

# Digger

*Aceasta este o problemă interactivă*

## Task

În timpul săpăturilor arheologice efectuate la ruinele vechiului oraș-regat Salamis, arheologii au descoperit un mormânt regal ascuns, îngropat la o adâncime necunoscută de  $n$  metri sub suprafață. Sarcina ta este să afli numărul  $n$  efectuând experimente cu robotul tău de săpat.

În fiecare experiment, îi dai robotului o listă de numere întregi  $a_1, a_2, \dots, a_k$  și îi ceri să sape de la suprafață (adâncime 0) până la adâncimea  $n$ , folosind la fiecare moment doar pași de lungime  $a_i$ . Asta înseamnă că, dacă adâncimea curentă a robotului este  $x$ , acesta se poate deplasa la adâncimile  $x + a_1, x + a_2, \dots$ , sau  $x + a_k$ . Când robotul ajunge la adâncimea  $n$ , îți spune numărul de pași pe care i-a făcut. Robotul este foarte inteligent și face întotdeauna numărul minim posibil de pași pentru a ajunge la adâncimea  $n$ .

De exemplu, să presupunem că numărul ascuns este  $n = 23$ . Dacă îi dai robotului o listă  $a = [1, 3, 8]$ , acesta poate ajunge la adâncimea  $n$  în 5 pași (de exemplu,  $0 \rightarrow 3 \rightarrow 4 \rightarrow 7 \rightarrow 15 \rightarrow 23$ ), iar dacă îi dai robotului o listă  $a = [9, 5]$ , acesta poate ajunge la adâncimea  $n$  în 3 pași ( $0 \rightarrow 9 \rightarrow 18 \rightarrow 23$ ).

Sarcina ta este să ghicești numărul  $n$  după ce faci mai multe experimente cu următoarele restricții:

- Numărul ascuns  $n$  este cuprins între 1 și 30000 (inclusiv).
- Numărul total de experimente trebuie să fie de cel mult 20.
- Dimensiunea totală a mulțimilor  $a$  în toate experimentele trebuie să fie de cel mult 250.
- În fiecare experiment, robotul trebuie să poată atinge adâncimea  $n$ .

## Interaction

Pentru a face un experiment, programul tău trebuie să afișeze o linie de forma:  $? \ k \ a_1 \ a_2 \ \dots \ a_k$ , unde  $k$  ( $1 \leq k \leq 250$ ) este dimensiunea listei și  $a_i$  ( $1 \leq a_i \leq 30000$ ) sunt elementele listei. După aceea, programul tău trebuie să citească un singur număr întreg  $m$ , numărul minim de pași pe care robotul a trebuit să îi facă pentru a ajunge la adâncimea  $n$ .

Când programul tău este gata să ghicească numărul  $n$ , trebuie să afișeze următoarea linie: `! n`. După aceea, programul tău trebuie să se închidă.

După afișarea fiecărei interogări, nu uitați să afișați sfârșitul liniei și să dați `flush` la ieșire.

Dacă, la orice pas de interacțiune, citiți `-1` în loc de date valide, soluția trebuie să se închidă imediat. Acest lucru înseamnă că soluția ta va fi considerată incorectă din cauza unui experiment invalid sau a oricărei alte greșeli. Neînchiderea programului poate duce la un verdict arbitrar, deoarece soluția va continua să citească dintr-un flux închis.

Pentru a da `flush` la ieșire, utilizați:

- `fflush(stdout)` or `cout.flush()` în C++;
- `sys.stdout.flush()` în Python;

---

## Example

| Input | Output    |
|-------|-----------|
|       | ? 3 1 3 8 |
| 5     |           |
|       | ? 2 9 5   |
| 3     |           |
|       | ! 23      |

---

## Scoring

Soluția va fi testată pe 100 de teste. Dacă soluția nu reușește să ghicească numărul  $n$  sau depășește limitele date în cel puțin un test, va primi 0 puncte.

Dacă soluția ghicește cu succes numărul  $n$  și nu depășește limitele în toate testele, scorul va fi calculat pe baza numărului total de pași pe care robotul i-a făcut în toate experimentele.

Fie  $s_i$  numărul total de pași pe care robotul i-a făcut în toate experimentele din testul  $i$ , și fie  $S$  maximul lui  $s_i$  pentru toate cele 100 de teste. Atunci scorul dvs. va fi calculat pe baza următorului tabel:

| $S$           | Score |
|---------------|-------|
| $\leq 100$    | 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     |
| $\geq 30001$  | 0     |