

Задача А. Ярый коллекционер бабочек

Имя входного файла: `collect.in`
Имя выходного файла: `collect.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 256 мегабайт

Как известно, Андрей Сергеевич — ярый коллекционер бабочек. Он имеет огромную коллекцию, экспонаты которой собраны со всего мира. Будем считать, что в мире существует 2 000 000 000 видов бабочек.

Чтобы не запутаться, Андрей Сергеевич присвоил каждому виду уникальный номер. Нумерация видов бабочек начинается с единицы.

Теперь он хочет знать, есть ли бабочка с видом K в его коллекции, или же её придётся добывать, затрачивая уйму сил и денег.

Формат входного файла

В первой строке входного файла содержится единственное число N ($1 \leq N \leq 100\,000$) — количество видов бабочек в коллекции Андрея Сергеевича.

В следующей строке через пробел находятся N упорядоченных по возрастанию чисел — номера видов бабочек в коллекции.

Все виды бабочки в коллекции имеют различные номера.

В третьей строке файла записано число M ($1 \leq M \leq 100\,000$), количество видов бабочек, про которых Андрей Сергеевич хочет узнать, есть ли они у него в коллекции или же нет. В последней строке входного файла содержатся через пробел M чисел — номера видов бабочек, наличие которых необходимо проверить.

Формат выходного файла

Выходной файл должен содержать M строчек. Для каждого запроса выведите число “YES”, если бабочка с данным номером содержится в коллекции, и “NO” — в противном случае.

Примеры

<code>collect.in</code>	<code>collect.out</code>
7	NO
10 47 50 63 89 90 99	NO
4	YES
84 33 10 82	NO

Задача В. Коровы - в стойла

Имя входного файла: `cows.in`
Имя выходного файла: `cows.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

На прямой расположены стойла, в которые необходимо расставить коров так, чтобы минимальное расстояние между коровами было как можно больше.

Формат входного файла

В первой строке вводятся числа N ($3 \leq N \leq 10\,000$) — количество стойл и K ($2 \leq K < N$) — количество коров. Во второй строке задаются N натуральных чисел в порядке возрастания — координаты стойл (координаты не превосходят 10^9).

Формат выходного файла

Выведите одно число — наибольшее возможное допустимое расстояние.

Примеры

<code>cows.in</code>	<code>cows.out</code>
5 3 1 2 3 100 1000	99

Задача С. Поляна Дров

Имя входного файла: `forest.in`
Имя выходного файла: `forest.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Маленький мальчик Ферма́ живет в деревне. Наступают холодные времена, поэтому бабушка попросила мальчика сходить в лес, чтобы собрать дров. В лесу около деревни, в которой живет Ферма, находится волшебная Поляна Дров, на которой всегда лежат дрова, и никогда не кончаются. Естественно, Ферма должен пойти именно туда.

Единственная проблема заключается в том, что идти до Поляны не очень близко, тем более что скорость передвижения по лесу намного меньше, чем скорость передвижения по полю, в котором находится деревня.

- Деревня находится в точке с координатами $(0, 1)$.
- Поляна находится в точке с координатами $(1, 0)$.
- Граница между лесом и полем — горизонтальная прямая $y = a$, где a — некоторое число $(0 \leq a \leq 1)$.
- Скорость передвижения по полю составляет V_p , скорость передвижения по лесу — V_f . Вдоль границы можно двигаться как по лесу, так и по полю.

Найдите точку, в которой мальчик Ферма должен войти в лес, чтобы дойти до Поляны Дров как можно быстрее.

Формат входного файла

В первой строке входного файла содержатся два положительных целых числа — V_p и V_f ($1 \leq V_p, V_f \leq 10^5$). Во второй строке содержится единственное вещественное число — координата по оси Oy границы между лесом и полем a ($0 \leq a \leq 1$)

Формат выходного файла

В единственной строке выходного файла выведите вещественное число с точностью не менее 8 знаков после запятой — координата по оси Ox точки, в которой мальчик Ферма должен войти в лес.

Примеры

forest.in	forest.out
5 3 0.4	0.783310604
5 5 0.5	0.500000000

Задача D. Корень кубического уравнения

Имя входного файла: `cubroot.in`
Имя выходного файла: `cubroot.out`
Ограничение по времени: 1 секунда
Ограничение по памяти: 256 мегабайт

Дано кубическое уравнение $ax^3 + bx^2 + cx + d = 0$ ($a \neq 0$). Известно, что у этого уравнения есть ровно один корень. Требуется его найти.

Формат входного файла

Во входном файле через пробел записаны четыре целых числа: $-1000 \leq a, b, c, d \leq 1000$.

Формат выходного файла

Выведите единственный корень уравнения с точностью не менее 6 знаков после десятичной точки.

Пример

<code>cubroot.in</code>	<code>cubroot.out</code>
1 -3 3 -1	1
-1 -6 -12 -7	-1.000000

Задача Е. Поезда

Имя входного файла: `trains.in`
Имя выходного файла: `trains.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

В связи с участвовавшим числом аварий на железнодорожной ветке Кострома–Судиславль, руководство железной дороги решило изменить график движения поездов. Тщательный анализ состояния железнодорожного полотна показал, что оптимальным является следующий график движения поездов с учетом остановок на станциях: сначала поезд идет на протяжении T_1 минут со скоростью V_1 метров в минуту, затем T_2 минут со скоростью V_2 метров в минуту, ..., наконец, T_N минут со скоростью V_N метров в минуту. В течение некоторых интервалов поезд может стоять (скорость равна 0).

По действующей инструкции обеспечения безопасности движения поездов расстояние между локомотивами двух следующих друг за другом поездов должно быть не менее L метров. Определите минимально допустимый интервал в минутах между отправлениями поездов, позволяющий им двигаться по этому графику без опасного сближения.

Формат входного файла

В первых двух строках входного файла содержится два натуральных числа, задающие минимально допустимое расстояние L и количество участков пути N ($100 \leq L \leq 10\,000$, $1 \leq N \leq 1000$). Далее следует N пар целых чисел T_i и V_i , задающих график движения поездов ($1 \leq T_i \leq 1000$, $0 \leq V_i \leq 1000$).

Формат выходного файла

В выходной файл необходимо вывести искомый интервал между отправлениями поездов в минутах, не менее чем с тремя верными знаками после десятичной точки.

Пример

<code>trains.in</code>	<code>trains.out</code>
1000 4 10 0 30 80 15 0 20 100	27.500