

Hvor er Waldo?

Oppgavenavn	whereswaldo
Tidsbegrensning	11 sekunder
Minnebegrensning	1 gigabyte

Det finnes en skjult permutasjon P_0, P_1, \dots, P_{N-1} av lengde N som er garantert til å være plukket uniformt tilfeldig. Permutasjonen inneholder tallene $1, 2, 3, \dots, N$ nøyaktig én gang hver, i en ukjent rekkefølge.

Du kan velge posisjoner l og r , og stille spørsmål på formen: "Hva er summen av $P_l + P_{l+1} + \dots + P_r$?"

Din oppgave er å finne posisjonen til 1 i P , ved å bruke så få spørsmål som mulig. Scoren din vil avhenge av antall spørsmål stilt.

Interaksjon

Programmet ditt skal først lese to heltall på én enkelt linje, T og N . T gir antallet runder programmet ditt vil bli testet på, og N gir lengden av P .

Etter dette følger T runder:

Når en runde starter har du anledning til å stille et spørsmål. Print en linje " $? a b$ " for å spørre om summen av tallene fra og med posisjon a , til og med posisjon b , hvor $0 \leq a \leq b \leq N - 1$.

Etter hvert spørsmål skal programmet ditt lese ett heltall, summen av tallene i intervallet.

Straks du har identifisert posisjonen til 1-tallet, print en linje på formen " $! i$ ", hvor i er indeksen slik at $P_i = 1$. Etter denne linjen er printet vil neste runde begynne.

Du må flushe output etter å ha stilt et spørsmål, ellers kan programmet ditt bli klassifisert som Time Limit Exceeded. I Python gjøres dette automatisk av `print()`. I C++ flusher `cout << endl;` output samtidig med å printe en ny linje. Anvender du `printf`, bruk `fflush(stdout)`.

Begrensninger og scoring

Programmet ditt vil bli testet mot **en enkelt test, hvor $N = T = 1000$** . Permutasjonene som opptrer i rundene er garantert til å være **generert tilfeldig**.

Hvis løsningen din gir feil svar i noen av rundene, vil programmet ditt bli rettet som *Wrong Answer*.

Ellers vil programmet ditt bli gitt poeng etter følgende formel:

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

hvor M er antall spørsmål programmet ditt har stilt totalt i løpet av alle de T rundene.

Scoren vil avrundes til nærmeste heltall. Om tallet blir negativt vil det bli sett på som null poeng.

Dermed vil du få 0 poeng om du bruker mer enn 550 000 spørsmål, og bruker du mindre enn 300 000 spørsmål får du 100 poeng. Mellom dette stiger poengscoren lineært.

Testverktøy

For å tilrettelegge for testing av løsningen din er et enkelt verktøy gjort tilgjengelig for nedlasting, se "attachments" på bunnen av oppgavesiden på Kattis. Verktøyet er frivillig å bruke, og du står fritt til å endre programmet. Merk at det offisielle retteprogrammet (engelsk: grader) på kattis er annet enn dette gitte verktøyet.

Eksempelbruk (hvor $T = 1000$, $N = 10$):

For pythonprogrammer, si `solution.py` (normalt kjørt som `python3 solution.py`):

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

For C++-programmer, kompiler først programmet (for eks. med `g++ -std=gnu++17 solution.cpp -o solution.out`) og kjør så:

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

Eksempel

I eksempeltestcasen er $T = 2$ og $N = 10$. For den første av disse to rundene, la oss si at den skjulte permutasjonen er "6 10 8 7 9 1 2 4 5 3". Det første spørsmålet ? 0 9 spør om summen av alle disse tallene, som jo er 55. Det andre spørsmålet ? 0 4 spør om $6 + 10 + 8 + 7 + 9 = 40$.

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