

## Cadê o Waldo?

Nome do Problema	cadeowaldo
Limite de Tempo	11 segundos
Limite de Memória	1 gigabyte

Há uma permutação secreta  $P_0, P_1, \dots, P_{N-1}$  de comprimento  $N$ , que tem a garantia de ser gerada de forma uniformemente aleatória. A permutação contém os números  $1, 2, 3, \dots, N$  exatamente uma vez cada, em alguma ordem desconhecida.

Você pode escolher as posições  $l$  e  $r$  e fazer perguntas do tipo: "Qual é a soma de  $P_l + P_{l+1} + \dots + P_r$ ?"

Sua tarefa é encontrar a posição do número 1 em  $P$  usando o menor número possível de perguntas. Sua pontuação dependerá do número de perguntas usadas.

### Interação

Seu programa deve primeiramente ler dois números inteiros em uma única linha,  $T$  e  $N$ .  $T$  é o número de rodadas em que seu programa será testado e  $N$  é o comprimento de  $P$ .

Depois disso, vêm  $T$  rodadas:

Quando uma rodada começa, você pode começar a fazer perguntas. Imprima uma linha com "? a b" para perguntar qual a soma dos números entre as posições  $a$  e  $b$  inclusive ( $0 \leq a \leq b \leq N - 1$ ).

Após cada pergunta, seu programa deve ler um número inteiro, a soma dos números no intervalo.

Depois de encontrar a posição do 1, imprima uma linha no formato "! i", onde  $i$  é o índice tal que  $P_i = 1$ . Depois de imprimir isso, a próxima rodada começará.

Certifique-se de dar `flush` na saída padrão após fazer uma pergunta, caso contrário, seu programa poderá ser julgado como *Time Limit Exceeded*. Em Python, `print()` faz o `flush` automaticamente. Em C++, o `cout << endl;` também faz o `flush`, além de imprimir uma quebra de linha; se estiver usando `printf`, use `fflush(stdout)`.

### Limites e Pontuação

Seu programa será testado em **um único caso de teste, com  $N = T = 1000$** . É garantido que a permutação em cada teste é **gerada aleatoriamente**.

Se a sua solução errar qualquer uma das rodadas, sua submissão será julgada como *Wrong Answer*.

Caso contrário, a pontuação será calculada da seguinte forma:

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

onde  $M$  é o número total de perguntas que seu programa faz nas  $T$  rodadas como um todo.

A pontuação será arredondada para o número inteiro mais próximo. Se a pontuação for negativa, ela será tratada como zero pontos.

Portanto, se você usar mais de 550 000 perguntas, receberá 0 pontos, e se usar 300 000 ou menos perguntas, você receberá 100 pontos. Entre os dois, sua pontuação cresce linearmente.

## Corretor Exemplo

Para facilitar o teste de sua solução, fornecemos um corretor exemplo que você pode baixar. Consulte "attachments" (anexos) na parte inferior da página do problema no kattis. O uso do corretor exemplo é opcional e você tem permissão para alterá-lo. Observe que o corretor oficial de avaliação no kattis é diferente do corretor exemplo.

Exemplo de uso (com  $T=1000$ ,  $N=10$ ):

Para programas em python, use `solution.py` (normalmente executado como `pypy3 solution.py`):

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

Para programas em C++, primeiro compile-o (por exemplo, com `g++ -std=gnu++17 solution.cpp -o solution.out`) e depois execute:

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

## Exemplo

No exemplo de teste,  $T = 2$  e  $N = 10$ . Para a primeira dessas duas rodadas, digamos que a permutação oculta seja "6 10 8 7 9 1 2 4 5 3". A primeira pergunta ? 0 9 pede a soma de todos os números, que é de fato 55, e a segunda pergunta ? 0 4 pede  $6 + 10 + 8 + 7 + 9 = 40$ .

saída do corretor	sua saída
2 10	
	? 0 9
55	
	? 0 4
40	
	? 5 5
1	
	! 5
	? 0 0
1	
	! 0