

## Missä on Waldo?

Tehtävän nimi	whereswaldo
Aikaraja	11 sekuntia
Muistiraja	1 gigatavu

Olkoon  $P$  satunnaisesti generoitu  $N$ :n pituinen tuntematon permutaatio  $P_0, P_1, \dots, P_{N-1}$ . Permutaatio koostuu luvuista  $1, 2, 3, \dots, N$ , ja jokainen luku esiintyy permutaatiossa tasan kerran.

Valitsemalla jotkin indeksit  $l$  ja  $r$ , voidaan laskea summa  $P_l + P_{l+1} + \dots + P_r$ .

Tee ohjelma, joka etsii luvun 1 sijainnin  $P$ :stä mahdollisimman harvalla kysymyksellä. Ohjelma arvostellaan kysytyjen kysymysten määrän perusteella.

## Interaktio

Ensin ohjelmasi tulee lukea kaksi kokonaislukua,  $T$  ja  $N$ , yhdeltä riviltä.  $T$  on ohjelmasi testaukseen käytettävien kierrosten määrä ja  $N$  permutaation  $P$  pituus.

Tämän jälkeen tulee  $T$  kierrosta:

Toiston alkaessa saat alkaa esittää kysymyksiä. Kysyäksesi summaa välillä  $a$ - $b$ , kun ( $0 \leq a \leq b \leq N - 1$ ), tulosta rivi "? a b".

Jokaisen kysymyksen jälkeen ohjelmasi tulee lukea välin summa kokonaislukuna.

Kun löydät luvun 1 sijainnin permutaatiossa, tulosta "! i", jossa  $i$  on indeksi, jolla  $P_i = 1$ . Tämän tulostamisen jälkeen seuraava kierros alkaa.

Varmista, että pus kuri tyhjennetään jokaisen kysymyksen jälkeen, ettei ohjelmasi tulkita ylittävän aikarajaa (*Time Limit Exceeded*). Pythonissa `print()` tyhjentää puskurin automaattisesti. C++:ssa `cout << endl;` tyhjentää puskurin tulostamisen lisäksi. Jos käytät `printf`:ää, käytä lisäksi komentoa `fflush(stdout)`.

## Rajat ja pisteytys

Ohjelmasi testaukseen käytetään **joka kerta samoja arvoja**  $N = T = 1000$ . Jokaisella testikerralla permutaatio luodaan satunnaisesti.

Jos ohjelmasi arvaa väärin kertaakaan, palautuksesi tulkitaan virheelliseksi (*Wrong Answer*).

Muutoin palautus pisteytetään seuraavasti:

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

jossa  $M$  on kaikkien kysytyjen kysymysten määrä kaikilla  $T$ :llä kierroksella.

Pisteet pyöristetään lähimpään kokonaislukuun. Negatiiviset pisteet tulkitaan nollana pisteenä.

Toisin sanoen jos ohjelmasi kysyy yli 550 000 kysymystä, saat 0 pistettä. Jos ohjelma kysyy 300 000 kysymystä tai vähemmän, saat 100 pistettä. Kysymysten määrä muuttaa pisteitä lineaarisesti.

## Testaustyökalut

Ratkaisusi testaamiseksi on käytössä yksinkertainen ohjelma, joka on ladattavissa Kattiksen kautta. Ohjelma löytyy Kattiksen tehtäväsivun alalaidasta liitteistä kohdasta "attachments". Testausohjelmaa ei ole pakko käyttää ja sitä saa muuttaa. Muista että kilpailun virallinen testausohjelma on erilainen.

Esimerkki käytöstä, kun  $T = 1000$  ja  $N = 10$ :

Python-ohjelmille kirjoita `solution.py` (tavallisesti suoritettuna `pypy3 solution.py`):

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

Jos käytät C++:aa, käännä ensin ohjelma (esim. `g++ -std=gnu++17 solution.cpp -o solution.out`) ja sitten suorita:

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

## Esimerkki

Olkoon  $T = 2$  ja  $N = 10$ . Olkoon permutaatio ensimmäisellä kierroksella "6 10 8 7 9 1 2 4 5 3". Ensimmäinen kysymys ? 0 9 kysyy kaikkien lukujen summaa, joka on 55, ja seuraava kysymys ? 0 4 kysyy summaa  $6 + 10 + 8 + 7 + 9 = 40$ .

arvostelijan tuloste	ohjelmasi tuloste
2 10	
	? 0 9
55	
	? 0 4
40	
	? 5 5
1	
	! 5
	? 0 0
1	
	! 0