Задача A. Устойчивая сортировка пар

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мебибайт

В этой задаче нужно отсортировать пары чисел, заданных во входных данных. Зададим *устойчивый порядок* на парах следующим образом. Пусть (a_i, b_i) и (a_j, b_j) — две пары из исходной последовательности, i, j — номера этих пар.

Будем говорить, что пара (a_i, b_i) меньше пары (a_j, b_j), в двух случаях:

1. если $a_i < a_j$,
2. если $a_i = a_j$ и $i < j$.

Таким образом, требуется отсортировать последовательность пар так, чтобы первые числа пар были расположены в неубывающем порядке, и при этом сортировка была устойчивой.

Формат входных данных

В первой строке входных данных задано целое число n ($1 \leqslant n \leqslant 5000$). В следующих n строках заданы через пробел по два целых числа, a_i и b_i ($1 \leqslant a_i, b_i \leqslant 10^9$).

Формат выходных данных

В первых n строках выведите пары чисел в требуемом порядке.

Примеры

стандартный ввод	стандартный вывод
3 1 2 3 1 1 1	1 2 1 1 3 1
4 4 1 3 2 2 2 3 3	2 2 3 2 3 3 4 1

Задача B. Субъективная ценность

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мебибайт

Двое пиратов, Рая и Сэм, делят $2n$ сокровищ. Каждый пират собирается взять себе ровно половину: n сокровищ.

У каждого сокровища i есть ценность r_i для Раи и ценность s_i для Сэма. Если сокровище достаётся Рае, то это сокровище получает стоимость r_i, а если Сэму, то стоимость s_i. Как пиратам поделить сокровища, чтобы сумма стоимостей оказалась максимально возможной?

В первой строке задано число n — количество сокровищ, которые достанутся каждому пирату (от 2 до 10). Во второй строке записаны $2n$ положительных чисел $r_1, r_2, \ldots, r_{2n}$, разделённые пробелами — ценности сокровищ для Раи (от 1 до 10). В третьей строке записаны $2n$ положительных чисел $s_1, s_2, \ldots, s_{2n}$, разделённые пробелами — ценности сокровищ для Сэма (от 1 до 10).

Выведите одно число: максимальную суммарную стоимость.

Примеры

стандартный ввод	стандартный вывод
2 1 2 3 4 5 6 7 8	18
3 3 5 6 8 1 3 8 2 6 8 4 5	36

Задача C. Раствор

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мебибайт

Перед химиком Колей стоит непростая задача. Для химического эксперимента ему требуется поместить в колбу объёма w как можно большее количество определённого вещества. Это вещество содержится в n пробирках, наполненных растворами; общий объём раствора в i-й пробирке равен q_i, а количество нужного вещества во всём этом растворе — d_i. Считается, что вещество в пробирке равномерно распределено по раствору.

Из каждой пробирки Коля может перелить всё содержимое, или любую его часть, в колбу; различные растворы в колбе смешиваются, а количество вещества и общий объём раствора в колбе складываются. Общий объём раствора, получающегося в колбе, не должен превышать объём колбы w; Коля не может переливать растворы, кроме как из пробирок в колбу. Выясните, какое максимальное количество вещества можно поместить в колбу.

Формат входных данных

В первой строке входных данных заданы через пробел два целых числа n и w — количество пробирок и объём колбы, соответственно ($1 \leqslant n \leqslant 100\,000$, $1 \leqslant w \leqslant 10^9$). Следующие n строк содержат параметры пробирок; i-я из этих строк описывает i-ю пробирку двумя целыми числами d_i и q_i — это количество вещества и общий объём пробирки, соответственно ($1 \leqslant d_i, q_i \leqslant 10^9$). Для удобства записи количество вещества измеряется в одних условных единицах, а объём колбы и общий объём пробирок — в других.

Формат выходных данных

В первой строке выведите максимальное возможное количество вещества в полученном растворе. При выводе представьте это количество в виде смешанной дроби в формате «a+b/c», где a, b и c — целые числа, $a \geqslant 0$, $0 \leqslant b < c$ и $c > 1$. Числа b и c должны быть взаимно просты. Если $a = 0$, то запись должна выглядеть как «b/c». Если $b = 0$, то запись должна состоять только из числа «a».

