

Де Вальдо?

Назва задачі	whereswaldo
Обмеження на час виконання	11 с
Обмеження на використання пам'яті	1024 МБ

Існує прихована перестановка P_0, P_1, \dots, P_{N-1} довжини N , яка гарантовано була згенерована випадковим чином. Перестановка містить числа $1, 2, 3, \dots, N$ рівно по одному разу в якомусь невідомому порядку.

Ви можете вибрати дві позиції l і r і робити запити типу: «Яка сума $P_l + P_{l+1} + \dots + P_r$?»

Ваше завдання знайти позицію числа 1 в P , використовуючи якомога менше запитів. Вам буде нараховано бали в залежності від кількості виконаних запитів.

Формат взаємодії

Ваша програма повинна спочатку прочитати два цілих числа в одному рядку, T і N . T — це кількість раундів, на яких тестуватиметься ваша програма, а N — це довжина P .

Після цього задаються T раундів:

Коли починається раунд, ви можете почати виконувати запити. Виведіть рядок із “? a b”, щоб запитати про суму чисел на відрізку від a до b включно ($0 \leq a \leq b \leq N - 1$).

Після кожного запиту ваша програма має зчитати ціле число — суму чисел у відрізку.

Коли ви знайдете позицію числа 1, виведіть рядок у формі “! i”, де i — такий індекс, що $P_i = 1$. Після того, як ви виведете це, розпочнеться наступний раунд.

Обов'язково очистіть буфер стандартного виводу після того, як виконаєте запит, інакше ваше рішення може перевищити ліміт часу. У Python `print()` скидає буфер автоматично. У C++, `cout << endl;` також очищає та виводить новий рядок; якщо використовується `printf`, використовуйте `fflush(stdout)`.

Обмеження та оцінювання

Ваше рішення буде перевірено на **одному тестовому прикладі**, з $N = T = 1000$. Перестановка в кожному тесті гарантовано буде **згенерована випадковим чином**.

Якщо в будь-якому раунді ваше рішення не вгадає, воно буде оцінене як *Неправильна відповідь*.

В іншому випадку оцінка буде розрахована таким чином:

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

де M — загальна кількість запитів, які ваша програма задає за всі T раундів.

Кількість балів буде округлена до найближчого цілого числа. Якщо кількість балів стає від'ємною, вона буде розглядатися як нуль балів.

Таким чином, якщо ви використовуєте понад 550 000 запитань, ви отримуєте 0 балів, а якщо ви використовуєте 300 000 або менше запитань ви отримуєте 100 балів. Між ними ваша оцінка лінійно зростає.

Прикладовий інструмент перевірки

Щоб полегшити тестування вашого рішення, ми пропонуємо простий інструмент, який ви можете завантажити. Дивіться «вкладення» внизу сторінки задачі на Kattis. Інструмент необов'язковий для використання, і ви можете його змінювати. Зауважте, що офіційна програма грейдера на kattis відрізняється від цього інструменту перевірки.

Приклад використання (з $T=1000$, $N=10$):

Для програм на Python, з назвою, скажімо `solution.py` (зазвичай запускається як `py3 solution.py`):

```
python3 testing_tool.py py3 solution.py <<<"1000 10"
```

Для програм на C++ спочатку скомпілюйте його (наприклад, за допомогою `g++ -std=gnu++17 solution.cpp -o solution.out`), а потім запустіть:

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

Приклад використання

У тестовому прикладі $T = 2$ і $N = 10$. Для першого з цих двох раундів, скажімо, прихована перестановка «6 10 8 7 9 1 2 4 5 3». Перший запит ? 0 9 як відповідь має суму всіх чисел, яка дорівнює 55, а другий запит ? 0 4 як відповідь має $6 + 10 + 8 + 7 + 9 = 40$.

вивід грейдера	вивід рішення
2 10	
	? 0 9
55	
	? 0 4
40	
	? 5 5
1	
	! 5
	? 0 0
1	
	! 0