

## Where's Waldo?

Aufgabenname	whereswaldo
Zeitlimit	11 Sekunden
Speicherlimit	1 Gigabyte

Es gibt eine versteckte Permutation  $P_0, P_1, \dots, P_{N-1}$  der Länge  $N$ , welche garantiert einheitlich zufällig generiert zu werden. Die Permutation enthält die Zahlen  $1, 2, 3, \dots, N$  je genau einmal in einer unbekanntem Reihenfolge.

Du kannst die Positionen  $l$  und  $r$  auswählen, und dann eine Frage der Form "Was ist die Summe von  $P_l + P_{l+1} + \dots + P_r$ ?" stellen.

Deine Aufgabe ist es, die Position der 1 in  $P$  mit möglichst wenigen Fragen zu finden. Du wirst bewertet aufgrund der Anzahl an gestellten Fragen.

### Interaktion

Dein Programm sollte zuerst zwei Zahlen,  $T$  und  $N$ , aus einer Zeile einlesen.  $T$  ist die Anzahl an Runden, die dein Programm getestet wird, und  $N$  ist die Länge von  $P$ .

Danach kommen  $T$  Runden:

Wenn eine Runde beginnt, fängst du an Fragen zu stellen. Gib eine Zeile mit "? a b" aus, um über die Summe der Zahlen zwischen den Positionen  $a$  und  $b$  (inklusive) zu fragen ( $0 \leq a \leq b \leq N - 1$ ).

Nach jeder Frage sollte dein Programm eine Zahl einlesen, die Summe der Zahlen in dem Intervall.

Sobald du die Position der 1 gefunden hast, gib eine Zeile der Form "! i" aus, bei welcher  $i$  der Index ist, sodass  $P_i = 1$ . Nach der Ausgabe wird die nächste Runde beginnen.

Stelle sicher, dass du deine Ausgabe flushst, nachdem du eine Frage gestellt hast, sonst bekommt dein Programm möglicherweise ein "Timelimit exceeded". In Python flusht `print()` automatisch. In C++ flusht `cout << endl;` und zusätzlich dazu beginnt es eine neue Zeile; wenn du `printf` nutzt, nutze `fflush(stdout)`.

### Beschränkungen and Bewertung

Dein Programm wird mit **einem einzigen Testfall mit  $N = T = 1000$**  getestet. Die Permutation in jedem Test ist garantiert **zufällig generiert zu sein**.

Wenn deine Lösung in irgendeiner Runde falsch rät, wird die Einsendung mit *Wrong Answer* bewertet.

Andernfalls wird die Punktzahl wie folgt berechnet:

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

wobei  $M$  die Anzahl an Fragen über alle  $T$  Runden ist.

Die Punktzahl wird zu der nächsten Zahl gerundet. Wenn die Punktzahl negativ wird, gibt es 0 Punkte.

Du bekommst 0 Punkte, wenn du mehr als 550.000 Fragen brauchst und 100 Punkte, wenn du 300.000 oder weniger Fragen brauchst. Dazwischen wächst deine Punktzahl linear.

## Testwerkzeug

Um das Testen deiner Lösung zu vereinfachen, bieten wir ein einfaches Werkzeug an, welches du dir herunterladen kannst. Du findest dies unter "attachments" am unteren Ende der Kattis Aufgabenseite. Das Programm ist optional und du darfst es ändern. Außerdem nutzt der offizielle Grader auf Kattis ein anderes Programm.

Beispielnutzung (mit  $T=1000$ ,  $N=10$ ):

Für Python Programme führe `solution.py` aus (normalerweise ausgeführt als `pypy3 solution.py`):

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

Für C++ Programme, kompiliere es zuerst (z.B. mit `g++ -std=gnu++17 solution.cpp -o solution.out`) und führe dann folgenden Befehl aus:

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

## Beispiel

Im Beispielstestfall ist  $T = 2$  und  $N = 10$ . Für die erste dieser zwei Runden sagen wir die versteckte Permutation ist "6 10 8 7 9 1 2 4 5 3". Die erste Frage ? 0 9 fragt für die Summe aller Zahlen, welche tatsächlich 55 ist und die zweite Frage ? 0 4 fragt nach  $6 + 10 + 8 + 7 + 9 = 40$ .

Graderausgabe	deine Ausgabe
2 10	
	? 0 9
55	
	? 0 4
40	
	? 5 5
1	
	! 5
	? 0 0
1	
	! 0