Примеры

стандартный ввод	*стандартный вывод*
2 5 3 2 2 5	4+1/5
1 2 2 5	4/5
1 3 3 2	3

Задача D. Белоснежка и n гномов

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мебибайт

«Ну не гномы, а наказание какое-то!» — подумала Белоснежка, в очередной раз пытаясь уложить гномов спать. Одного уложишь — другой уже проснулся! И так всю ночь.

У Белоснежки n гномов, и все они очень разные. Она знает, что для того, чтобы уложить спать i-го гнома, нужно a_i минут, и после этого он будет спать ровно b_i минут. Помогите Белоснежке узнать, может ли она получить хотя бы минутку отдыха, когда все гномы будут спать, и если да, то в каком порядке для этого нужно укладывать гномов спать.

Например, пусть есть всего два гнома, $a_1 = 1$, $b_1 = 10$, $a_2 = 10$, $b_2 = 20$. Если Белоснежка сначала начнёт укладывать первого гнома, то потом ей потребуется целых 10 минут, чтобы уложить второго, а за это время проснётся первый. Если же она начнёт со второго гнома, то затем успеет уложить первого — и получит целых 10 минут отдыха, пока оба спят.

Формат входных данных

Первая строка входных данных содержит число n ($1 \leqslant n \leqslant 10^5$). Вторая строка содержит числа $a_1, a_2, \ldots, a_n$, третья — числа $b_1, b_2, \ldots, b_n$ ($1 \leqslant a_i, b_i \leqslant 10^9$).

Формат выходных данных

Выведите n чисел — порядок, в котором нужно укладывать гномов спать, чтобы получить хотя бы минутку отдыха. Если возможных ответов несколько, выведите любой из них. Если Белоснежке отдохнуть не удастся, выведите число -1.

Примеры

стандартный ввод	*стандартный вывод*
2 1 10 10 20	2 1
2 10 10 10 10	-1
3 1 4 1 5 3 4	2 1 3

Задача E. Почти отсортированный массив

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мебибайт

Это интерактивная задача.

У аксолотля Вики есть массив a длины n. В массиве хранится перестановка: каждое целое число от 1 до n встречается в нём ровно один раз.

Вике хотелось бы, чтобы массив был упорядочен по возрастанию: в нём стояли по порядку числа $1, 2, \ldots, n$. К сожалению, пока это немного не так. Вика посчитала *инверсии* в своём массиве — количество пар позиций (i, j), в которых элементы стоят не по порядку: $i < j$, но $a_i > a_j$. Получилось ровно n инверсий.

Аксолотль Аня — подруга Вики. Аня сказала, что знает алгоритм сортировки, работающий быстро, когда массив почти отсортирован.

Вика хочет проверить Анин алгоритм на деле. Для этого она будет держать сам массив в секрете. Аня будет задавать вопросы вида «верно ли, что $a_i > a_j$?». Если $a_i < a_j$, Вика будет просто говорить «нет». А если $a_i > a_j$, Вика будет не только говорить «да», но и сразу менять эти элементы местами.

Чтобы доказать, что алгоритм сортировки действительно быстрый, Ане нужно отсортировать Викин массив, задав не больше $5n$ вопросов. Примерьте на себя роль Ани и решите эту задачу.

Протокол взаимодействия

В первой строке входных данных записано целое число n — размер массива ($3 \leqslant n \leqslant 10\,000$). Далее решение даёт команды, а программа жюри их выполняет и выводит ответы.

Чтобы задать вопрос вида «верно ли, что $a_i > a_j$?», выведите строку в формате «`check` i j», где $1 \leqslant i < j \leqslant n$. В ответ вы получите строку с одним целым числом:

- -1, если выведенная строка некорректна;
- 0, если $a_i < a_j$;
- 1, если $a_i > a_j$, и при этом жюри поменяет a_i и a_j местами;
- 2 вместо 1, если после этого вопроса массив оказался отсортирован.

