

Kde je Waldo?

Problem Name	whereswaldo
Time Limit	11 seconds
Memory Limit	1 gigabyte

Testovač si rovnomerne náhodne vygeneroval nejakú permutáciu čísel od 1 po N . Jej prvky si označíme P_0, P_1, \dots, P_{N-1} .

Tvojim cieľom je zistiť index, na ktorom sa v P sa nachádza číslo 1. Tvoje skóre bude závisieť od počtu otázok, ktoré na to budeš potrebovať.

V každej otázke si môžeš zvoliť dve pozície a a b . Tým sa testovača opýtaš otázku "Aký je súčet $P_a + P_{a+1} + \dots + P_b$?" a on ti na ňu pravdivo odpovie.

Interaction

Tvoj program by mal začať tým, že načíta jeden riadok, v ktorom sú medzerou oddelené celé čísla T a N . T je počet kôl testovania, ktoré budú nasledovať, a N je dĺžka permutácie P v každom z nich.

Následne prebehne T nezávislých kôl - každé s nanovo vygenerovanou permutáciou danej dĺžky.

V každom kole sa tvoj program môže pýtať otázky. Otázku položí tak, že vypíše riadok tvaru " $? \ a \ b$ " ($0 \leq a \leq b \leq N - 1$) a následne načíta riadok s jedným celým číslom: súčtom prvkov od P_a po P_b vrátane.

Akonáhle tvoj program zistí index i taký, že $P_i = 1$, oznámi to vypísaním riadku tvaru " $! \ i$ ". Výpis takéhoto riadku ukončí aktuálne kolo (a ak sa ešte kolá neminuli, rovno sa začne nové).

Aby ste nedostali Time Limit Exceeded, je potrebné po každom vypísanom riadku flushnúť štandardný výstup. V Pythone to `print()` spraví automaticky. V C++ `cout << endl;` tiež okrem vypísania znaku konca riadku aj flushne výstup. Ak používate `printf`, použite po ňom príkaz `fflush(stdout)`.

Constraints and Scoring

Tvoj program bude otestovaný **na jedinom testovacom vstupe**. V ňom platí $T = N = 1000$. Každá z T použitých permutácií **bola náhodne vygenerovaná**.

Ak tvoje riešenie v niektorom kole odovzdá zlú odpoveď, dostaneš verdikt *Wrong Answer*.

Ak všetky testy vyriešiš správne, tvoje skóre bude vypočítané nasledovne:

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

kde M je celkový počet otázok položených твоjím programom (dokopy za všetkých T kôl).

Skóre bude zaokrúhľené na najbližšie celé číslo. Záporné skóre bude zmenené na nulu.

Slovne: ak použiješ viac ako 550 000 otázok, dostaneš nulu. Ak použiješ 300 000 otázok alebo menej, dostaneš plných 100 bodov. Medzi týmito dvoma hranicami skóre lineárne rastie.

Testing Tool

Aby sa ti tvoje riešenie ľahšie testovalo na tvojom počítači, môžeš si stiahnuť jednoduchý pomocný nástroj. (V Kattise na stránke tejto úlohy ho nájdeš dole v sekcii "Attachments".)

Používanie tohto nástroja je dobrovoľné. Je povolené ľubovoľne si ho upraviť. (Upozorňujeme, že pri oficiálnom testovaní odovzdaných riešení sa používa iná verzia tohto nástroja.)

Príklad použitia (s hodnotami $T=1000$ a $N=10$):

V Pythone: Nech je tvoje riešenie v súbore `solution.py` (ktorý by si normálne spúšťala príkazom `pypy3 solution.py`). Spusti:

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

Program v C++ treba najskôr štandardne skompilovať (napr. príkazom `g++ -std=gnu++17 solution.cpp -o solution.out`) a potom ho spustiť nasledovne:

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

Example

V nižšie uvedenej ukážke interakcie máme $T = 2$ a $N = 10$.

V prvom kole je skrytá permutácia "6 10 8 7 9 1 2 4 5 3". Prvá otázka ? 0 9 sa pýta na súčet všetkých jej prvkov. Druhá otázka ? 0 4 má odpoveď $6 + 10 + 8 + 7 + 9 = 40$.

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