

## Biscoitos dos Gêmeos

Nome do problema	Biscoitos dos Gêmeos
Arquivo de entrada	entrada padrão
Arquivo de saída	saída padrão
Tempo limite	1 segundo
Limite de memória	256 megabytes

*Este é um problema interativo. Seu programa se comunicará com nosso avaliador alternadamente escrevendo mensagens na saída padrão e lendo mensagens da entrada padrão.*

Sophie está preparando a festa de aniversário de seus irmãos gêmeos. Os gêmeos adoram biscoitos. Para seu aniversário, eles gostariam de experimentar algo novo: biscoitos da Unique Cookie Tastiness Company (UCTC, Companhia dos Biscoitos de Nível de Sabor Único).

Cada biscoito produzido pela UCTC tem um valor inteiro de nível de sabor entre 1 e  $10^{16}$  inclusive. Como os gêmeos da Sophie ficam com ciúmes um do outro, cada um deles deve receber biscoitos com a mesma soma de valores de nível de sabor.

A UCTC só aceita pedidos de **exatamente**  $n$  biscoitos. Em cada pedido, o cliente especifica o nível de sabor de cada um dos  $n$  biscoitos que deseja.

Permanecendo fiel ao seu nome, a Unique Cookie Tastiness Company se recusa a produzir dois biscoitos com o mesmo nível de sabor para o mesmo cliente. Sophie deve certificar-se de nunca pedir o mesmo nível de sabor duas vezes - nem no mesmo pedido, nem em dois pedidos diferentes. Sophie nunca comprou da UCTC antes, portanto ela pode pedir cada nível de sabor disponível uma vez.

Há mais um obstáculo no caminho da Sophie: é um fato conhecido que o serviço de entrega da UCTC é horrível. Sempre que um cliente pede  $n$  biscoitos, apenas um desses  $n$  biscoitos chega de fato ao cliente. O resto é comido ao longo do caminho pelos funcionários do serviço de entrega. A cliente não pode influenciar qual dos  $n$  biscoitos encomendados será realmente entregue a ela.

Como o aniversário está se aproximando rapidamente, Sophie tem tempo para fazer, no máximo, 101 pedidos. Sua tarefa é ajudá-la.

Mais especificamente, você deve fazer o seguinte:

1. Primeiro, encomende os biscoitos. Você pode fazer, no máximo, 101 pedidos, cada um consistindo em exatamente  $n$  valores de nível de sabor desejados. Você faz um pedido de cada vez. **Imediatamente após cada pedido, você recebe o nível de sabor do biscoito que realmente foi recebido.**

Lembre-se que você não pode usar o mesmo valor de nível de sabor várias vezes, mesmo em vários pedidos. (Em particular, se você encomendar um biscoito com nível de sabor  $t$  mas ele não chegar, você **não poderá** encomendar um biscoito com o mesmo nível de sabor novamente).

2. Então, divida os biscoitos. Uma vez que você tenha recebido biscoitos suficientes, você deve distribuir **alguns** dos biscoitos recebidos entre os gêmeos. Ambos os gêmeos devem receber pelo menos um biscoito e cada gêmeo deve receber biscoitos com a mesma soma total de níveis de sabor. **Você não precisa usar todos os biscoitos recebidos!**

## Saída

Cada vez que o seu programa escrever uma ou mais linhas na saída padrão, você deve seguir essa ação **limpando a saída com flush**. Isto é necessário para garantir que os dados que você escreveu na saída cheguem imediatamente ao avaliador.

Exemplos de como isto pode ser feito:

- Em C++, há múltiplas opções.
  - `fflush(stdout);`
  - `std::cout << std::flush;`
  - `std::cout << std::endl;` (note que isto também imprime uma nova linha)
  - leitura de `std::cin` também limpa a saída com flush
- em Java, você pode utilizar `System.out.flush()`
- em Python, você pode utilizar `sys.stdout.flush()`

## Protocolo de Interação

Seu programa deve realizar a seguinte sequência de ações:

1. Leia o valor  $n$  da entrada padrão.
2. No máximo 101 vezes:
  1. Primeiro, escreva uma linha descrevendo um pedido de  $n$  biscoitos para a saída padrão.
  2. Em seguida, leia o nível de sabor do biscoito que você recebeu da entrada padrão. É garantido que este valor está entre os  $n$  valores que estavam no pedido atual.
3. Escreva na saída três linhas descrevendo uma forma válida de dar alguns dos

biscoitos que você recebeu para os gêmeos.

O avaliador escreverá cada número inteiro em uma linha separada.

Para encomendar biscoitos, escreva na saída uma única linha com ? seguido de  $n$  inteiros: os valores de nível de sabor dos biscoitos que você deseja encomendar. Escreva um único espaço antes de cada um dos  $n$  números inteiros.

Lembre-se que você pode fazer no máximo 101 pedidos e que não é permitido utilizar o mesmo valor de nível de sabor duas vezes.

Uma vez que você tenha encomendado biscoitos suficientes, escreva na saída as três linhas finais que descrevem quais biscoitos a Sophie deve dar aos gêmeos.

A primeira destas linhas deve ter a forma " $m k$ " com  $m, k > 0$ : o número de biscoitos que o primeiro e o segundo gêmeo, respectivamente, devem receber.

A segunda destas linhas deve conter  $m$  inteiros separados por um único espaço: os valores de nível de sabor dos biscoitos que o primeiro gêmeo deve receber.

Da mesma forma, a terceira linha deve conter  $k$  inteiros separados por um único espaço: os valores de nível de sabor dos biscoitos que o segundo gêmeo deve receber.

A saída deve satisfazer as seguintes condições:

1. Cada gêmeo deve receber pelo menos um biscoito.
2. Cada gêmeo deve receber biscoitos com o mesmo nível de sabor total.
3. Somente biscoitos que você realmente recebeu após seus pedidos podem ser utilizados.
4. Cada um desses biscoitos só pode ser dado a, no máximo, um dos gêmeos.

Qualquer saída que satisfaça estas condições será aceita. Em particular, você pode escrever na saída os biscoitos selecionados em qualquer ordem.

Após escrever as três linhas finais na saída, limpe a saída com flush uma última vez e depois **finalize seu programa normalmente**.

## Pontuação

Sub-tarefa 1 (8 pontos):  $n = 1$

Sub-tarefa 2 (9 pontos):  $1 \leq n \leq 2$

Sub-tarefa 3 (18 pontos):  $1 \leq n \leq 25$

Sub-tarefa 4 (16 pontos):  $1 \leq n \leq 200$

Sub-tarefa 5 (13 pontos):  $1 \leq n \leq 1000$

Sub-tarefa 6 (36 pontos):  $1 \leq n \leq 5000$

## Exemplos

entrada padrão	saída padrão
1	? 13
13	? 7
7	? 31
31	? 12
12	? 5
5	? 3
3	! 2 3
	7 13
	12 5 3
2	? 3 7
7	? 2 8
2	? 1 5
5	! 2 1
	2 5
	7

## Observações

Exemplos de entrada e saída devem ser lidos linha por linha. Seu programa alternadamente lê um valor da entrada padrão e escreve uma linha (ou três linhas no final) na saída padrão.

O avaliador escolhe qual biscoito deve retornar arbitrariamente. Isto significa que o avaliador pode ser adaptável às suas consultas em alguns testes, mas também pode escolher os biscoitos aleatoriamente em outros testes. Em particular, para  $n = 2$ , se você fizer a mesma sequência de pedidos que no segundo exemplo, você poderá receber um conjunto diferente de biscoitos.