



В. Приоритизация заданий (10 баллов)

ограничение по времени на тест: 3 секунды

ограничение по памяти на тест: 256 мегабайт

ввод: стандартный ввод

вывод: стандартный вывод

Вам надо разработать подсистему простой приоритизации заданий.

В системе есть n заданий, i -е задание имеет важность r_i ($1 \leq r_i \leq 10^9$).

Каждому заданию надо назначить пiсе-значение — целое число от 1 и более. Чем ниже это число, тем выше приоритет у задания.

Пусть максимальное значение важности среди всех заданий равно max_r , тогда пiсе-значение равное 1 получают все задания с важностью max_r и те, важность которых на 1 меньше.

После этого удалим из рассмотрения все задания с назначенным пiсе-значением и назначим пiсе-значение 2, повторив описанную выше процедуру.

После этого удалим из рассмотрения все задания с назначенным пiсе-значением и назначим пiсе-значение 3, повторив описанную выше процедуру.

И так далее до тех пор, пока есть хотя бы одно задание без пiсе-значения.

Для каждого задания выведите его пiсе-значение.

Входные данные

В первой строке записано целое число t ($1 \leq t \leq 10^4$) — количество наборов входных данных в тесте.

Далее следуют описания наборов, каждый набор задается двумя строками.

Первая из них содержит целое число n ($1 \leq n \leq 2 \cdot 10^5$) — количество заданий.

Вторая строка содержит n целых чисел r_1, r_2, \dots, r_n ($1 \leq r_i \leq 10^9$) — важности заданий.

Гарантируется, что сумма значений n в тесте не превосходит $2 \cdot 10^5$.

Неполные решения этой задачи могут быть оценены частичным баллом.

Выходные данные

Для каждого набора входных данных выведите одну строку, которая содержит n чисел v_1, v_2, \dots, v_n , где v_i — пiсе-значение, которое получило задание i .

Пример

входные данные	Скопировать
5 7 4 95 101 2 100 101 96 1 1000000000 10 1 2 3 4 5 6 7 8 9 10 13 3 1 4 1 5 9 2 6 5 3 5 8 9 2 1000000000 1	
выходные данные	Скопировать
3 2 1 4 1 1 2 1 5 5 4 4 3 3 2 2 1 1 3 4 3 4 2 1 4 2 2 3 2 1 1 1 2	