Если вы получили 0 или 1, взаимодействие продолжается. Если же вы получили -1 или 2, больше вопросы задавать нельзя. Решение считается правильным, если оно получило 2, завершилось корректно, и при этом задало не более $5n$ вопросов.

Не забывайте очищать буфер вывода сразу после печати строки. Например, в C и C++ это можно сделать командой `fflush(stdout)`, а в Python — командой `sys.stdout.flush()`.

В каждом тесте исходный массив зафиксирован заранее, то есть интерактор не адаптивный.

Пример

стандартный ввод	*стандартный вывод*	*массив*
4		3 2 4 1
	check 1 2	
1		2 3 4 1
	check 1 3	
0		2 3 4 1
	check 1 4	
1		1 3 4 2
	check 2 3	
0		1 3 4 2
	check 2 4	
1		1 2 4 3
	check 3 4	
2		1 2 3 4

Задача F. Сортировка на плоскости

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мебибайт

Это интерактивная задача.

Есть n векторов на плоскости, ненулевых и попарно неколлинеарных. Вектор с номером i идёт из начала координат в точку (x_i, y_i). Но эти координаты мы вам не скажем.

Вместо этого вы можете задавать вопросы следующего вида: «Верно ли, что векторы i и j образуют правую пару?» Формально векторы образуют правую пару, если $x_i \cdot y_j > x_j \cdot y_i$. Геометрически правая пара означает, что, если мы стоим в начале координат и смотрим на конец вектора i, то, чтобы как можно скорее повернуться к концу вектора j, следует поворачиваться против часовой стрелки.

Нужно, задавая жюри вопросы, прийти к одному из двух выводов:

1. Все векторы лежат в одной полуплоскости, граница которой проходит через начало координат. Тогда нужно отсортировать их: вывести такой набор различных индексов $i_1, i_2, \ldots, i_n$, что для любых $p < q$ векторы i_p и i_q образуют правую пару.
2. Нет такой полуплоскости, граница которой проходит через начало координат и в которой лежат все векторы. Тогда нужно доказать это: вывести такую последовательность различных индексов $i_1, i_2, \ldots, i_k$, что каждый вектор, кроме последнего, образует правую пару со следующим, а последний (i_k) образует правую пару с первым (i_1).

Протокол взаимодействия

Сначала вашей программе подаётся в отдельной строке число n — количество векторов ($1 \leqslant n \leqslant 500$). Векторы в каждом тесте зафиксированы заранее, но держатся в секрете.

Затем вы можете делать следующее:

1. Спросить у жюри: «верно ли, что векторы i и j образуют правую пару?»

   Для этого ваша программа должна вывести строку следующего вида: «`? i j`». Индексы должны быть корректными: $1 \leqslant i, j \leqslant n$.

   В ответ программа жюри выдаст в отдельной строке число: 1, если ответ «да», и 0, если ответ «нет».

   Чтобы предотвратить буферизацию вывода, после каждого выведенного вопроса следует вставить команду очистки буфера вывода: например, это может быть `fflush (stdout)` в C или C++, или `sys.stdout.flush ()` в Python.

   Вы можете задать не более 20 000 вопросов.

2. Вывести ответ. В этом случае ваша программа должна вывести две строки.

   Если все векторы лежат в одной полуплоскости, первая строка должна иметь вид «`! YES`», а во второй должны быть выведены через пробел n различных чисел от 1 до n: индексы векторов в порядке сортировки.

   Если такой полуплоскости нет, выведите в первой строке «`! NO`». В начале второй строки выведите k — число векторов в доказательстве, а затем выведите k различных чисел от 1 до n — само доказательство. Если возможных доказательств несколько, выведите любое из них.

   После вывода ответа ваша программа должна сразу корректно завершить работу.

Примеры

стандартный ввод	*стандартный вывод*
3 0 1 0	 ? 1 3 ? 3 2 ? 2 1 ! YES 3 1 2
3 1 0 0	 ? 1 2 ? 3 2 ? 1 3 ! NO 3 1 2 3

Задача G. Порядковые статистики

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мебибайт

В этой задаче требуется уметь быстро находить *k-ю порядковую статистику последовательности* — то есть элемент, который после сортировки этой последовательности по неубыванию будет стоять в ней на k-м месте.

