

## Ciasteczka dla bliźniaków

Nazwa angielska	Twin Cookies
Plik wejściowy	Wejście standardowe
Plik wyjściowy	Wyjście standardowe
Limit czasowy	1 sekunda
Limit pamięciowy	256 megabajtów

*To zadanie jest interaktywne. Twój program będzie komunikował się z systemem testującym naprzemiennie wypisując komunikaty na standardowe wyjście i wczytując jego komunikaty ze standardowego wejścia.*

Zofia przygotowuje przyjęcie urodzinowe dla swoich bliźniaków. Bliźniacy uwielbiają cukierki. Na swoje urodziny dostaną coś specjalnego: ciasteczka z firmy Unique Cookie Tastiness Company (UCTC) (Unikalne Smaki Ciasteczek).

Każde ciasteczko wyprodukowane przez UCTC ma smak liczby naturalnej między 1, a  $10^{16}$  włącznie. Bliźniaki bywają zazdrosne, więc każdy z nich musi dostać identyczną łączną sumę smaków.

UCTC przyjmuje zamówienia wyłącznie na **dokładnie**  $n$  ciasteczek. W każdym zamówieniu klient zamawia dowolnie wybrane  $n$  różnych smaków.

Będąc wierną swojej nazwie, firma UCPC odmawia produkcji dwóch ciasteczek o tym samym smaku dla jednego klienta. Zofia musi dbać o to, żeby żaden smak się nie powtórzył --- ani w tym samym zamówieniu, ani w żadnych dwóch złożonych przez nią zamówieniach. Zofia nigdy jeszcze nie kupowała ciasteczek w tej firmie, więc może zamówić każdy smak raz.

Zofia ma jeszcze jedną przeszkodę do pokonania. Jak powszechnie wiadomo, kurierzy firmy UCPC są okropni. Z każdego zamówienia, tylko jedno z zamówionych  $n$  ciasteczek dociera do klienta, ponieważ cała reszta jest zjadana przez kuriera po drodze. Klient w dodatku nie ma wpływu na to, które ciasteczko przetrwa transport.

Urodziny zbliżają się szybko i Zofia może zrobić co najwyżej 101 zamówień. Poprosiła Cię o pomoc.

Nieco dokładniej, powinnaś wykonać następujące czynności:

1. Najpierw zamówienia. Wykonujesz co najwyżej 101 zamówień, każde składające się z dokładnie  $n$  smaków. Zamówienia składasz pojedynczo. **Natychmiast po złożeniu każdego zamówienia dostaniesz informację o smaku ciasteczka, które z niego dotrze do Zofii.** Pamiętaj, że nie wolno Ci użyć tego samego smaku więcej niż jeden raz, nawet w innych zamówieniach. (W szczególności, jeśli zamówisz pewien smak  $t$ , a on nie dotrze, już nigdy później **nie wolno** Ci zamówić smaku  $t$ ).
2. Następnie podziel cukierki. Gdy tylko uda Ci się zebrać stosowny zbiór smaków, Wybierz z nich **pewien podzbiór** i utwórz dwa zestawy o identycznych sumach smaków. **Nie musisz używać wszystkich ciasteczek, które masz.**

## Wyjście

Za każdym razem, gdy Twój program wypisze jedną lub więcej wierszy na standardowe wyjście, musisz zakończyć tę akcję **synchronizacją strumienia wyjściowego**. To zapewni natychmiastową wysyłkę danych do sprawdzaczki.

Przykłady, jak można to zrobić:

- w C++, możesz użyć `fflush(stdout)` lub `cout << flush`
- w Javie, możesz użyć `System.out.flush()`
- w Pythonie, możesz użyć `sys.stdout.flush()`

## Komunikacja z systemem testującym

Twój program powinien wykonać akcje w następującej kolejności

1. Wczytaj wartość  $n$  ze standardowego wejścia.
2. Co najwyżej 101 razy:
  1. Najpierw na standardowe wyjście wypisz jeden wiersz opisujący zamówienie  $n$  smaków.
  2. Potem odczytaj ze standardowego wejścia smak, który zostanie dostarczony. Gwarantujemy, że wartość ta będzie jedną z podanych przez Ciebie w bieżącym zamówieniu.
3. Wypisz trzy wiersze określające poprawne zestawy ciasteczek dla bliźniaków.

System w jednym wierszu będzie wypisywać zawsze jedną wartość.

Aby utworzyć zamówienie, zacznij wiersz od znaku `?`, po którym powinno nastąpić  $n$  liczb całkowitych: smaki, które zamierzasz zamówić. Smaki powinny być porozdzielane pojedynczymi spacjami.

Pamiętaj, że możesz wykonać co najwyżej 101 zamówień i że nie wolno Ci użyć tego samego smaku dwa razy.

Gdy tylko będziesz miała wystarczający zbiór smaków, wypisz trzy wiersze opisujące,

jak Zofia mogłaby utworzyć zestawy dla każdego z bliźniaków.

Pierwszy z tych wierszy powinien mieć postać " $m\ k$ " przy czym  $m, k > 0$ , to liczby ciasteczek w pierwszym i drugim zestawie.

Drugi z tych wierszy powinien zawierać  $m$  liczb całkowitych oddzielonych pojedynczymi spacjami: smaki, które znajdują się w pierwszym zestawie.

Podobnie, trzeci wiersz powinien zawierać  $k$  liczb całkowitych oddzielonych pojedynczymi spacjami: smaki, które znajdują się w drugim zestawie.

Wyjście musi spełniać następujące warunki:

1. Każdy bliźniak powinien dostać przynajmniej jedno ciasteczko.
2. Obaj bliźniacy powinni dostać tę samą sumę smaków.
3. Można tworzyć zestawy wyłącznie z tych smaków, które dotarły.
4. Każdy smak, który dotarł, może się znaleźć w co najwyżej jednym zestawie.

Każde wyjście, które będzie spełniało te warunki zostanie zaakceptowane. W szczególności możesz w wierszu wypisać smaki w dowolnej kolejności.

Po wypisaniu tych trzech wierszy zsynchronizuj wyjście po raz ostatni i **zakończ działanie programu**.

## Ocenianie

Podzadanie 1 (8 punktów):  $n = 1$

Podzadanie 2 (9 punktów):  $1 \leq n \leq 2$

Podzadanie 3 (18 punktów):  $1 \leq n \leq 25$

Podzadanie 4 (16 punktów):  $1 \leq n \leq 200$

Podzadanie 5 (13 punktów):  $1 \leq n \leq 1000$

Podzadanie 6 (36 punktów):  $1 \leq n \leq 5000$

## Przykłady

standardowe wejście	standardowe wyjście
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

## Wyjaśnienie do przykładów

Powyższe przykłady wejść i wyjść czytamy wiersz po wierszu. Program na przemian czyta wartość ze standardowego wejścia i wypisuje jeden wiersz (a na końcu trzy wiersze) na standardowe wyjście.

System testujący wybiera ciasteczko w sposób nieprzewidywalny. Na przykład dla  $n = 2$ , jeśli wykonasz ten sam ciąg zamówień, co w drugim przykładzie, możesz skończyć z innym zestawem ciasteczek.