

# Where's Waldo?

Problem Name	whereswaldo
Time Limit	11 seconds
Memory Limit	1 gigabyte

Există o permutare ascunsă  $P_0, P_1, \dots, P_{N-1}$  de lungime  $N$ , care este garantat generată uniform la întâmplare. Permutarea conține numerele  $1, 2, 3, \dots, N$  exact o dată fiecare, într-o ordine necunoscută.

Se pot alege pozițiile  $l$  și  $r$ , și se pot pune întrebări de forma: "Care este suma elementelor  $P_l + P_{l+1} + \dots + P_r$ ?"

Sarcina dvs este să găsiți poziția lui 1 în  $P$  folosind cât mai puține întrebări. Veți fi punctat în funcție de numărul de întrebări folosite.

## Interacțiunea

Programul dvs. ar trebui să citească mai întâi două numere întregi de pe o singură linie,  $T$  și  $N$ .  $T$  este numărul de runde pe care va fi testat programul dumneavoastră, iar  $N$  este lungimea lui  $P$ .

După sunt prezentate  $T$  runde:

Când începe o rundă, puteți începe să puneți întrebări. Tipăriți o linie cu „? a b” pentru a întreba despre suma numerelor dintre pozițiile  $a$  și  $b$  inclusiv ( $0 \leq a \leq b \leq N - 1$ ).

După fiecare întrebare, programul dvs. ar trebui să citească un număr întreg, suma numerelor din interval.

Odată ce ați găsit poziția lui 1, imprimați o linie de forma “! i”, unde  $i$  este indexul astfel încât  $P_i = 1$ . După ce imprimați indexul, va începe următoarea rundă.

Asigurați-vă că ștergeți ieșirea standard după ce ați pus o întrebare, în caz contrar programul dvs. ar putea fi considerat că a depășit limita de timp.

În Python, `print()` șterge automat. În C++, `cout << endl;` de asemenea șterge, în plus adaugă o nouă linie; dacă utilizați `printf`, utilizați `fflush(stdout)`.

## Restricții și Punctaj

Programul dvs. va fi testat pe **un singur caz de testare cu  $N = T = 1000$** . Permutarea în fiecare test este garantată a fi **generată la întâmplare**.

Dacă soluția dvs. ghicește greșit în oricare dintre runde, soluția dvs. va fi evaluată ca *Răspuns greșit*.

În caz contrar, punctajul va fi calculat după cum urmează:

$$\text{score} = \min \left( 220 - \frac{M}{2500}, 100 \right) \text{ puncte,}$$

unde  $M$  este numărul de întrebări pe care programul dvs. le adresează în total în toate  $T$  runde.

Punctajul va fi rotunjit la cel mai apropiat număr întreg. Dacă punctajul devine negativ, acesta va fi tratat ca zero puncte.

Astfel, dacă utilizați mai mult de 550 000 întrebări, veți primi 0 puncte, iar dacă utilizați 300 000 sau mai puține întrebări, veți primi 100 puncte. Între timp, punctajul dvs. crește liniar.

## Instrumentul de testare

Pentru a facilita testarea soluției dumneavoastră, vă punem la dispoziție un instrument simplu pe care îl puteți descărca.

Accesați "attachments" în partea de jos a paginii cu probleme kattis.

Utilizarea instrumentului este opțională, iar dumneavoastră aveți dreptul de a-l modifica. Rețineți că programul grader oficial de pe kattis este diferit de instrumentul de testare.

Exemplu de utilizare (cu  $T=1000$ ,  $N=10$ ):

Pentru programe python fie `solution.py` (normal se rulează cu `pypy3 solution.py`):

```
python3 testing_tool.py pypy3 solution.py <<<"1000 10"
```

Pentru programe C++, mai întâi compilați-l (de exemplu cu `g++ -std=gnu++17 solution.cpp -o solution.out`) și apoi rulați:

```
python3 testing_tool.py ./solution.out <<<"1000 10"
```

## Exemplu

Fie următoarele date  $T = 2$  și  $N = 10$ . Pentru prima dintre aceste două runde, fie că permutarea ascunsă este "6 10 8 7 9 1 2 4 5 3". Prima întrebare ? 0 9 cere suma tuturor numerelor,

care este într-adevăr 55, iar la a doua întrebare ? 0 4 cere  $6 + 10 + 8 + 7 + 9 = 40$ .

grader output	your output
2 10	
	? 0 9
55	
	? 0 4
40	
	? 5 5
1	
	! 5
	? 0 0
1	
	! 0