

Kekse für Zwillinge

Name der Aufgabe	Twin Cookies
Eingabe	standard input
Ausgabe	standard output
Zeitlimit	1 Sekunde
Speicherlimit	256 MB

Dies ist eine interaktive Aufgabe. Dein Programm wird mit unserem Grader kommunizieren, indem es abwechselnd Nachrichten auf die Standardausgabe schreibt und Nachrichten von der Standardeingabe liest.

Sophie trifft Vorbereitungen für die Geburtstagsparty ihrer Zwillingstöchter. Die Zwillinge lieben Kekse! Für ihren Geburtstag würden sie daher gern etwas neues ausprobieren: Kekse von der Unique Cookie Tastiness Company (UCTC).

Jeder der UCTC-Kekse hat einen ganzzahligen Geschmackswert von 1 bis 10^{16} (inklusive). Die Summe der Geschmackswerte der Kekse, die beide Zwillingstöchter erhalten, muss exakt gleich sein, weil sonst natürlich eine von ihnen eifersüchtig würde.

Die UCTC akzeptiert nur Bestellungen von **genau** n Keksen. In jeder Bestellung muss die Käuferin angeben, welche Geschmackswerte die n bestellten Kekse haben sollen.

Um ihrem Namen gerecht zu werden weigert sich die Unique Cookie Tastiness Company, zwei oder mehr Kekse des gleichen Geschmackswertes für die gleiche Käuferin zu produzieren. Also muss Sophie darauf achten, nicht zweimal den gleichen Geschmackswert zu bestellen -- weder in der gleichen, noch in zwei verschiedenen Bestellungen. Sophie hat zuvor noch nie bei UCTC bestellt, daher kann sie jeden Geschmackswert genau einmal bestellen.

Es gibt allerdings noch ein weiteres Problem: Bekanntlich ist der Lieferservice von UCTC grauenhaft. Wenn eine Käuferin n Kekse bestellt, kommt nur einer dieser n Kekse tatsächlich bei der Käuferin an. Der Rest wird auf dem Weg von Angestellten des Lieferservices gegessen. Die Käuferin kann leider nicht beeinflussen, welcher der n Kekse tatsächlich bei ihr ankommt.

Der Geburtstag naht, deshalb hat Sophie gerade so noch Zeit für höchstens 101

Bestellungen! Deine Aufgabe ist es, ihr zu helfen.

Genauer gesagt sollst du Folgendes tun:

1. Zuerst bestellst du die Kekse. Du darfst höchstens 101 Bestellungen machen, jede davon bestehend aus genau n gewünschten Geschmackswerten. Dabei machst du immer eine Bestellung auf einmal. **Direkt nach jeder Bestellung erfährst du, welcher Keks tatsächlich bei dir angekommen ist.**

Denk daran, dass du niemals den gleichen Geschmackswert bestellen darfst, nicht einmal in verschiedenen Bestellungen. (Insbesondere darfst du einen Geschmackswert **nicht** nochmal bestellen, wenn der Keks beim ersten Mal nicht angekommen ist.)

2. Dann verteilst du die Kekse. Nachdem du genug Kekse erhalten hast, solltest du **einige** der erhaltenen Kekse an die Zwillingsschwestern verteilen. Beide müssen natürlich mindestens einen Keks erhalten, und die Summe der Geschmackswerte der Kekse, die beide Zwillingstöchter erhalten, muss exakt gleich sein. **Du musst dabei nicht alle Kekse, die du erhalten hast, benutzen.**

Ausgabe

Jedes Mal, wenn du etwas ausgibst, musst du den **Outputstream flushen**. Das ist nötig, damit deine Ausgabe unseren Grader sofort erreicht.

Das geht zum Beispiel so:

- In C++ gibt es verschiedene Möglichkeiten.
 - `fflush(stdout);`
 - `std::cout << std::flush;`
 - `std::cout << std::endl;` (beachte, dass das zusätzlich einen Zeilenumbruch ausgibt!)
 - Von `std::cin` einlesen flusht auch die Ausgabe (aber nur, sofern du nicht `std::cout.tie(0)` verwendet hast).
- In Java kannst du `System.out.flush()` verwenden.
- In Python kannst du `sys.stdout.flush()` verwenden.

