

**Задача А. Коровы**

Имя входного файла: cow.in  
 Имя выходного файла: cow.out  
 Ограничение по времени: 1 секунда  
 Ограничение по памяти: 64 мегабайта

Как известно, коровы, пасущиеся на пастбище, должны быть привязаны как можно дальше друг от друга, чтобы каждой из них досталось побольше травы. У фермера Джона есть  $k$  коров, а на пастбище в один ряд вбито  $n$  кольшков ( $n > k$ ). Помогите фермеру привязать коров к некоторым кольшкам так, чтобы минимальное расстояние между коровами было как можно больше.

**Формат входного файла**

В первой строке вводятся числа  $n$  ( $2 < n < 10001$ ) — количество кольшков и  $k$  ( $1 < k < n$ ) — количество коров. Во второй строке задаются  $n$  натуральных чисел в порядке возрастания — координаты кольшков на прямой (координаты не превосходят  $10^9$ ).

**Формат выходного файла**

Выведите одно число — наибольшее возможное допустимое расстояние между коровами.

**Пример**

cow.in	cow.out
5 3 1 2 3 100 1000	99

**Задача В. Для любителей статистики**

Имя входного файла: queries.in  
 Имя выходного файла: queries.out  
 Ограничение по времени: 1 секунда  
 Ограничение по памяти: 64 мегабайта

Вы никогда не задумывались над тем, сколько человек за год перевозят трамваи города с десятиллионным населением, в котором каждый третий житель пользуется трамваем по два раза в день?

Предположим, что на планете Земля  $n$  городов, в которых есть трамвай. Любители статистики подсчитали для каждого из этих городов, сколько человек перевезено трамваями этого города за последний год. Из этих данных была составлена таблица, в которой города были отсортированы по алфавиту. Позже выяснилось, что для статистики названия городов несущественны, и тогда их просто заменили числами от 1 до  $n$ . Поисковая система, работающая с этими данными, должна уметь быстро отвечать на вопрос, есть ли среди городов с номерами от  $l$  до  $r$  такой, что за год трамвай этого города перевезли ровно  $x$  человек. Вам предстоит реализовать этот модуль системы.

**Формат входного файла**

В первой строке дано целое число  $n$ ,  $0 < n < 70\,000$ . В следующей строке приведены статистические данные в виде списка целых чисел через пробел,  $i$ -е число в этом списке — количество человек, перевезенных за год трамваями  $i$ -го города. Все числа в списке положительны и не превосходят  $10^9 - 1$ . В третьей строке дано количество запросов  $q$ ,  $0 < q < 70\,000$ . В следующих  $q$  строках перечислены запросы. Каждый запрос — это тройка целых чисел  $l$ ,  $r$  и  $x$ , записанных через пробел ( $1 \leq l \leq r \leq n$ ,  $0 < x < 10^9$ ).

**Формат выходного файла**

Выведите строку длины  $q$ , в которой  $i$ -й символ равен 1, если ответ на  $i$ -й запрос утвердителен, и 0 в противном случае.

**Пример**

queries.in	queries.out
5 123 666 314 666 434 5 1 5 314 1 5 578 2 4 666 4 4 713 1 1 123	10101

**Задача С. Поляна Дров**

Имя входного файла: forest.in  
 Имя выходного файла: forest.out  
 Ограничение по времени: 1 секунда  
 Ограничение по памяти: 64 мегабайта

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.

**Пример**

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

Указание. Для обеспечения требуемой точности используйте тип данных `extended` в Паскале и `long double` в Си.

**Задача D. Корень кубического уравнения**

Имя входного файла: cubroot.in  
 Имя выходного файла: cubroot.out  
 Ограничение по времени: 1 секунда  
 Ограничение по памяти: 64 мегабайта

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

cubeoot.in	cubeoot.out
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: 1 секунда  
Ограничение по памяти: 64 мегабайта

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

trains.in	trains.out
1000	27.500
4	
10 0	
30 80	
15 0	
20 100	