόγραμμα σας πρέπει να διαβάσει

έναν μοναδικό ακέραιο m , τον ελάχιστο αριθμό των κινήσεων που χρειάστηκε το ρομπότ για να φτάσει στο βάθος n .

Όταν το πρόγραμμά σας έχει ανακαλύψει το βάθος n , θα πρέπει να τυπώσει τη γραμμή: `! n`. Μετά από αυτό, το πρόγραμμά σας πρέπει να τελειώσει.

Μετά από την εκτύπωση κάθε πειράματος, μην ξεχάσετε να τυπώσετε μια αλλαγή γραμμής και να κάνετε flush την έξοδο.

Αν σε οποιοδήποτε πείραμα, πάρετε για απάντηση `-1` αντί για έγκυρη απάντηση, το πρόγραμμά σας πρέπει να τερματίσει άμεσα. Αυτό σημαίνει ότι το πρόγραμμά σας θα λάβει `Wrong answer` λόγω μη έγκυρων δεδομένων πειράματος ή κάποιο άλλο σφάλμα. Αν δεν τερματιστεί το πρόγραμμά σας μπορεί να λάβει αυθαίρετη αξιολόγηση (verdict) καθώς θα συνεχίσει να διαβάζει από μια κλειστή ροή εισόδου.

Για να κάνετε flush την έξοδο, χρησιμοποιήστε:

- `fflush(stdout)` ή `cout.flush()` στη C++;
- `sys.stdout.flush()` στην Python;

Παράδειγμα

| Είσοδος | Έξοδος |
|---------|-----------|
| | ? 3 1 3 8 |
| 5 | |
| | ? 2 9 5 |
| 3 | |
| | ! 23 |

Βαθμολόγηση

Το πρόγραμμά σας θα εξεταστεί με 100 δοκιμές. Αν αποτύχει να ανακαλύψει το βάθος n , ή ξεπεράσει τους περιορισμούς σε τουλάχιστον μια δοκιμή, θα πάρει 0 βαθμούς.

Αν το πρόγραμμά σας ανακαλύψει με επιτυχία τους αριθμούς n , και δεν υπερβεί τους περιορισμούς σε όλες τις δοκιμές, η βαθμολογία σας θα υπολογιστεί με βάση τον συνολικό αριθμό των κινήσεων που έκανε το ρομπότ σε όλα τα πειράματα.

Έστω s_i ο συνολικός αριθμός των κινήσεων που έκανε το ρομπότ σε όλα τα πειράματα στη δοκιμή i και έστω S ο μέγιστος αριθμός από όλα τα s_i για όλες τις 100 δοκιμές. Η βαθμολογία σας θα υπολογιστεί με βάση τον παρακάτω πίνακα:

| S | Βαθμολογία |
|---------------|------------|
| ≤ 100 | 100 |
| 101 – 150 | 95 |
| 151 – 200 | 90 |
| 201 – 250 | 80 |
| 251 – 300 | 70 |
| 301 – 500 | 60 |
| 501 – 1000 | 50 |
| 1001 – 5000 | 20 |
| 5001 – 20000 | 10 |
| 20001 – 30000 | 5 |
| ≥ 30001 | 0 |