

## Where's Waldo?

Feladatnév	whereswaldo
Időkorlát	11 másodperc
Memóriakorlát	1 gigabyte

Adott az  $N$  hosszúságú titkos permutáció  $P_0, P_1, \dots, P_{N-1}$ , amely garantáltan egyenletesen véletlenszerűen generálódik. A permutáció az  $1, 2, 3, \dots, N$  számok mindegyikét pontosan egyszer tartalmazza, valamilyen ismeretlen sorrendben.

Kiválasztva az  $l$  és  $r$  indexeket felteheted a következő kérdést: "Mennyi a  $P_l + P_{l+1} + \dots + P_r$  összeg?"

A feladatod az, hogy a lehető legkevesebb kérdés segítségével megtaláld az 1 helyét  $P$  permutációban. A pontozás az általad felhasznált kérdések számától függően történik.

## Interakció

A programodnak először két egész számot kell beolvasnia:  $T$ -t és  $N$ -t.  $T$  a fordulósám, ahányszor a programodat tesztelni fogod,  $N$  a  $P$  permutáció hossza.

Ezután következik  $T$  forduló:

Amikor egy forduló elkezdődik, elkezdhetsz kérdéseket feltenni. A "`""? a b""`" szöveg az  $a$  és  $b$  indexek közötti (beleértve a határokat is) számok összegére kérdez rá ( $0 \leq a \leq b \leq N - 1$ ).

Minden kérdésed után egy egész számot kapsz válaszul: a kért intervallumban lévő számok összegét.

Ha megtaláltad az 1 indexét, írd ki egy "`! i`" formájú sort, ahol  $i$  az az index, ahol  $P_i = 1$ . Miután ezt kiírtad, kezdődik a következő forduló, újabb sorozattal.

Ügyelj arra, hogy a kérdésfeltevés után a standard kimenetet ki kell írni, különben a programod időlimit túllépésként értékelhetik. Pythonban a `print()` automatikusan megteszi ezt. C++-ban a `cout << endl;` is kiírja, egy új sor kiírásával; ha `printf` függvényt használasz, akkor add ki a `fflush(stdout)` parancsot.

## Korlátok és pontozás

A programodat **egyetlen tesztesetként fogjuk tesztelni,  $N=T=1000$  értékkel**. A permutáció minden egyes tesztben garantáltan **véletlenszerűen generálódik**. Ha a megoldásod bármelyik fordulóban rosszul tippel, a beadványodat *Hibás válaszként* értékeljük. Ellenkező esetben a pontszámot a következőképpen számoljuk ki:

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

ahol  $M$  a programod által összesen feltett kérdések száma a  $T$  fordulóban. A pontszámot a legközelebbi egész számra kerekítjük. Ha a pontszám negatív lesz, az 0 pont. Ha több mint 550 000 kérdést használsz, akkor 0 pontot kapsz, ha 300 000 vagy kevesebb kérdést használsz, akkor 100 pontot kapsz. A kettő között a pontszám lineárisan növekszik.

## Tesztelő eszköz

A megoldásod tesztelésének megkönnyítése érdekében egy egyszerű eszközt biztosítunk, amelyet letölthetsz. Ezt a feladatoldal alján, az "attachments" menüpont alatt találod. Az eszköz használata opcionális, és szabadon megváltoztathatod. Vedd figyelembe, hogy a végső értékelő eltér a tesztelő eszköztől.

Példa a használatra ( $T=1000$ ,  $N=10$  esetben):

Python program esetén `solution.py` programra (általában a `pypy3 solution.py` futtatás):

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

C++ programnyelv esetén, fordítsd le a programod (például: `g++ -std=gnu++17 solution.cpp -o solution.out` paranccsal) és utána futtasd a következő utasítással az értékelőt:

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

## Példa

A példa tesztesetben,  $T = 2$  és  $N = 10$ . A két forduló közül az elsőnél a rejtett permutáció mondjuk a "6 10 8 7 9 1 2 4 5 3". Az első kérdés: ? 0 9 rákérdez az összes szám összegére, ami persze 55, és a második kérdés: ? 0 4 rákérdez a  $6 + 10 + 8 + 7 + 9 = 40$  összegre.

Értékelő kimenete	A Te kimeneted
2 10	
	? 0 9
55	
	? 0 4
40	
	? 5 5
1	
	! 5
	? 0 0
1	
	! 0