

Kje je Waldo?

Naloga	whereswaldo
Časovna omejitev	11 sekund
Omejitev spomina	1 gigabyte

Obstaja skrita permutacija P_0, P_1, \dots, P_{N-1} dolžine N , ki je zagotovljeno generirana uniformno naključno. V permutaciji se vsako število $1, 2, 3, \dots, N$ pojavi natančno enkrat, vrstni red števil pa je neznan.

Izbereš lahko položaja l in r , in vprašaš vprašanje oblike "Kakšna je vsota števil $P_l + P_{l+1} + \dots + P_r$?"

Tvoja naloga je, da najdeš položaj 1 v P s čim manj vprašanji. Tvoj program bo ocenjen glede na število vprašanj, ki jih bo uporabil.

Interakcija

Tvoj program naj najprej prebere vrstico z dvema celima številoma T in N . T je število krogov ugibanj, na katerih bo program testiran, N pa je dolžina P .

Po tem sledi T krogov:

Na začetku testa lahko začneš s postavljanjem vprašanj. Izpiši vrstico z " $? a b$ ", da vprašaš po vsoti števil med pozicijama a in vključno b ($0 \leq a \leq b \leq N - 1$).

Po vsakem vprašanju naj tvoj program prebere integer - vsoto števil znotraj intervala.

Ko najdeš položaj 1 , izpiši vrstico oblike " $! i$ ", kjer je i indeks s $P_i = 1$. Po izpisani vrstivi te oblike se bo začel naslednji krog.

Poskrbi, da izplakneš (flush) standardni izhod po zastavljanju vprašanja, sicer bo tvoj program morda ocenjen s prekoračenim časom (Time Limit Exceeded). V Pythonu `print()` izplakuje avtomatično. V C++, `cout << endl;` tudi izplakne skupaj z izpisom nove vrstice, če pa uporabljaš `printf`, uporabi `fflush(stdout)`.

Constraints and Scoring

Tvoj program bo testiran na **enem samem testnem primeru z $N = T = 1000$** . Permutacija v vsakem krogu je zagotovljena **generirana naključno**.

Če tvoja rešitev ugame narobe v katerem koli krogu, bo tvoja oddaja ocenjena kot napačen odgovor (*Wrong Answer*).

Sicer bo tvoja ocena izračunana takole:

$$\text{\text{ocena}} = \mathrm{\min}\left(220 - \frac{M}{2500}, 100\right)\text{\text{ točk}},$$

kjer je M skupno število vprašanj, ki jih tvoj program vpraša v vseh T krogih.

Ocena bo zaokrožena na najbližje celo število. Če je ocena negativna, bo obravnavana kot nič točk.

Torej, če uporabiš več kot 550 000 vprašanj bo tvoj program ocenjen z 0 točk, če pa uporabiš 300 000 vprašanj ali manj, boš prejela 100 točk. Vmes bo število točk naraščalo linearno.

Orodje za testiranje

Da ti omogočimo testiranje rešitve, ti ponujamo preprosto orodje za testiranje, ki si ga lahko preneseš. Poglej med priponke ("attachments") na dnu kattis strani z nalogo. Orodje uporablaj po želji in lahko ga spremeniš. Uradni ocenjevalnik na kattis je drugačen od orodja za testiranje.

Primer uporabe (za $T = 1000$, $N = 10$):

Za python program z imenom `solution.py` (običajno ga poženeš z `pypy3 solution.py`):

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

C++ programe najprej prevedi (na primer `g++ -std=gnu++17 solution.cpp -o solution.out`) in potem poženi:

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

Primer

Za ponazoritev vzamemo $T = 2$ in $N = 10$. V prvem od teh dveh krogov je skrita permutacija "6 10 8 7 9 1 2 4 5 3". Prvo vprašanje ? 0 9 vpraša za vsoto vseh števil, ki je 55, drugo vprašanje ? 0 4 pa vpraša po $6 + 10 + 8 + 7 + 9 = 40$.

izhod ocenjevalnika	tvoj izhod
2 10	
	? 0 9
55	
	? 0 4
40	
	? 5 5
1	
	! 5
	? 0 0
1	
	! 0