Есть две замкнутые в кольцо ленты, на каждой из которых записана последовательность чисел: числа a_1, a_2, …, a_n на первой ленте и числа b_1, b_2, …, b_n на второй. Назовём *p-м циклическим сдвигом* положение, в котором a_1 находится над b_{p+1}, a_2 над b_{p+2}, …, a_{n-p} над b_n, a_{n-p+1} над b_1, …, a_{n-1} над b_{p-1} и a_n над b_p. Рассмотрим последовательность дробей

$$\frac{a_1}{b_{p+1}}, \frac{a_2}{b_{p+2}}, \ldots, \frac{a_{n-p}}{b_n}, \frac{a_{n-p+1}}{b_1}, \ldots, \frac{a_{n-1}}{b_{p-1}}, \frac{a_n}{b_p}$$

и обозначим k-ю порядковую статистику этой последовательности как $s_k^{(p)}$.

По данным числам на лентах, а также числу k, найдите последовательность $s_k^{(1)}$, $s_k^{(2)}$, …, $s_k^{(n)}$.

Формат входных данных

В первой строке входных данных заданы через пробел два целых числа n и k — размер лент и номер порядковой статистики ($1 \leqslant k \leqslant n \leqslant 5000$). Во второй строке записаны n чисел a_1, a_2, …, a_n — содержимое ленты с числителями ($1 \leqslant a_i \leqslant 10^9$). В третьей строке записаны n чисел b_1, b_2, …, b_n — содержимое ленты со знаменателями ($1 \leqslant b_i \leqslant 10^9$).

Формат выходных данных

В первой строке выведите n чисел через пробел — k-е порядковые статистики $s_k^{(1)}$, $s_k^{(2)}$, …, $s_k^{(n)}$. Каждое число должно быть представлено в виде несократимой дроби.

Примеры

стандартный ввод	*стандартный вывод*
3 1 1 2 3 1 2 3	1/2 1/3 1/1
2 2 4 9 4 2	9/4 9/2
3 2 2 2 2 3 4 5	1/2 1/2 1/2

Задача H. Количество инверсий

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мебибайт

Напишите программу, которая для заданного массива $A = \langle a_1, a_2, \ldots, a_n \rangle$ находит количество пар (i, j) таких, что $i < j$ и $a_i > a_j$.

Формат входных данных

Первая строка входных данных содержит натуральное число n ($1 \leqslant n \leqslant 200\,000$) — количество элементов массива. Вторая строка содержит n попарно различных элементов массива A.

Формат выходных данных

Выведите одно число — ответ на задачу.

Примеры

стандартный ввод	*стандартный вывод*
4 1 2 4 5	0
4 5 4 2 1	6

Задача I. Поиск чисел

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мебибайт

В этой задаче нужно уметь выяснять, содержится ли число в последовательности.

Формат входных данных

В первой строке заданы через пробел два целых числа n и k ($1 \leqslant n \leqslant 300\,000$, $1 \leqslant k \leqslant 300\,000$). Во второй строке задана последовательность из n отсортированных по неубыванию целых чисел $a_1, a_2, \ldots, a_n$, записанных через пробел ($1 \leqslant a_i \leqslant 10^9$). В третьей строке записаны запросы — k целых чисел $b_1, b_2, \ldots, b_k$ через пробел в порядке возрастания ($1 \leqslant b_j \leqslant 10^9$).

Формат выходных данных

Выведите k строк. В j-й строке выведите «YES», если число b_j содержится в последовательности $\{a_i\}$, и «NO» в противном случае.

Примеры

стандартный ввод	*стандартный вывод*
3 3 2 3 5 1 2 3	NO YES YES
3 4 1 2 2 1 2 4 5	YES YES NO NO

Задача J. Подсчёт на отрезках

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мебибайт

Дан массив из n целых чисел. Все числа лежат в пределах от -10^9 до 10^9 включительно.