Interaktion

Dein Programm sollte Folgendes tun:

1. Die Ganzzahl n aus der Standardeingabe lesen.
2. Höchstens 101 Mal:
 1. Zuerst gibst du eine Zeile auf der Standardausgabe aus, die eine Bestellung von n Keksen beschreibt.
 2. Anschliessend liest du den Geschmackswert des Kekses, den du erhalten

hast, von der Standardeingabe. Es wird garantiert, dass dies einer der Geschmackswerte ist, die du in dieser Bestellung bestellt hast.

3. Gib drei Zeilen aus, die eine valide Möglichkeit beschreiben, einige der Kekse an die Zwillinge zu verteilen.

Der Grader wird jede Ganzzahl in eine eigene Zeile schreiben.

Um Kekse zu bestellen, gib eine einzelne Zeile mit ? aus, gefolgt von n durch einzelne Leerzeichen getrennten Ganzzahlen: Den Geschmackswerten der Kekse, die du bestellen möchtest.

Denk daran, dass du höchstens 101 Bestellungen machen darfst, und dass du den gleichen Geschmackswert nicht mehrfach verwenden kannst.

Nachdem du genug Kekse bestellt hast, gib die letzten drei Zeilen aus, die beschreiben, welche Kekse Sophie ihren Zwillingstöchtern geben soll.

Die erste dieser Zeilen sollte die Form " $! m k$ " mit $m, k > 0$ haben: Die Anzahl der Kekse, die die erste und zweite Tochter jeweils erhalten soll, in dieser Reihenfolge.

Die zweite dieser Zeilen sollte m Ganzzahlen, die durch einzelne Leerzeichen getrennt sind, enthalten: Die Geschmackswerte der Kekse, die die erste Zwillingstochter erhalten soll.

Analog dazu soll die dritte dieser Zeilen k durch einzelne Leerzeichen getrennte Ganzzahlen enthalten: Die Geschmackswerte der Kekse, die die zweite Zwillingstochter erhält.

Die Ausgabe muss die folgenden Bedingungen erfüllen:

1. Jede der Zwillingstöchter muss mindestens einen Keks erhalten.
2. Jede der Zwillingstöchter muss Kekse mit der selben Summe der Geschmackswerte erhalten.
3. Nur Kekse, die bei den Bestellungen tatsächlich angekommen sind, dürfen verwendet werden.
4. Jeder Keks kann höchstens einer der Schwestern gegeben werden.

Jede Ausgabe, die diese Bedingungen erfüllt, wird akzeptiert. Insbesondere kannst du die Kekse in beliebiger Reihenfolge ausgeben.

Nachdem du die drei letzten Zeilen ausgegeben hast, musst du ein letztes Mal den Outputstream flushen und dann dein **Programm normal beenden**.

Teilaufgaben

Teilaufgabe 1 (8 Punkte): $n = 1$

Teilaufgabe 2 (9 Punkte): $1 \leq n \leq 2$

Teilaufgabe 3 (18 Punkte): $1 \leq n \leq 25$

Teilaufgabe 4 (16 Punkte): $1 \leq n \leq 200$

Teilaufgabe 5 (13 Punkte): $1 \leq n \leq 1000$

Teilaufgabe 6 (36 Punkte): $1 \leq n \leq 5000$

Beispiele

Eingabe	Ausgabe
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

Die Beispiele von Ein- und Ausgabe sollten Zeile für Zeile gelesen werden. Das Programm liest abwechselnd einen Wert von der Standardeingabe und schreibt eine Zeile (bzw. drei Zeilen am Ende) in die Standardausgabe.

Der Grader wählt willkürlich, welcher Keks geliefert wird. Das bedeutet, der Grader kann sich in manchen Testfällen an deine Ausgaben anpassen, und in manchen Testfällen einfach zufällig einen Keks wählen. Insbesondere könntest du für $n = 2$, selbst wenn du die gleichen Bestellungen wie im Beispiel machst, eine andere Menge Kekse erhalten.