

Galletas gemelas

| Problem name | Twin Cookies |
|--------------|-----------------|
| Input file | standard input |
| Output file | standard output |
| Time limit | 1 second |
| Memory limit | 256 megabytes |

Este es un problema interactivo. Tu programa se comunicará con el evaluador alternando mensajes escritos hacia la salida estándar y leyendo mensajes de la entrada estándar.

Sophie está preparando la fiesta de cumpleaños de sus gemelas. Las gemelas aman las galletas. Para su cumpleaños quieren probar algo nuevo: galletas de la Compañía de Sabrosura Única de Galletas (UCTC por sus siglas en inglés).

Cada galleta producida por la UCTC tiene una sabrosura que es un valor entero entre 1 y 10^{16} inclusive. Como las gemelas se ponen celosas una de la otra, las galletas que reciba cada gemela deben sumar la misma sabrosura.

La UCTC solo acepta pedidos de **exactamente** n galletas. En cada pedido el cliente especifica la sabrosura de cada una de las n galletas que quiere.

Siendo fiel a su nombre, la UCTC se rehúsa a producir dos galletas con la misma sabrosura para el mismo cliente. Sophie debe estar segura que nunca pide la misma sabrosura dos veces -- ni en el mismo pedido, ni en pedidos diferentes. Sophie nunca ha comprado galletas en la UCTC antes, así que puede pedir cada sabrosura disponible una vez.

Hay un obstáculo más en el camino de Sophie: Se sabe que el servicio de entrega a domicilio de la UCTC es terrible. Cada vez que un cliente pide n galletas, solo una de esas n galletas llegan al cliente. Las demás se las comen los repartidores en el camino. El cliente no puede influenciar cuál de las n galletas que pidió es la que va a recibir.

Como el cumpleaños está muy cercano, Sophie tiene tiempo de hacer a lo más 101 pedidos. Tu tarea es ayudarla.

Específicamente, debes hacer lo siguiente:

1. Primero pide las galletas. Puedes hacer a lo más 101 pedidos, cada uno con exactamente n sabrosuras deseadas. Haces un pedido a la vez. **Inmediatamente después de cada orden, se te dará la sabrosura de la galleta que recibiste.**

Recuerda que no puedes usar la misma sabrosura más de una vez, incluso en pedidos distintos (particularmente, si pides una galleta con sabrosura t y no te llega, **no puedes** pedir una galleta con la misma sabrosura de nuevo).

2. Después divide las galletas. Una vez que tengas suficientes galletas, debes distribuir **algunas** de las galletas recibidas entre las gemelas. Ambas gemelas deben recibir al menos una galleta y cada gemela debe recibir galletas con la misma sabrosura total. **¡No es necesario usar todas las galletas que recibiste!**

Salida

Cada vez que tu programa escriba una o más líneas a la salida, debes hacer **flush del stream de salida**. Esto es necesario para asegurarte que los datos que escribiste lleguen inmediatamente al evaluador.

Ejemplos de cómo puedes hacer esto:

- En C++, hay varias opciones.
 - `fflush(stdout);`
 - `std::cout << std::flush;`
 - `std::cout << std::endl;` (recuerda que esto también imprime una línea nueva)
 - leyendo con `std::cin` también hace flush del stream de salida.
- En Java, puedes usar `System.out.flush()`
- En Python, puedes usar `sys.stdout.flush()`

Protocolo de interacción

Tu programa debe hacer la siguiente secuencia de acciones:

1. Leer el valor de n de entrada estándar.
2. A lo más 101 veces:
 1. Primero escribe (a la salida estándar) una línea que represente un pedido de n galletas.
 2. Luego lee la sabrosura de la galleta que recibiste. Se te garantiza que este valor está entre los n valores que están en el pedido actual.
 3. Escribe tres líneas describiendo una forma válida de repartir a las gemelas las galletas que recibiste.

El evaluador escribirá cada entero en una línea separada.

Para pedir galletas, escribe solo una línea con un ? seguido de n enteros: las sabrosuras de las galletas que quieres pedir. Escribe un espacio antes de cada uno de los n enteros.

Recuerda que puedes hacer a lo más 101 pedidos y que no puedes usar la misma sabrosura más de una vez.

Una vez que hayas pedido suficientes galletas, escribe las últimas tres líneas que describen qué galletas les dará Sophie a las gemelas.

En la primera línea debes escribir " $m k$ " donde $m, k > 0$: el número de las galletas que reciben la primera y la segunda gemela respectivamente.

En la segunda línea debes escribir m enteros separados por un espacio: la sabrosura de las galletas que recibe la primera gemela.

En la tercera línea debes escribir k enteros separados por un espacio: la sabrosura de las galletas que recibe la segunda gemela.

La salida debe satisfacer las siguientes condiciones:

1. Cada gemela debe recibir al menos una galleta.
2. Cada gemela debe recibir galletas que sumen la misma sabrosura total.
3. Solo puedes usar las galletas que recibiste después de hacer tu pedido.
4. Cada galleta se puede dar solo una vez a las gemelas.

Cada salida que cumpla estas condiciones, será aceptada. Esto quiere decir que puedes imprimir las galletas seleccionadas en cualquier orden.

Después de que imprimas las últimas tres líneas, haz flush del stream de salida una última vez y **termina ejecución normalmente**.

Evaluación

Subtarea 1 (8 points): $n = 1$

Subtarea 2 (9 points): $1 \leq n \leq 2$

Subtarea 3 (18 points): $1 \leq n \leq 25$

Subtarea 4 (16 points): $1 \leq n \leq 200$

Subtarea 5 (13 points): $1 \leq n \leq 1000$

Subtarea 6 (36 points): $1 \leq n \leq 5000$

Ejemplos

| entrada | salida |
|---------|--------|
| 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 |

Nota

Los ejemplos de entrada y salida se deben leer línea por línea. Tu programa alterna el leer un valor de entrada y escribir una línea de salida (tres líneas al final).

El evaluador escoge qué galleta regresar arbitrariamente. Esto quiere decir que el evaluador puede ser adaptativo a tus preguntas en algunos casos, pero también puede escoger galletas aleatoriamente para otros casos, por ejemplo para $n = 2$, si haces la misma secuencia de pedidos que en el segundo ejemplo, puedes recibir un conjunto diferente de galletas.