Нужно уметь отвечать на запросы вида «*Сколько чисел имеют значения от ℓ до r?*».

Формат входных данных

В первой строке задано число n ($1 \leqslant n \leqslant 10^5$). Во второй строке заданы n целых чисел, каждое из которых лежит в пределах от -10^9 до 10^9 включительно.

Следующая строка содержит число запросов k ($1 \leqslant k \leqslant 10^5$). Каждая из следующих k строк содержит пару целых чисел ℓ и r ($-10^9 \leqslant \ell \leqslant r \leqslant 10^9$) — сами запросы.

Формат выходных данных

Выведите k чисел — ответы на запросы.

Пример

стандартный ввод	*стандартный вывод*
5 10 1 10 3 4 4 1 10 2 9 3 4 2 2	5 2 2 0

Задача K. Сумма двух

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мебибайт

Дана последовательность чисел, упорядоченная по возрастанию. Выясните, представимо ли число x в виде суммы двух чисел из этой последовательности.

Формат входных данных

В первой строке заданы через пробел два целых числа n и x ($1 \leqslant n \leqslant 100\,000$, $2 \leqslant x \leqslant 2 \cdot 10^9$).

Во второй строке задана последовательность из n целых чисел $a_1, a_2, \ldots, a_n$, записанных через пробел ($1 \leqslant a_i \leqslant 10^9$, все элементы различны, последовательность упорядочена по возрастанию).

Формат выходных данных

Выведите «YES», если число x представимо в виде $a_i + a_j$ для некоторых (не обязательно различных) индексов i и j, и «NO» в противном случае.

Примеры

стандартный ввод	*стандартный вывод*
3 4 1 2 3	YES
3 5 1 2 5	NO
4 6 1 2 4 8	YES

Задача L. Поиск целого числа

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мебибайт

Это интерактивная задача.

Жюри загадывает целое число x от 1 до 10^9. Участник задаёт до 30 вопросов вида «это число y?». На каждый вопрос жюри отвечает либо «нет, $y < x$», либо «да, $y = x$», либо «нет, $y > x$». Задача участника — получить ответ «да».

Интерактор не адаптивный: жюри загадывает число заранее и честно отвечает на все вопросы.

Протокол взаимодействия

Чтобы задать вопрос «это число y?» (y целое, $1 \leqslant y \leqslant 10^9$), выведите на отдельной строке число y. Не забудьте вывести перевод строки и очистить буфер вывода, чтобы не получить вердикт `IL` (`Idleness Limit Exceeded`).

После этого прочитайте строку с одним символом:

- «<», если $y < x$,
- «=», если $y = x$,
- «>», если $y > x$.

Если вы добились ответа «=» — сразу завершите работу программы, чтобы получить вердикт `OK` (`Accepted`).

Если вы задали 30 вопросов, но так не добились ответа «=» — сразу завершите работу программы, чтобы точно получить вердикт `WA` (`Wrong Answer`), а не какой-то ещё.

Пример

стандартный ввод	*стандартный вывод*
< > =	4 6 5

Задача M. Поиск границы

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мебибайт

Это интерактивная задача.

Жюри загадывает целое число x от 1 до 10^9. Участник задаёт до 30 вопросов вида «верно ли, что $y \geqslant x$?». На каждый вопрос жюри отвечает либо «нет, $y < x$», либо «да, $y \geqslant x$», Задача участника — найти точное значение x.

Интерактор не адаптивный: жюри загадывает число заранее и честно отвечает на все вопросы.

Протокол взаимодействия

Чтобы задать вопрос «верно ли, что $y \geqslant x$?» (y целое, $0 \leqslant y \leqslant 10^9 + 1$), выведите строку вида «`?` y». Не забудьте вывести перевод строки и очистить буфер вывода, чтобы не получить вердикт `IL` (`Idleness Limit Exceeded`). Можно задать не более 30 таких вопросов.

После этого прочитайте строку:

- «<», если $y < x$,
- «>=», если $y \geqslant x$,

Чтобы дать ответ, выведите строку вида «`!` x» ($1 \leqslant x \leqslant 10^9$) и завершите работу программы.

