1

Сложность двоичного поиска

2

Квадратный корень и квадратный квадрат

3

Приближенный двоичный поиск

4

Дипломы

5

Субботник

6

Космическое поселение

7

Коровы - в стойла

8

Вырубка леса

9

Детский праздник

Задача 1.

Вводная

**Сложность двоичного поиска**

Вася загадал число от 1 до N. За какое наименьшее количество вопросов (на которые Вася отвечает "да" или "нет") Петя может угадать Васино число?

ВХОДНЫЕ ДАННЫЕ

Вводится одно число N

ВЫХОДНЫЕ ДАННЫЕ

Выведите наименьшее количество вопросов, которого гарантированно хватит Пете, чтобы угадать Васино число.

|  |  |
| --- | --- |
| ввод | вывод |
| 5 | 3 |

Решение задачи

Как мы помним, сложность бинарного поиска это двоичный логарифм от числа N. Можно просто взять функцию log2 от N и округлить вверх.

Задача 2.

Лёгкая

**Квадратный корень и квадратный квадрат**

Найдите такое число x, что x2+x=C, с точностью не менее 6 знаков после точки.

ВХОДНЫЕ ДАННЫЕ

В единственной строке содержится вещественное число 1.0≤C≤1010

ВЫХОДНЫЕ ДАННЫЕ

Выведите одно число — искомый x.

|  |  |
| --- | --- |
| ввод | вывод |
| 2.0000000000 | 1.000000000 |
| 18.0000000000 | 4.000000000 |

Начало формы

Задача решена верно

OK. Your score is = 100, 30/30 tests passed

Решение задачи

Функция возрастающая, поэтому сделаем бинарный поиск. Не забудем, что требуется точность 6 знаков после запятой, учтем это.

Конец формы

Задача 3.

Средняя

**Приближенный двоичный поиск**

Реализуйте алгоритм приближенного бинарного поиска.

Входные данные

В первой строке входных данных содержатся числа N и K (0 < N, K < 100001). Во второй строке задаются N чисел первого массива, отсортированного по неубыванию, а в третьей строке – K чисел второго массива. Каждое число в обоих массивах по модулю не превосходит 2⋅109.

Выходные данные

Для каждого из K чисел выведите в отдельную строку число из первого массива, наиболее близкое к данному. Если таких несколько, выведите меньшее из них.

Примеры

входные данные

5 5

1 3 5 7 9

2 4 8 1 6

выходные данные

1

3

7

1

5

Задача 4.

Средняя

**Дипломы**

**Ограничение по времени работы программы: 1 секунда**

Когда Петя учился в школе, он часто участвовал в олимпиадах по информатике, математике и физике. Так как он был достаточно способным мальчиком и усердно учился, то на многих из этих олимпиад он получал дипломы. К окончанию школы у него накопилось n дипломов, причём, как оказалось, все они имели одинаковые размеры: w в ширину и h в высоту.

Сейчас Петя учится в одном из лучших российских университетов и живёт в общежитии со своими одногруппниками. Он решил украсить свою комнату, повесив на одну из стен свои дипломы за школьные олимпиады. Так как к бетонной стене прикрепить дипломы достаточно трудно, то он решил купить специальную доску из пробкового дерева, чтобы прикрепить её к стене, а к ней — дипломы. Для того чтобы эта конструкция выглядела более красиво, Петя хочет, чтобы доска была квадратной и занимала как можно меньше места на стене. Каждый диплом должен быть размещён строго в прямоугольнике размером w×h. Дипломы запрещается поворачивать на 90 градусов. Прямоугольники, соответствующие различным дипломам, не должны иметь общих внутренних точек.

Требуется написать программу, которая вычислит минимальный размер стороны доски, которая потребуется Пете для размещения всех своих дипломов.

ВХОДНЫЕ ДАННЫЕ

Программа получает на вход три целых числа (в одной строке через пробел): w, h, n (1⩽w,h,n⩽109).

ВЫХОДНЫЕ ДАННЫЕ

Необходимо вывести ответ на поставленную задачу.

ПРИМЕР

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 2 3 10 | 9 |

Начало формы

Задача решена верно

OK. Your score is = 100, 21/21 tests passed

Решение задачи

Будем решать задачу двоичным поиском по ответу. Пусть L — размер стороны доски. Подсчитаем количество дипломом, которое можно разместить на доске размером L×L. Оно равно (L//w) \* (L//h). Если это число больше или равно n, значит, доска размером L подходит.

В качестве начальной левой и правой границы для двоичного поиска нужно выбрать такие значения L, которые заведомо не подходят и заведомо подходят. Возьмем left = 0 в качестве левой границы (на доске размером 0 нельзя поместить ни одного диплома) и right = n \* max(w, h) в качестве правой границы - на доске такого размера заведомо можно разместить n дипломов в один ряд.

После окончания двоичного поиска необходимо вывести значение правой границы.

Конец формы

Задача 5.

Сложная

**Субботник**

**Ограничение по времени работы: 1 секунда**

Для переноски бревна на субботнике нужно выбрать двоих человек, разность роста которых не превышает M (если рост различается более, чем на M, то бревно носить неудобно). На субботник пришло N человек, известен рост каждого человека. Определите количество способов выбрать из них двоих человек с соблюдением данного требования.

ВХОДНЫЕ ДАННЫЕ

Программа получает на вход числа N и M в первой строке (1⩽N⩽105, 0⩽M⩽109), во второй строке записано N натуральных чисел через пробел — значения роста (натуральные числа, не превосходящие 109).

