

Два печива

Назва	Twin Cookies
Вхідний файл	стандартний ввід
Вихідний файл	стандартний вивід
Обмеження часу	1 секунда
Обмеження пам'яті	256 мегабайтів

Це інтерактивна задача. Ваша програма буде комунікувати з нашим грейдером, виводячи дані у стандартний вивід і зчитуючи дані зі стандартного вводу.

Софі готує день народження своїх близнюків. Близнюки люблять печиво. На день народження вони хотіли б спробувати щось нове: печиво від Unique Cookie Tastiness Company (UCTC).

Кожне печиво, виготовлене UCTC, має цілочисельне значення солодкості від 1 до 10^{16} включно. Оскільки близнюки Софі заздрять одне одному, кожен з них повинен отримати печиво з однаковою сумарною солодкістю.

UCTC приймає замовлення лише з **рівно** n печив. У кожному замовленні клієнт вказує солодкість кожного з n печив, які він хоче.

Залишаючись вірним своєму імені, Unique Cookie Tastiness Company відмовляється виробляти два печива однакової солодкості для одного клієнта. Софі повинна переконатися, що вона ніколи не замовляє одну і ту ж солодкість двічі - ні в одному замовленні, ні в двох різних замовленнях. Софі ніколи раніше не купувала печиво в UCTC, тому вона може замовити печиво кожної солодкості не більше одного разу.

На шляху Софі є ще одна перешкода: загальновідомо, що служба доставки UCTC є жахливою. Щоразу, коли клієнт замовляє n печив, лише одне із цих n печив насправді приходить до клієнта. Решту їдять по дорозі працівники служби доставки. Софі не може впливати на те, яке із замовлених n печив буде фактично доставлено їй.

Оскільки день народження дуже скоро, Софі має час зробити щонайбільше 101 замовлення. Ваше завдання допомогти їй.

Більш конкретно, вам слід зробити наступне:

1. Спочатку замовте печиво. Ви можете зробити не більше 101 замовлення, кожне з яких складається рівно з n бажаних значень солодкості. Ви робите одне замовлення за раз. **Відразу після кожного замовлення ви отримуєте солодкість одного печива, яке ви фактично отримали.** Пам'ятайте, що вам не дозволяється використовувати одне і те ж значення солодкості кілька разів, навіть для різних замовлень. (Зокрема, якщо ви замовляєте печиво з деякою солодкістю t , але воно не прийшло, ви **не зможете** замовити печиво з такою ж солодкістю знову.)
2. Потім розділіть печива. Отримавши достатню кількість печива, ви повинні розділити **деякі** печива між близнюками. Обидва близнюки повинні отримати принаймні одне печиво, а кожен близнюк повинен отримати печиво з однаковою сумарною солодкістю. **Вам не обов'язково використовувати всі отримані печива.**

Вихідні дані

Кожного разу, коли ваша програма виводить один або кілька рядків у стандартний вивід, ви повинні **виконувати операцію 'flush'**. Це необхідно для того, щоб дані, які ви вивели, відразу надходили до грейдера.

Приклади того, як це можна зробити:

- В C++, є кілька варіантів як це зробити.
 - `fflush(stdout);`
 - `std::cout << std::flush;`
 - `std::cout << std::endl;` (зверніть увагу, що ця операція також виводить лишній рядок)
 - зчитування з `std::cin` також робить цю операцію
- В Java, ви можете використовувати `System.out.flush()`
- В Python, ви можете використовувати `sys.stdout.flush()`

Протокол взаємодії

Ваша програма повинна виконувати наступну послідовність дій:

1. Зчитати число n з стандартного вводу.
2. Не більше 101 разу:
 1. Спочатку виведіть один рядок, що описує n печив, до стандартного виводу.
 2. Потім зчитайте солодкість печива, яке ви отримали зі стандартного вводу. Гарантується, що це значення є серед n значень, які були щойно виведені учасником.
3. Виведіть три рядки, які описують один валідний спосіб подарувати деякі з отриманих печив близнюкам.

Грейдер виведе кожне ціле число в окремий рядок.

Щоб замовити печиво, введіть один рядок із знаком $?$, а потім n цілих чисел: значення солодкості печив, які ви хочете замовити. Виведіть по одному пробілу перед кожним з n цілих чисел.

Пам'ятайте, що ви можете зробити щонайбільше 101 замовлення і що вам не дозволяється використовувати одне і те ж значення солодкості двічі.

Після того, як ви замовили достатню кількість печива, введіть останні три рядки, які описують, які печива Софі повинна дати двійнятам.

Перший з цих рядків повинен мати вигляд " $! m k$ " де $m, k > 0$: кількість печива, які повинні отримати, відповідно, перший і другий близнюки.

Другий із цих рядків повинен містити m цілих чисел, розділених одинарними пробілами: значення солодкості печива, які повинен отримати перший близнюк.

Третій із цих рядків повинен містити k цілих чисел, розділених одинарними пробілами: значення солодкості печива, які повинен отримати другий близнюк.

Вивід повинен відповідати наступним умовам:

1. Кожен близнюк повинен отримати принаймні одне печиво.
2. Кожен близнюк повинен отримати печиво з однаковою сумарною солодкістю.
3. Використовувати можна лише печива, які ви фактично отримали після замовлень.
4. Кожне з цих печив може бути надане лише одному з близнюків.

Буде прийнятий будь-який вивід, який відповідає цим умовам. Зокрема, ви можете виводити вибрані печива у будь-якому порядку.

Після того, як ви введете останні три рядки, останній раз виконайте операцію 'flush', а потім **нормально завершіть програму**.

Оцінювання

Блок 1 (8 балів): $n = 1$

Блок 2 (9 балів): $1 \leq n \leq 2$

Блок 3 (18 балів): $1 \leq n \leq 25$

Блок 4 (16 балів): $1 \leq n \leq 200$

Блок 5 (13 балів): $1 \leq n \leq 1000$

Блок 6 (36 балів): $1 \leq n \leq 5000$

Приклади

стандартний ввід	стандартний вивід
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

Примітка

Приклади введення та виведення слід читати рядок за рядком. Ваша програма по черзі зчитує одне значення зі стандартного вводу і виводить один рядок (або три рядки в кінці) у стандартний вивід.

Грейдер вибирає, яке печиво повертати довільно. Це означає, що грейдер може бути адаптивним до ваших запитів у деяких тестах, але він також може вибрати випадкові печива в інших тестах. Зокрема, для $n = 2$, якщо ви зробите ту ж послідовність замовлень, що і у другому прикладі, ви можете отримати інший набір печив.