

# Kopač

*Ta naloga je interaktivna*

## Naloga

Med izkopavanji ruševin antičnega mestnega kraljestva Salamina arheologi naletijo na skrito kraljevo grobnico, zakopano na neznani globini  $n$  metrov pod površjem. Z izvajanjem eksperimentov z robotom za kopanje ugotovite število  $n$ .

V vsakem eksperimentu robotu podate seznam celih števil  $a_1, a_2, \dots, a_k$  in ga prosite, naj koplje od površine (globina 0) do globine  $n$ , pri čemer lahko uporabi le korake dolžine  $a_i$ . To pomeni, če je trenutna globina robota  $x$ , se lahko premakne na globino  $x + a_1$ ,  $x + a_2$ , ..., ali  $x + a_k$ . Ko robot doseže globino  $n$ , vam pove, koliko korakov je naredil. Robot je zelo inteligenčen in vedno naredi minimalno možno število korakov, da doseže globino  $n$ .

Na primer, naj bo iskano število  $n = 23$ . Če robotu podate seznam  $a = [1, 3, 8]$ , doseže globino  $n$  v 5 korakih (na primer  $0 \rightarrow 3 \rightarrow 4 \rightarrow 7 \rightarrow 15 \rightarrow 23$ ), in če podate robotu seznam  $a = [9, 5]$ , doseže globino  $n$  v 3 korakih ( $0 \rightarrow 9 \rightarrow 18 \rightarrow 23$ ).

Vaša naloga je uganiti število  $n$  po izvedbi več eksperimentov z naslednjimi omejitvami:

- Iskano število  $n$  je v razponu od 1 do 30000 (vključno).
- Skupno število eksperimentov mora biti največ 20.
- Skupna dolžina vseh seznamov  $a$  vseh eksperimentov skupaj mora biti največ 250.
- V vsakem eksperimentu mora biti robotu s podanim seznamom  $a$  mogoče doseči globino  $n$ .

## Interakcija

Za izvedbo eksperimenta vaša rešitev izpiše eno vrstico oblike:  $? \ k \ a_1 \ a_2 \ \dots \ a_k$ , kjer  $k$  ( $1 \leq k \leq 250$ ) predstavlja dolžino seznama in  $a_i$  ( $1 \leq a_i \leq 30000$ ) elemente seznama. Nato vaša rešitev prebere celo število  $m$ , minimalno število korakov, ki jih robot naredi, da doseže globino  $n$ .

Ko vaša rešitev ugotovi število  $n$ , izpiše vrstico oblike  $! \ n$ . Po tem mora vaša rešitev končati izvajanje.

Po vsakem izpisu eksperimenta ne pozabite izpisati tudi znaka za novo vrstico in počistiti izhodnega medpomnilnika.

Za čiščenje uporabite:

- `fflush(stdout)` ali `cout.flush()` v C++;
- `sys.stdout.flush()` v Pythonu;

Če na katerem koli koraku interakcije namesto veljavnih podatkov dobite `-1`, mora vaša rešitev takoj končati izvajanje. To pomeni, da je vaša rešitev zavrnjena zaradi neveljavnega eksperimenta ali katere koli druge napake. Neuspešno končanje lahko vrne katero koli napako, ker bo vaša rešitev še naprej brala iz zaprtega toka.

---

## Primer

Vhod	Izhod
	? 3 1 3 8
5	
	? 2 9 5
3	
	! 23

---

## Točkovanje

Vaša rešitev bo testirana na 100 testih. Če vaša rešitev ne uspe uganiti števila  $n$ , ali preseže dane omejitve vsaj na enem testu, bo prejela 0 točk.

Če vaša rešitev uspešno ugame število  $n$ , in ne preseže omejitev pri vseh testih, se točkuje na naslednji način:

naj bo  $s_i$  skupno število korakov, ki jih je robot opravil v vseh eksperimentih testa  $i$ , in naj bo  $S$  maksimum  $s_i$  za vseh 100 testov. Potem se vaš rezultat izračuna na podlagi naslednje tabele:

$S$	Točke
$\leq 100$	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