

Twin Cookies

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

Aceasta este o problemă interactivă. Programul tău va comunica cu graderul prin scrierea alternantă a mesajelor către standard output citirea mesajelor din standard input.

Sophie se pregătește pentru serbarea gemenilor săi. Gemenii iubesc prăjiturile. Pentru serbarea de ziua lor, ei vor să guste ceva nou: biscuiți de la Unique Cookie Tastiness Company (UCTC).

fiecare biscuit produs de UCTC are o valoare întreagă de gustoșenie între 1 și 10^{16} inclusiv. Deoarece gemenii Sophiei sunt reciproc geloși, fiecare din ei va primi biscuiți cu același inidator total de gustoșenie.

UCTC acceptă comenzi de **exact** n biscuiți. În fiecare comandă clientul specifică gradul de gustoșenie a fiecăruia dintre cei n biscuiți comandați.

Rămânând fideli numelui lor Unique Cookie Tastiness Company refuză să producă doi biscuiți cu același grad de gustoșenie pentru unul și același client. Sophie trebuie să fie sigură că nu a comandat biscuiți cu același grad de gustoșenie -- nici în aceeași comandă, nici în două comenzi diferite. Sophie nu a comandat anterior de la UCTC astfel că ea poate comenzi biscuiți cu orice grad de gustoșenie.

Mai este un obstacol pentru Sophie: e bine cunoscut că serviciul de livrare a UCTC este îngrozitor. Când un client comandă n biscuiți, numai unul dintre acești n biscuiți ajunge în realitate la client. Restul sunt mâncați pe drum de angajații de la livrare. Clientul nu poate influența dintre cei n biscuiți comandați să ajungă la destinație.

Pentru că serbarea se apropie repede, Sophie nu are timp pentru 101 comenzi. Misiunea ta este să o ajuți.

Mai exact, trebuie să faci următoarele:

1. Mai întâi comandă biscuiții. Poți face cel mult 101 comenzi, fiecare din exact n valori de gustoșenie dorite. Faci o comandă într-o unitate de timp . **Imediat după fiecare comandă primești indicele de gustoșenie a biscuitului, pe care l-ai primit real**

Ține minte că nu ai dreptul să folosești același indice de gustoșenie de mai multe ori, chiar în comenzi diferite. (În particular, dacă ai comandat un biscuit cu indicele t dar acesta nu a ajuns, **nu mai poți** comanda un biscuit cu același nivel de gustoșenie din nou.)

2. Apoi împarte biscuiții. Odată ce ai primit suficienți biscuiți, urmează să distribuie **unii** biscuiți între gemeni. Fiecare dintre ei va primi cel puțin câte un biscuit, și fiecare va primi biscuiți cu același indice total de gustoșenie. **Nu este obligatoriu să folosești toți biscuiții!**

Output

De fiecare dată când programul tău afișează în output sau câteva linii din standard output, trebuie să urmezi și operație **flushare a fluxului de ieșire**. Este necesar pentru a fi sigură că datele afișate ajung la grader imediat.

Exemple cum poate fi făcut:

- În C++, sunt opțiuni multiple.
 - `fflush(stdout);`
 - `std::cout << std::flush;`
 - `std::cout << std::endl;` (se notat că se printează un NewLine - extra)
 - citirea din `std::cin` de asemenea curăță buferul
- în Java, poți folosi `System.out.flush()`
- în Python, poți folosi `sys.stdout.flush()`

Protocolul de Interacțiune

Programul tău va trebui să efectueze următoarea succesiune de acțiuni:

1. Citească valoarea n din standard input.
2. De cel mult 101 ori:
 1. Mai întâi, scrie o linie descriind o comandă n biscuiți în standard output.
 2. Apoi citește nivelul de gustoșenie a biscuitului primit din inputul standard. Se garantează că această valoare este dintre cele n valori folosite în comanda curentă.
3. afișați trei linii care descriu o metodă validă de a da unii dintre biscuiții recepționați gemenilor.

Graderul va scrie fiecare număr întreg într-o linie separată.

pentru a comanda biscuiții afișează o singură linie cu n urmat de n întregi: gradele de gustosenie ale biscuiților pe care dorești să îi comanzi. Afișează un singur spațiu înaintea fiecăruia dintre cele n numere.

Nu uita că poți efectua cel mult 101 comenzi și că nu ai voie să utilizezi aceeași valoare a gustoseniei de două ori.

După ce ai comandat suficienți biscuiți, afișază ultimele trei rânduri care descriu ce biscuiți ar trebui să le dea Sophie gemenilor.

Prima linie dintre acele trei trebuie să aibă forma " $m k$ " cu $m, k > 0$: numărul de biscuiți pe care ar trebui să le primească primul și respectiv al doilea dintre gemeni.

A doua linie va conține m numere întregi separate prin câte un spațiu: valorile de gust ale biscuiților pe care ar trebui să le primească primul dintre gemeni.

Similar a treia linie va conține k numere întregi separate prin câte un spațiu: valorile de gust ale biscuiților pe care ar trebui să le primească al doilea dintre gemeni..

Outputul trebuie să satisfacă următoarele condiții:

1. Fiecare gemeni ar trebui să primească cel puțin un biscuite.
2. Fiecare gemeni ar trebui să primească biscuiți cu același grad de gustosenie total.
3. Pot fi utilizate doar biscuiții pe care i-ați primit efectiv după livrarea comenzii.
4. Fiecare dintre biscuiți poate fi dat doar cel mult unuia dintre gemeni.

Orice soluție care îndeplinește aceste condiții va fi acceptată. În special, puteți afișa biscuiții selectați în orice ordine.

După ce ați lansat ultimele trei linii, curățați fluxul de ieșire pentru ultima dată și apoi **terminați programul normal** .

Scoring

Subtask 1 (8 puncte): $n = 1$

Subtask 2 (9 puncte): $1 \leq n \leq 2$

Subtask 3 (18 puncte): $1 \leq n \leq 25$

Subtask 4 (16 puncte): $1 \leq n \leq 200$

Subtask 5 (13 puncte): $1 \leq n \leq 1000$

Subtask 6 (36 puncte): $1 \leq n \leq 5000$

Exemple

standard input	standard output
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

Note

Exemplele de input și output urmează să fie citite linie după linie. Programul alternativ citește o valoare din inputul standard și scrie o linie(sau trei la sfârșit) în standard output.

Grader-ul alege în mod aleator care biscuite să rămână. Aceasta înseamnă că graderul trebuie să fie adaptiv la interogările tale în unele teste, dar el va putea alege biscuiții aleator în alte teste. În particular, pentru $n = 2$, dacă faci aceeași succesiune de comenzi ca în al doilea exemplu, este posibil să primești un set diferit de biscuiți.