Twin Cookies

Nombre del problema	Twin Cookies
Archivo de entrada	entrada estándard
Archivo de salida	salida estándard
Límite de tiempo	1 segundo
Límite de memoria	256 megabytes

Este es un problema interactivo. Tu programa debe comunicarse con el grader escribiendo y leyendo de manera alterna mensajes de la salida estándard y de la entrada estándard.

Sofía se prepara para la fiesta de aniversario de sus hijas mellizas. A las mellizas les encantan las galletas. Es su aniversario y quieren probar algo nuevo: galletas de la CGSU, la Compañía de Galletas de Sabores Únicos.

Cada galleta producida por la CGSU tiene un entero que define el valor del nivel de sabor entre 1 y 10^{16}, ambos incluidos. Como las hijas de Sofía se pone celosas entre ellas, cada una tiene que recibir galletas con la misma suma de niveles de sabor.

La CGSU solo acepta pedidos de **exactamente** n galletas. En cada pedido el cliente especifica el nivel de sabor de cada una de las n que quiere.

Para ser fieles a su nombre, la Compañía de Galletas de Sabores Únicos se niega a producir dos galletas con el mismo nivel de sabor para un mismo cliente. Sofía tiene que asegurarse de que nunca pide dos veces el mismo nivel —ni en el mismo pedido, ni en dos pedidos distintos. Sofía nunca ha comprado en CGSU antes, con lo que puede pedir cualquier valor de nivel de sabor una sola vez.

Hay un último obstáculo en el camino de Sofía: todo el mundo sabe que la compañía que hace las entregas de CGSU es horrible. Cuando un cliente pide n galletas, sólo una de esas galletas llega al cliente. El resto son devoradas por los empleados del servicio de entrega. El cliente no puede influenciar cual de las n galletas pedidas será entregada.

Como el aniversario se acerca, Sofía tiene tiempo de hacer como mucho 101 pedidos. Tu tarea es ayudarla.

Más específicamente, deberías de hacer lo siguiente:

1. Primero, hacer un pedido de galletas. Puedes hacer como mucho 101 pedidos, cada uno conteniendo n galletas de valor de sabor. Haces los pedidos uno cada vez. **Justo después de cada pedido se te dará el valor de la única galleta que realmente se te entrega.**

Acuerdate que no se puede pedir el mismo valor de nivel de sabor múltiples veces, ni en pedidos distintos. (En particular, si tu pedido contiene una galleta con valor t que no te llega, **no puedes** volver a pedir ese valor de galleta.)

1. Después, divide las galletas. Una vez has recibido suficientes galletas, deberías distribuir **algunas** de las galletas recibidas entre las dos mellizas. Las dos mellizas tienen que recibir al menos una galleta, y cada melliza tiene que recibir galletas con el mismo valor total de sabor. **¡No tienes la obligación de usar todas las galletas recibidas!**

Salida

Cada vez que tu programa escriba una o más líneas a la salida estándard, tienes que usar la acción de **flushing la salida estándard**. Esto es necesario para asegurarte de que los datos llegan al grader inmediatamente.

Ejemplos de como se puede hacer:

- En C++, tienes distintas opciones.
  - `fflush(stdout);`
  - `std::cout << std::flush;`
  - `std::cout << std::endl;` (Notese que esto imprime una línea extra)
  - leer de `std::cin` también flushes la salida
- en Java, puedes usar `System.out.flush()`
- en Python, puedes usar `sys.stdout.flush()`

Protocolo de interacción

Tu programa debería realizar la siguiente secuencia de acciones:

1. Leer el valor de n de la entrada estándar.
2. Como mucho 101 veces:
   1. Primero, escribe una línea con tu pedido de las n galletas en la salida estándard.
   2. Luego, lee el valor de sabor que recibas en la entrada estándard. Se te garantiza que estará entre los n valores de tu pedido actual.
3. Escribe tres líneas describiendo una distribución valida de las galletas recibidas para las mellizas.

El grader escribirá cada entero en una línea separada.

Para hacer pedidos, escribe un signo ? seguido de n enteros: los valores de nivel de sabor de las galletas que quieres pedir. Escribe un solo espacio entre antes de cada uno de los n enteros.

Acuerdate que como mucho puedes hacer 101 pedidos, y que no se te permite usar el mismo valor de nivel de sabor dos veces.

Una vez hayas pedido suficientes galletas, escribe tres líneas describiendo cómo Sofía da las galletas a las mellizas.

La primera de estas líneas tiene que tener la forma "! m k" con $m, k > 0$: el número de galletas que darás para la primera y la segunda de las mellizas, respectivamente.

La segunda de estas líneas debería contener m enteros separados por un espacio simple: el nivel de sabor de las galletas que recibirá la primera melliza.

Similarmente, la tercera línea tendría que contener k enteros separados por un espacio simple: el nivel de sabor de las galletas que recibirá la segunda melliza.

La salida debe satisfacer las siguientes condiciones:

1. Cada melliza debe recibir como mínimo una galleta.
2. Cada melliza tiene que recibir galletas con el mismo valor total de sabor.
3. Solo galletas que se hayan entregado después de tus pedidos se pueden usar.
4. Cada una de esas galletas se pueden dar a como mucho una melliza.

Cualquier solución que satisfaga estas condiciones será aceptada. En particular, puedes escribir las galletas seleccionadas en cualquier orden.

Después de cada salida, flush el flujo de salida una última vez y luego **finaliza tu programa de manera estándard**.

Puntuación

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

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

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

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

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

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

Ejemplos

standard input	standard output
1 13 7 31 12 5 3	? 13 ? 7 ? 31 ? 12 ? 5 ? 3 ! 2 3 7 13 12 5 3
2 7 2 5	? 3 7 ? 2 8 ? 1 5 ! 2 1 2 5 7

Nota

Los ejemplos de entrada y salida se tendrían que leer línea por línea. Tu programa alternativamente lee un valor de la entrada estándar y escribe una línea (o tres al final) a la salida estándard.

El grader elige que galleta devolver arbitrariamente. Esto quiere decir que el grader puede adaptarse a tus pedidos en algunos tests, pero también puede elegir las galletas de manera aleatoria en otros tests. En particular para $n = 2$, si tu haces la misma secuencia de pedidos como en el segundo ejemplo, se te puede devolver un conjunto diferente de galletas.