

## Twin Cookies

Име на задачата	Twin Cookies
Входен файл	стандартен вход
Изходен файл	стандартен изход
Ограничение по време	1 секунда
Ограничение по памет	256 мегабайта

*Това е интерактивна задача. Вашата програма ще комуникира с грейдъра на журито, като алтернативно ще пише съобщения на стандартния изход и ще чете съобщения от стандартния вход.*

София приготвя партито по случай рождения ден на своите близначки. Те обожават бисквитките. За рождения си ден, те биха желали да пробват нещо ново: бисквитки от Уникално бисквиткова-вкусовата компания (аббревиатура на английски - UCTC).

Всяка бисквитка, приготвена от UCTC има цяло число като стойност на вкуса - между 1 и  $10^{16}$ , включително. Тъй като близначките на София започват да ревнуват една от друга, всяка от тях трябва да получи бисквитки, чиято сума от стойностите на вкусовете е една и съща.

UCTC единствено приема поръчки за **точно**  $n$  бисквитки. За всяка поръчка, клиентът посочва вкусовата стойност на  $n$ -те бисквитки, които иска.

В духа на името си, Уникално бисквиткова-вкусовата компания отказва да приготвя две бисквитки с една и съща вкусова стойност за един и същ клиент. София трябва да се погрижи да не поръчва с една и съща вкусова стойност повече от веднъж -- нито в една и съща поръчка, нито в различни поръчки. Тя никога не е поръчвала от UCTC преди, така че тя може да поръча от всяка налична вкусова стойност по веднъж.

Има още едно препятствие пред София: добре известно е, че услугите по доставяне на UCTC са отворителни. Винаги, когато клиент поръча  $n$  бисквитки, само една от тези  $n$  бисквитки достига до клиента. Останалите се изяждат по пътя от служителите по достяването. Клиентът не може да влияе коя от  $n$ -те поръчани бисквитки ще пристигне до него.

Понеже рожденият ден приближава все по-бързо, София има време да направи най-много 101 поръчки. Вашата задача е да ѝ помогнете.

По-точно, трябва да направите следното:

1. Първо, трябва да поръчате бисквитките. Може да направите най-много 101 поръчки, всяка състояща се от точно  $n$  желани вкусови стойности. Правите една поръчка на веднъж. **Веднага след всяка поръчка ви се дава вкусовата стойност на бисквитката, която всъщност получавате.**

Запомнете, че не ви е позволено да използвате една и съща вкусова стойност няколко пъти, дори и при различни поръчки. (По-точно, ако поръчате бисквитка с вкусова стойност  $t$ , но тя не пристигне, **не** може да поръчате бисквитка с тази стойност отново.)

2. След това, трябва да разделите бисквитките. Когато прецените, че сте получили, достатъчно бисквитки, трябва да разделите **някои** от получените бисквитки между близначките. Те двете трябва да получат поне една бисквитка и всяка трябва да получи бисквитки с една и съща сума на вкусовите стойности. **Не е задължително да използвате всички бисквитки, които сте получили!**

## Изход

Всеки път, когато вашата програма отпечата един или повече редове на стандартния изход, трябва да направите **flush на изходния поток**. Това е нужно, за да осигурите, че данните, които сте отпечатали, достигат до грейдъра незабавно.

Примери как това може да бъде направено:

- в C++, може да използвате `fflush(stdout)` или `cout << flush`
- в Java, може да използвате `System.out.flush()`
- в Python, може да използвате `sys.stdout.flush()`

## Протокол за комуникация

Вашата програма трябва да направи следната поредица от действия:

1. Въвеждане на стойността на  $n$  от стандартния вход.
2. Най-много 101 пъти:
  1. Първо, отпечатайте един ред на стандартния изход, описващ поръчка на  $n$  бисквитки.
  2. След което въведете от стандартния вход, вкусовата стойност на бисквитката, която сте получили. Гарантирано е, че тази стойност е измежду  $n$ -те стойности, които са били в последната поръчка.

3. Отпечатайте три реда, описващи един валиден начин за разделяне на получените бисквитки между близначките.

Грейдърът ще отпечатава всяко цяло число на отделен ред.

За да поръчате бисквитки, отпечатайте един ред с  $?$ , последвана от  $n$  цели числа: вкусовите стойности на бисквитките, които искате да поръчате. Отпечатайте по един интервал преди всяко от  $n$ -те цели числа.

Напомняме, че можете да правите най-много 101 поръчки и че не е позволено да използвате една и съща вкусова стойност повече от веднъж.

След като сте поръчали достатъчно бисквитки, отпечатайте последните три реда, които описват кои бисквитки ще раздаде София на близначките.

Първият от тези редове трябва да е във формата " $! m k$ ", като  $m, k > 0$ : броят на бисквитките, които ще получат първата и втората близначка, съответно.

Вторият от редовете трябва да съдържа  $m$  цели числа, разделени с по един интервал: вкусовите стойности на бисквитките, които получава първата близначка.

Подобно, третият ред трябва да съдържа  $k$  цели числа, разделени с по един интервал: вкусовите стойности на бисквитките, които получава втората близначка.

Изходът трябва да спазва следните условия:

1. Всяка близначка трябва да получи поне една бисквитка.
2. Близначките трябва да получат бисквитки с една и съща обща вкусова стойност.
3. Само бисквитки, които са получени от поръчките, могат да бъдат използвани.
4. Всяка от получените бисквитки, може да бъде дадена само на една близначка.

Всеки изход, който спазва тези условия, ще бъде приет за верен. Също така, може да отпечатате бисквитките за всяка близначка в произволен ред.

След като отпечатате последните три реда, направете flush на изходния поток за последно и накрая **прекратете нормално вашата програма**.

## Оценяване

Подзадача 1 (8 точки):  $n = 1$

Подзадача 2 (9 точки):  $1 \leq n \leq 2$

Подзадача 3 (18 точки):  $1 \leq n \leq 25$

Подзадача 4 (16 точки):  $1 \leq n \leq 200$

Подзадача 5 (13 точки):  $1 \leq n \leq 1000$

Подзадача 6 (36 точки):  $1 \leq n \leq 5000$

## Примери

примерен вход	примерен изход
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

## Обяснение

Примерите за вход и изход трябва да бъдат гледани ред по ред. Вашата програма алтернативно чете една стойност от стандартния вход и отпечатва един ред на стандартния изход (или три реда накрая).

Грейдърът избира произволно коя бисквитка ще върне. Това означава, че грейдърът може да е адаптивен за някои ваши заявки, а за други да избира произволни бисквитките. По-точно, за  $n = 2$ , ако направите същата последователност от поръчки, като във втория пример, може да получите различни бисквитки.