

## Where's Waldo?

Numele problemei	whereswaldo
Limita de timp	11 secunde
Limita de memorie	1 gigabyte

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

Poți să alegi două poziții,  $l$  și  $r$ , și să pui întrebări de forma: "Care este suma  $P_l + P_{l+1} + \dots + P_r$ ?"

Task-ul tău este să găsești poziția elementului 1 în  $P$ , utilizând cat mai puține întrebări posibile. Scorul problemei va fi bazat pe numărul de întrebări folosite.

### Interaction

Programul tău citește inițial două numere întregi pe o singura linie,  $T$  și  $N$ .  $T$  este numărul de runde în care va fi testat programul, iar  $N$  reprezintă lungimea permutării  $P$ .

După acestea, urmează  $T$  runde:

Când începe o rundă, poți începe să pui întrebări. Afișează o linie " $? a b$ " pentru a întreba care este suma numerelor dintre pozițiile  $a$  și  $b$ , inclusiv ( $0 \leq a \leq b \leq N - 1$ ).

După fiecare întrebare, programul citește un număr întreg, reprezentând suma numerelor din interval.

Odată ce ai găsit poziția elementului 1, afișează o linie de forma " $! i$ ", unde  $i$  este indexul pentru care  $P_i = 1$ . După aceasta, începe următoarea rundă.

Asigură-te că dai flush la output după ce pui o întrebare, altfel, programul poate lua Time Limit Exceeded. În Python, `print()` dă flush automat. În C++, `cout << endl;` dă flush, pe langa afișatul unei linii noi. Pentru `printf`, se folosește `fflush(stdout)`.

### Constraints and Scoring

Programul va fi rulat o singură dată, **pe un singur test, cu  $N = T = 1000$** . Permutarea pentru fiecare rundă se garantează că este generată random.

Dacă programul răspunde greșit în cel puțin o rundă, soluția va primi verdictul *Wrong Answer*.

Altfel, scorul va fi calculat astfel:

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

unde  $M$  este numărul de întrebări puse în total de program în toate cele  $T$  runde.

Scorul va fi aproximat la cel mai apropiat număr întreg. Dacă scorul devine negativ, va fi considerat drept 0.

Prin urmare, dacă sunt puse mai mult de 550 000 de întrebări, vei primi 0 puncte, iar dacă sunt puse 300 000 sau mai puține întrebări, vei primi 100 de puncte. Scorul crește liniar când numărul de întrebări este între 300 000 și 550 000.

## Testing Tool

Pentru a ușura testarea soluției, vă punem la dispoziție o arhivă pe care o puteți descarca. Vezi atașamentele din josul paginii problemei de pe kattis. Folosirea arhivei este opțională, și aveți posibilitatea să o modificați. Grader-ul oficial de pe kattis este diferit față de cel din arhiva.

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

Pentru programele python, `solution.py` (normal rulată cu `pypy3 solution.py`):

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

Pentru programele C++, mai întâi se compilează (exemplu: `g++ -std=gnu++17 solution.cpp -o solution.out`), apoi se rulează:

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

## Example

În exemplu,  $T = 2$  și  $N = 10$ . Pentru prima rundă, să spunem că permutarea ascunsă este "6 10 8 7 9 1 2 4 5 3". Prima întrebare ? 0 9 cere suma tuturor numerelor, care este 55, iar a doua întrebare ? 0 4 cere suma  $6 + 10 + 8 + 7 + 9 = 40$ .

outputul graderului	outputul tau
2 10	
	? 0 9
55	
	? 0 4
40	
	? 5 5
1	
	! 5
	? 0 0
1	
	! 0