

## Задача 10. Маленькая сортирующая машина

Источник:	космической сложности
Имя входного файла:	<code>input.txt</code>
Имя выходного файла:	<code>output.txt</code>
Ограничение по времени:	1 секунда*
Ограничение по памяти:	разумное

Вам предлагается испытать себя в оптимизации сортировки массива маленького размера.

В каждом тесте имеется массив из  $N$  элементов, у каждого элемента есть ключ и значение.

И ключ, и значение являются беззнаковыми четырёхбайтовыми целыми числами.

Изначально во входном файле заданы только значения всех элементов массива. Далее нужно выполнить  $R$  раундов. На каждом раунде нужно:

1. Сгенерировать и записать  $N$  случайных чисел в ключи элементов массива.
2. Отсортировать элементы массива в порядке возрастания ключа.

В результате каждого раунда значения в массиве переставляются в некотором порядке, который зависит от генератора псевдослучайных чисел.

В данной задаче нужно использовать генератор псевдослучайных чисел xorwow. Исходный код этого генератора:

```
uint32_t xorwow(uint32_t state[5]) {
    uint32_t s, t = state[3];
    t ^= t >> 2;
    t ^= t << 1;
    state[3] = state[2]; state[2] = state[1]; state[1] = s = state[0];
    t ^= s;
    t ^= s << 4;
    state[0] = t;
    return t + (state[4] += 362437);
}
```

Начальное состояние генератора (пять чисел в `state`) задаётся в каждом тесте. В начале каждого раунда ключи генерируются для элементов в их текущем порядке, причём старшие два бита отбрасываются:

```
for (int i = 0; i < n; i++)
    elements[i].key = xorwow(state) & 0x3fffffff;
```

В конце теста нужно вывести значения всех элементов массива в их финальном порядке. Обратите внимание, что процесс полностью детерминирован, и только один финальный порядок является правильным.

### Формат входных данных

В первой строке записано целое число  $Q$  — сколько тестов записано в файле ( $1 \leq Q \leq 1000$ ). Далее описано  $Q$  тестов.

Описание теста начинается со строки с двумя целыми числами:  $N$  — размер массива и  $R$  — сколько раундов сортировки нужно выполнить ( $1 \leq N \leq 64$ ,  $0 \leq R$ ). Во второй строке теста записано пять шестнадцатеричных чисел, по восемь цифр в каждом — начальное содержимое массива `state` генератора xorwow. В третьей строке записано  $N$  целых чисел — значения элементов массива в том порядке, в котором они расположены изначально.

Гарантируется, что во всех раундах всех тестов все ключи будут различными. Суммарное количество раундов  $R$  по всем тестам в файле не превышает 1 500 000.

## Формат выходных данных

Для каждого из  $Q$  тестов нужно вывести ровно одну строку. В этой строке должно быть  $N$  целых чисел: значения элементов массива в их финальном порядке после всех раундов.

## Пример

input.txt
5 15 0 b1c6114b f18c80b8 059cace1 24e9297b 5cab5281 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 15 1 b1c6114b f18c80b8 059cace1 24e9297b 5cab5281 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 15 2 b1c6114b f18c80b8 059cace1 24e9297b 5cab5281 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 15 3 b1c6114b f18c80b8 059cace1 24e9297b 5cab5281 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 10 7 4c373cdb 0102026b a8b5ef27 370796de 5840f014 135 12 13 11 10 17 10 7 1 5
output.txt
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 2 6 5 10 3 12 8 14 13 11 4 9 7 15 1 1 11 12 4 5 3 6 14 2 10 8 15 9 13 7 5 12 1 11 8 13 14 15 3 10 2 9 7 6 4 12 1 11 13 135 10 7 10 17 5

## Комментарий

В данной задаче бессовестно жёсткое ограничение по времени.

Во входном файле под номером  $5 + R$  записано 10 тестов с  $50\,000 \cdot R$  раундами суммарно и с максимальным  $N$  (для каждого  $R = 1 \dots 30$ ).

**Внимание:** не пытайтесь применять многопоточность! В nsuts замеряется суммарное процессорное время по всем потокам, поэтому многопоточность не поможет.