Пример

стандартный ввод	*стандартный вывод*
>= >=	? 6 ? 4 ? 5 ! 5

Задача N. Поиск в сортированном массиве

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мебибайт

Это интерактивная задача.

Жюри загадывает массив длины $n \leqslant 1000$, строго отсортированный по возрастанию и состоящий из целых чисел от 1 до 10^9. Жюри также выбирает целое число x — один из элементов массива — и сообщает его участнику. Участник задаёт до 10 вопросов вида «какой элемент на позиции i?». На каждый вопрос жюри раскрывает указанный элемент. Задача участника — получить ответ x.

Интерактор не адаптивный: жюри загадывает массив заранее и честно отвечает на все вопросы.

Протокол взаимодействия

В первой строке заданы два целых числа n и x — размер массива и один из его элементов ($1 \leqslant n \leqslant 1000$; $1 \leqslant x \leqslant 10^9$).

Чтобы узнать элемент на позиции i ($1 \leqslant i \leqslant n$), выведите строку вида «`? i`». Не забудьте вывести перевод строки и очистить буфер вывода, чтобы не получить вердикт `IL` (`Idleness Limit Exceeded`).

После этого прочитайте строку, содержащую одно целое число y — элемент массива на позиции i.

Если вы добились ответа x — сразу завершите работу программы, чтобы получить вердикт `OK` (`Accepted`).

Если вы задали 10 вопросов, но так не добились ответа x — сразу завершите работу программы, чтобы точно получить вердикт `WA` (`Wrong Answer`), а не какой-то ещё.

Пример

стандартный ввод	*стандартный вывод*
3 6 2 9 6	? 1 ? 3 ? 2

Задача O. Поиск максимума

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мебибайт

Это интерактивная задача.

Жюри загадывает массив длины $n \leqslant 1000$, состоящий из целых чисел от 1 до 10^9. Массив выпуклый: для $1 < i < n$ верно $a_{i-1} + a_{i+1} < 2 \cdot a_i$. Задача участника — найти максимальный элемент этого массива.

Участник задаёт до 20 вопросов вида «какой элемент на позиции i?». На каждый вопрос жюри раскрывает указанный элемент. После этого участник должен вывести значение максимального элемента в массиве.

Интерактор не адаптивный: жюри загадывает массив заранее и честно отвечает на все вопросы.

Протокол взаимодействия

В первой строке задано целое число n — размер массива ($1 \leqslant n \leqslant 1000$).

Чтобы задать вопрос «какой элемент на позиции i?» ($1 \leqslant i \leqslant n$), выведите строку вида «`? i`». Не забудьте вывести перевод строки и очистить буфер вывода, чтобы не получить вердикт `IL` (`Idleness Limit Exceeded`). Можно задать не более 20 таких вопросов.

После этого прочитайте строку, содержащую одно целое число — элемент массива на позиции i.

Чтобы дать ответ, выведите строку вида «`! m`» ($1 \leqslant m \leqslant 10^9$) и завершите работу программы.

Пример

В примере загадан массив $(2, 9, 6)$.

стандартный ввод	*стандартный вывод*
3 2 9 6	? 1 ? 2 ? 3 ! 9

Задача P. Поиск суммы двух

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мебибайт

Это интерактивная задача.

Жюри загадывает массив длины $n \leqslant 1000$, строго отсортированный по возрастанию и состоящий из целых чисел от 1 до 10^9. Жюри также выбирает два элемента массива — возможно, один и тот же два раза — и сообщает участнику их сумму s. Задача участника — найти два элемента с такой суммой.

Участник задаёт до 1000 вопросов вида «верно ли, что сумма элементов на позициях i и j равна s?». На каждый вопрос жюри отвечает либо «нет, их сумма меньше s», либо «да, их сумма равна s», либо «нет, их сумма больше s». Задача участника — получить ответ «да».

Интерактор не адаптивный: жюри загадывает массив заранее и честно отвечает на все вопросы.

Протокол взаимодействия

В первой строке заданы два целых числа n и s — размер массива и сумма двух его элементов ($1 \leqslant n \leqslant 1000$; $2 \leqslant s \leqslant 2 \cdot 10^9$).