ВЫХОДНЫЕ ДАННЫЕ

Программа должна вывести одно целое число — искомое количество способов.

ПРИМЕР

|  |  |
| --- | --- |
| **ввод** | **вывод** |
| 4 2 1 4 3 5 | 4 |

Задача 6.

Лёгкая

**Космическое поселение**

Для освоения Марса требуется построить исследовательскую базу. База должна состоять из n одинаковых модулей, каждый из которых представляет собой прямоугольник. Каждый модуль представляет собой жилой отсек, который имеет форму прямоугольника размером a×b метров. Для повышения надежности модулей инженеры могут добавить вокруг каждого модуля слой дополнительной защиты. Толщина этого слоя должна составлять целое число метров, и все модули должны иметь одинаковую толщину дополнительной защиты. Модуль с защитой, толщина которой равна d метрам, будет иметь форму прямоугольника размером (a+2d)×(b+2d) метров. Все модули должны быть расположены на заранее подготовленном прямоугольном поле размером w×h метров. При этом они должны быть организованы в виде регулярной сетки: их стороны должны быть параллельны сторонам поля, и модули должны быть ориентированы одинаково. Требуется написать программу, которая по заданным количеству и размеру модулей, а также размеру поля для их размещения, определяет максимальную толщину слоя дополнительной защиты, который можно добавить к каждому модулю.

## **Входные данные**

Входной файл содержит пять разделенных пробелами целых чисел: n,a,b,w и h(1≤n,a,b,w,h≤1018). Гарантируется, что без дополнительной защиты все модули можно разместить в поселении описанным образом.

## **Выходные данные**

Выходной файл должен содержать одно целое число: максимальную возможную толщину дополнительной защиты. Если дополнительную защиту установить не удастся, требуется вывести число 0.

|  |  |
| --- | --- |
| ввод | вывод |
| 11 2 3 21 25 | 2 |
| 1 5 5 6 6 | 0 |

Начало формы

Задача решена верно

OK. Your score is = 100, 44/44 tests passed

Решение задачи

Зависимость количества отсеков от ширины защиты — монотонно убывающая функция. Сделаем по ней бинпоиск.

Конец формы

Задача 7.

Сложная

**Коровы - в стойла**

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

ВХОДНЫЕ ДАННЫЕ

В первой строке вводятся числа N  (2 < N < 10001) – количество стойл и K  (1 < K < N ) – количество коров. Во второй строке задаются N натуральных чисел в порядке возрастания – координаты стойл (координаты не превосходят 109)

ВЫХОДНЫЕ ДАННЫЕ

Выведите одно число – наибольшее возможное допустимое расстояние.

|  |  |
| --- | --- |
| ввод | вывод |
| 6 3  2 5 7 11 15 20 | 9 |

Начало формы

Задача решена верно

OK. Your score is = 100, 16/16 tests passed

Решение задачи

Нужно придумать функцию F, которая проверяет, сколько коров с переданным ей допустимым расстоянием можно расставить, и возвращает true, если можно расставить всех. Построим следующий график. Ось абсцисс целочисленная, ось ординат булевская. Отложим по оси абсцисс все допустимые расстояния, меньше которых между коровами быть не должно (от нуля до бесконечности), а по оси ординат — можем ли мы расставить всех коров с таким минимальным допустимым расстоянием. Легко видеть, что функция эта сначала принимает только значения true, а с некоторого момента — только значения false. По ней и будем делать бинпоиск. Нас интересует, где достигается крайнее правое значение true.

Берём отрезок [A; B] на оси абсцисс, покрывающий все возможные ответы на задачу. Для его середины M вызываем функцию F. Если она даёт true, то наш ответ лежит на отрезке [M; B], иначе — на полуинтервале [A; M). Рекурсивно вызываемся, творим бинпоиск и так далее.

Задача 8.

Сложная

**Вырубка леса**

Фермер Николай нанял двух лесорубов: Дмитрия и Федора, чтобы вырубить лес, на месте которого должно быть кукурузное поле. В лесу растут X деревьев.  
  
Дмитрий срубает по A деревьев в день, но каждый K-й день он отдыхает и не срубает ни одного дерева. Таким образом, Дмитрий отдыхает в K-й, 2K-й, 3K-й день, и т.д.  
  
Федор срубает по B деревьев в день, но каждый M-й день он отдыхает и не срубает ни одного дерева. Таким образом, Федор отдыхает в M-й, 2M-й, 3M-й день, и т.д.  
  
Лесорубы работают параллельно и, таким образом, в дни, когда никто из них не отдыхает, они срубают A + B деревьев, в дни, когда отдыхает только Федор — A деревьев, а в дни, когда отдыхает только Дмитрий — B деревьев. В дни, когда оба лесоруба отдыхают, ни одно дерево не срубается.  
  
Фермер Николай хочет понять, за сколько дней лесорубы срубят все деревья, и он сможет засеять кукурузное поле.   
  
Требуется написать программу, которая по заданным целым числам A, K, B, M и X определяет, за сколько дней все деревья в лесу будут вырублены.

ВХОДНЫЕ ДАННЫЕ

Входной файл содержит пять целых чисел, разделенных пробелами: A, K, B, M и X (1 ≤ A, B ≤ 109 , 2 ≤ K, M ≤ 1018, 1 ≤ X ≤ 1018).

ВЫХОДНЫЕ ДАННЫЕ

Выходной файл должен содержать одно целое число — искомое количество дней.

|  |  |
| --- | --- |
| ввод | вывод |
| 2 4 3 3 25 | 7 |

Конец формы