

Задача А. Генераторы списков

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

На входе программа получает список целых чисел s . Ваша задача — вывести следующие списки по одному на строку:

1. Список, состоящий из квадратов элементов s .
2. Список, состоящий из остатков от деления на 11 элементов s .
3. Список, состоящий только из чётных элементов s .
4. Список, состоящий только из элементов s с нечётным количеством цифр.
5. Список, состоящий только из двузначных элементов s , записанных 2 раза подряд.
6. Список, состоящий из элементов s , стоящих на позициях, не крытых 3.

Формат входных данных

В единственной строке записаны числа, разделённые пробелами.

Формат выходных данных

В каждой из шести строк выведите соответствующий список в стандартном формате python'a.

Примеры

<code>list.in</code>	<code>list.out</code>
8 5 15 101 42 1	[64, 25, 225, 10201, 1764, 1] [8, 5, 4, 2, 9, 1] [8, 42] [8, 5, 101, 1] [1515, 4242] [5, 15, 42, 1]

Замечание

Для вывода каждого из списков необходимо использовать единственную строку с генератором.
Например:

```
print([x for x in s])
```

Задача В. 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 С. Мутанты

Имя входного файла: `mutants.in`
Имя выходного файла: `mutants.out`
Ограничение по времени: 4 секунды
Ограничение по памяти: 128 мегабайта

Уже долгое время в Институте Искусств, Мутантов и Информационных Технологий разводят милых разноцветных зверюшек. Для удобства каждый цвет обозначен своим номером, всего цветов не более 10^9 . В один из прекрасных дней в питомнике случилось чудо: все зверюшки выстроились в ряд в порядке возрастания цветов. Пользуясь случаем, лаборанты решили посчитать, сколько зверюшек разных цветов живет в питомнике, и, по закону жанра, попросили вас написать программу, которая поможет им в решении этой нелегкой задачи.

Формат входных данных

В первой строке входного файла содержится единственное число N ($0 \leq N \leq 10^5$) — количество зверюшек в Институте. В следующей строке находятся N упорядоченных по неубыванию неотрицательных целых чисел, не превосходящих 10^9 и разделенных пробелами — их цвета. В третьей строке файла записано число M ($1 \leq M \leq 100\,000$) — количество запросов вашей программе, в следующей строке через пробел записаны M целых неотрицательных чисел (не превышающих $10^9 + 1$).

Формат выходных данных

Выходной файл должен содержать M строчек. Для каждого запроса выведите число зверюшек заданного цвета в питомнике.

Примеры

<code>mutants.in</code>	<code>mutants.out</code>
10	1
1 1 3 3 5 7 9 18 18 57	2
5	1
57 3 9 1 179	2
	0

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 ($2 < N < 10\,001$) — 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

<code>ice-cream.in</code>	<code>ice-cream.out</code>
5 3 1 2 3 100 1000	99