Чтобы задать вопрос «верно ли, что сумма элементов на позициях i и j равна s?» ($1 \leqslant i, j \leqslant n$), выведите строку вида «`? i j`». Не забудьте вывести перевод строки и очистить буфер вывода, чтобы не получить вердикт `IL` (`Idleness Limit Exceeded`).

После этого прочитайте строку с одним символом:

- «`<`», если их сумма меньше s,
- «`=`», если их сумма равна s,
- «`>`», если их сумма больше s.

Если вы добились ответа «`=`» — сразу завершите работу программы, чтобы получить вердикт `OK` (`Accepted`).

Если вы задали 1000 вопросов, но так не добились ответа «`=`» — сразу завершите работу программы, чтобы точно получить вердикт `WA` (`Wrong Answer`), а не какой-то ещё.

Пример

В примере загадан массив $(2, 6, 9)$.

стандартный ввод	*стандартный вывод*

`3 11`
 	
`<`
 	
`>`
 	
`=`
	
	`? 1 2`
	
	`? 2 3`
	
	`? 3 1`

Задача Q. Соседи

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мебибайт

На прямой отмечены n целых точек с координатами $x_1, x_2, \ldots, x_n$, заданными слева направо.

Добавим на прямую ещё n целых точек так, чтобы максимальное из расстояний между соседними точками оказалось как можно меньше. Чему равно это расстояние?

Формат входных данных

В первой строке записано целое число n: количество точек ($2 \leqslant n \leqslant 10\,000$). В следующей строке записаны n целых чисел $x_1, x_2, \ldots, x_n$: координаты точек ($1 \leqslant x_i \leqslant 10^{18}$, $x_i < x_{i+1}$).

Формат выходных данных

Выведите одно целое число: насколько малым может быть максимальное из расстояний между соседями после добавления ещё n целых точек.

Пример

стандартный ввод	*стандартный вывод*	*пояснение*
3 1 5 12	3	1 3 5 8 10 12 2 2 3 2 2

Задача R. Blue Water

Имя входного файла:	*стандартный ввод*
Имя выходного файла:	*стандартный вывод*
Ограничение по времени:	1 секунда
Ограничение по памяти:	512 мебибайт

У вас на вечеринке в честь успешного открытия компании с брендом «Blue Water» присутствуют n друзей. Известно, что все пьют воду с одной и той же скоростью, 1 галлон в секунду. Имеется резервуар с водой объемом c галлонов. Изначально у всех есть по стакану, каждый будет пользоваться своим стаканом до конца вечеринки. Стаканы бывают разной вместимости, все пронумерованы от 0 до $n-1$, и изначально все они пусты.

Во время вечеринки все пьют воду из своих стаканов, и если у кого-то стакан становится пустым, то он идёт доливать в него воду. Доливание воды занимает одну секунду. Если стакан не пуст, то доливать в него воду не разрешается. Если несколько человек одновременно идут наполнять стаканы, то они делают это в одну и ту же секунду; если при этом воды на всех не хватает, сперва наполняет свой стакан тот, у кого номер стакана меньше. Если в резервуаре осталось воды меньше, чем вмещает стакан, эту воду всё равно можно в него налить.

Зная вместимости всех стаканов, выберите себе тот, который гарантирует вам максимальное количество суммарно выпитой воды.

Формат входных данных

В первой строке задано через пробел два целых числа: количество стаканов n и объём резервуара c ($1 \leqslant n \leqslant 50$, $0 \leqslant c \leqslant 2\,000\,000\,000$). Во второй строке задано через пробел n целых чисел a_i: вместимости стаканов ($1 \leqslant a_i \leqslant 10\,000$).

Формат выходных данных

Выведите номер стакана, который нужно выбрать, чтобы максимизировать суммарное количество выпитой воды. Если ответов несколько, выведите наименьший. Стаканы нумеруются с нуля в порядке, в котором они заданы.

Примеры

стандартный ввод	*стандартный вывод*
2 100 1 20	1
2 101 8 10	0