



Where's Waldo?

Problem Name	whereswaldo
Time Limit	11 seconds
Memory Limit	1 gigabyte

Существует скрытая перестановка P_0, P_1, \dots, P_{N-1} длины N , которая гарантированно равномерно генерируется случайным образом. Перестановка содержит числа $1, 2, 3, \dots, N$ ровно по одному разу в каком-то неизвестном порядке.

Вы можете выбрать позиции l и r и задать вопросы вида: «Какова сумма $P_l + P_{l+1} + \dots + P_r$?»

Ваша задача — найти позицию 1 в перестановке P , используя как можно меньше вопросов. Ваше решение будет оцениваться в зависимости от количества использованных вопросов.

Interaction

Ваша программа должна сначала прочитать два целых числа в одной строке T и N . T — это количество раундов, на которых будет тестироваться ваша программа, а N — это длина P .

После этого идут T раундов:

Когда раунд начинается, вы можете начать задавать вопросы. Выведите строку «? a b», чтобы узнать сумму чисел между позициями a и b включительно ($0 \leq a \leq b \leq N - 1$).

После каждого вопроса ваша программа должна считывать целое число - сумму чисел в подотрезке.

Как только вы нашли позицию 1, выведите строку вида «! i», где i — индекс такой, что $P_i = 1$. После того, как вы выведете это, начнется следующий раунд.

Обязательно сбросьте стандартный вывод после того, как зададите вопрос, иначе ваша программа может быть оценена как превышение лимита времени. В Python `print()` сбрасывается автоматически. В C++ `cout << endl;` также сбрасывает в дополнение к печати новой строки; если вы используете `printf`, используйте `fflush(stdout)`.

Constraints and Scoring

Ваша программа будет проверена на **одном тесте** с $N = T = 1000$. Перестановка в каждом тесте гарантированно будет **сгенерирована случайным образом**.

Если ваше решение окажется ошибочным в любом из раундов, ваше решение будет оценено как *Неправильный ответ*.

В противном случае оценка будет рассчитываться следующим образом:

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

где M — общее количество вопросов, которые ваша программа задает за все T раундов.

Оценка будет округлена до ближайшего целого числа. Если оценка становится отрицательной, она будет считаться нулевой.

Таким образом, если вы используете более 550 000 вопросов, вы получите 0 баллов, а если вы используете 300 000 или меньше вопросов, вы получите 100 баллов. Между ними ваш счет растёт линейно.

Testing Tool

Чтобы упростить тестирование вашего решения, мы предоставляем простой инструмент, который вы можете скачать. См. «attachments» на странице с задачей. Инструмент тестирования является необязательным для использования, и вы можете изменить его. Обратите внимание, что официальная программа оценки на kattis отличается от инструмента тестирования.

Пример использования (с $T=1000$, $N=10$):

Для программ на Python скажем `solution.py` (обычно запускается как `python3 solution.py`):

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

Для программ C++ сначала скомпилируйте их. (например, с помощью `g++ -std=gnu++17 solution.cpp -o solution.out`) а затем запустите:

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

Example

В тестовом примере $T = 2$ и $N = 10$. Скажем, для первого из этих двух раундов скрытая перестановка «6 10 8 7 9 1 2 4 5 3». Первый вопрос ? 0 9 запрашивает сумму всех чисел, которая равна 55, а второй вопрос ? 0 4 просит $6 + 10 + 8 + 7 + 9 = 40$.

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