

Търсене на фалшиви монети

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 512 megabytes

Това е интерактивна задача.

Даден ви е комплект от n златни монети, измежду които има k фалшиви. Монетите са поставени в определен ред. Предполагамото тегло на i -тата монета е равно на i грама. Ако монетата е фалшива, теглото и е равно на 0 грама.

Забранено е да се пипат и местят монетите и единствената достъпна за вас операция е да се избере някое $1 \leq p \leq n$ и да се претеглят първите p монети. В резултат на това ще ви бъде казано колко е истинското общо тегло на тези монети.

Като използвате минимален брой операции определете, кои k монети са фалшиви. Броят на точките ще зависи от броя на заявките, зададени от вашето решение. Подробности вижте в системата за оценяване.

Interaction Protocol

Всеки тест се състои от t игри, в които вие трябва да разберете кои монети са фалшиви. От първия ред се задава единствено цяло число t ($1 \leq t \leq 50$) — броя игри. Всяка игра е съставена от взаимодействия в описания по-долу формат. След завършването на всички игри, вашата програма трябва да спре работа.

В началото на всяка игра ви се дават две цели числа n и k ($1 \leq n \leq 10^9$, $1 \leq k \leq \min(100, n)$). След това вие може да зададете няколко заявки за претегляне.

За да зададете заявка за претегляне изведете `"?p"`. В резултат на това ще ви бъде върнато единствено цяло число a . Ако $a = -1$, вашата програма е превишила допустимия лимит на заявки за претегляне в играта и трябва веднага да спре. На всяка игра е разрешено да се направят не повече от 3500 заявки за претегляне. Иначе $a \geq 0$ е истинското общо тегло на монетите $1, 2, \dots, p$.

За да обявите отгатнатото множество монети, изведете `"!i1 i2 ... ik"` където $1 \leq i_1, i_2, \dots, i_k \leq n$ са различни индекси на фалшиви монети в произволен ред. В резултат на това ще ви бъде върнато единствено цяло число a . Ако $a = -1$, вашият отговор е неправилен и вашата програма трябва веднага да спре. Иначе $a = 1$, и вашата програма трябва да продължи взаимодействието със следващата игра или да спре, ако това е била последната игра.

Обърнете внимание, че интерактора играе **адаптивно**. Не се гарантира, че множеството от фалшиви монети е фиксирано преди началото на играта. Единственото, което се гарантира е, че в произволен момент от време отговорите, които са дадени по време на играта, съответстват на поне едно множество от фалшиви монети. Вашият отговор на играта е правилен, ако той съответства на всички отговори на заявки, зададени в течение на играта, а също не съществува нито едно друго множество от фалшиви монети, което съответства на всички отговори.

След извеждане на всяко действие на вашата програма изведете знак за нов ред. След извеждане на всяко действие на вашата програма изчистете изходния поток.

Ако използвате `"writeln"` в Паскал, `"cout << ... << endl"` в C++, `"System.out.println"` в Java, `"print"` в Python, `"Console.WriteLine"` в C#, то почистването на изходния поток става автоматично и не е необходимо да се прави нищо друго. Ако използвате друг начин на изход, се препоръчва да се почисти изходния поток. Обърнете внимание, че знак за нов ред трябва да се изведе във всеки от случаите.

Example

standard input	standard output
2	
3 2	? 3
2	! 1 3
1	
10 4	? 5
13	? 6
13	? 8
20	? 10
29	! 10 8 6 2
1	

Note

В първата игра монетите 1, 3 са фалшиви. Тогава, истинските тегла на монетите са $[0, 2, 0]$. С помощта на една заявка вие ще разберете общото тегло на 2, след което еднозначно ще може да възстановите множеството на фалшивите монети.

Във втората игра монетите 2, 6, 8, 10 са фалшиви. Тогава, истинските тегла на монетите са $[1, 0, 3, 4, 5, 0, 7, 0, 9, 0]$. По отговора на заявките за претегляне може еднозначно да се възстанови множеството от фалшиви монети.

Scoring

Тестовете към тази задача се състоят от 6 групи. Нека q е броят на заявките за претегляне, които е направило решението по време на една игра.

Точките за първите 5 групи се получават само при преминаване на всички тестове в групата и всички тестове от необходимите групи. За всяка от първите 5 групи е фиксирано някакво число $maxQ$. Тестовете от първите 5 групи се приемат за преминати, ако $q \leq maxQ$.

Точките за всяка игра в последната група са равни на $\min\left(50, \left\lfloor 50\sqrt{\frac{k+30}{q}} \right\rfloor\right)$. Общият брой точки за теста е равен на минималния брой точки получени от всички игри. Общият брой точки за последната група е равен на минималния брой точки за всички тестове в тази група.

Обърнете внимание, че решението получава 100 точки, ако то прави $\leq k + 30$ заявки за претегляне за всички тестове за всички игри.

Група	Точки	Доп. ограничения			Необх. групи	Коментари
		n	k	$maxQ$		
0	0	–	–	$maxQ = 3500$	–	Тестовете от условието.
1	5	$n \leq 1000$	–	$maxQ = 1000$	0	
2	9	$n \leq 1000$	–	$maxQ = 600$	0, 1	
3	10	–	$k \leq 30$	$maxQ = 1000$	0	
4	13	–	$k = 3$	$maxQ = 33$	–	
5	13	–	$k = 4$	$maxQ = 34$	–	
6	≤ 50	–	–	$maxQ = 3500$	–	Частични точки.