

Задача А. Ярый коллекционер бабочек

Имя входного файла: `collect.in`
Имя выходного файла: `collect.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 64 мегабайта

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 входных данных

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 бабочек 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 выходных данных

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

Примеры

<code>collect.in</code>	<code>collect.out</code>
7	YES
23 27 36 45 54 72 98	YES
4	YES
54 36 23 45	YES

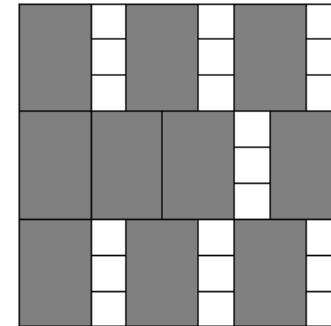
Задача В. Дипломы

Имя входного файла: `diplomas.in`
Имя выходного файла: `diplomas.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

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 на 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.

Формат входных данных

Входной файл содержит три целых числа: w , h , n ($1 \leq w, h, n \leq 10^9$).

Формат выходных данных

В выходной файл необходимо вывести ответ на поставленную задачу.

Примеры

<code>diplomas.in</code>	<code>diplomas.out</code>
2 3 10	9

Задача С. Очень легкая задача

Имя входного файла: `easy.in`
Имя выходного файла: `easy.out`
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Сегодня утром жюри решило добавить в вариант олимпиады еще одну, Очень Легкую Задачу. Ответственный секретарь Оргкомитета напечатал ее условие в одном экземпляре,

и теперь ему нужно до начала олимпиады успеть сделать еще N копий. В его распоряжении имеются два ксерокса, один из которых копирует лист за x секунд, а другой — за y . (Разрешается использовать как один ксерокс, так и оба одновременно. Можно копировать не только с оригинала, но и с копии.) Помогите ему выяснить, какое минимальное время для этого потребуется.

Формат входных данных

На вход программы поступают три натуральных числа N , x и y , разделенные пробелом ($1 \leq N \leq 2 \cdot 10^8$, $1 \leq x, y \leq 10$).

Формат выходных данных

Выведите одно число — минимальное время в секундах, необходимое для получения N копий.

Примеры

easy.in	easy.out
4 1 1	3
5 1 2	4

Задача D. Мороженое

Имя входного файла: ice-cream.in
Имя выходного файла: ice-cream.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Вдоль моря узкой полоской тянется пляж. В некоторых точках пляжа расположены ларьки с мороженым. В один прекрасный день не все мороженщики вышли на работу. Распределите мороженщиков по ларькам так, чтобы минимальное расстояние между мороженщиками было как можно больше. Так они будут меньше мешать друг другу.

Формат входных данных

В первой строке вводятся числа N ($2 < N < 10001$) — количество ларьков и K ($1 < K < N$) — количество мороженщиков, вышедших на работу. Во второй строке задаются N натуральных чисел в порядке возрастания — координаты ларьков (координаты не превосходят 10^9).

Формат выходных данных

Выведите одно число — минимальное расстояние между соседними ларьками в оптимальной расстановке.

Примеры

ice-cream.in	ice-cream.out
5 3 1 2 3 100 1000	99

Задача E. Заполнение

Имя входного файла: snake.in
Имя выходного файла: snake.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

Заполните массив $N \times N$ элементов змейкой (см. пример).

Формат входных данных

Во входном файле записано единственное число $N \leq 500$.

Формат выходных данных

В выходной файл выведите массив из $N \times N$, заполненный "змейкой".

Примеры

snake.in	snake.out
3	1 2 3 6 5 4 7 8 9

Задача F. Диагональки

Имя входного файла: diagonal.in
Имя выходного файла: diagonal.out
Ограничение по времени: 2 секунды
Ограничение по памяти: 64 мегабайта

В квадратной таблице $N \times N$ подсчитать суммы чисел, стоящих на диагоналях.

Формат входных данных

Вводится число N ($1 \leq N \leq 100$), а затем матрица $N \times N$. Элементы матрицы - числа из диапазона *integer*.

Формат выходных данных

Вывести сначала сумму чисел на главной, а затем - на побочной диагонали.

Примеры

diagonal.in	diagonal.out
1 451	451 451
4 134 475 30 424 303 151 419 235 248 166 90 42 318 237 184 36	411 1327