Курьерская служба

Вы работаете в отделе, оптимизирующем работу курьерской службы.

Недавно вам поступило *n* заказов, *i*-й заказ имеет массу *ai*, причем каждый из заказов имеет массу, равную 1, 2, 3 или 4 килограммам. Известно, что один курьер может нести с собой некоторый набор заказов (возможно, с одинаковыми массами), но **не может** нести с собой заказы с суммарной массой **более 4 килограмм**.

Примеры некоторых корректных наборов заказов:

* [1, 2] (суммарная масса равна 3);
* [4] (суммарная масса равна 4);
* [1, 1, 1, 1] (суммарная масса равна 4).

И примеры некоторых некорректных наборов заказов:

* [1, 2, 3] (суммарная масса равна 6);
* [4, 1] (суммарная масса равна 5);

Вам необходимо распределить эти заказы между **минимальным** количеством курьеров и сообщить, какие именно заказы должен доставлять каждый курьер. Обратите внимание, что в силу ограничений ответ всегда существует.

**Входные данные**

Первая строка входных данных содержит одно целое число *n* (1 ≤ *n* ≤ 2·105) — количество заказов.

Вторая строка входных данных содержит *n* целых чисел *a*1, *a*2, ..., *an* (1 ≤ *ai* ≤ 4), где *ai* равно массе *i*-го заказа.

**Выходные данные**

В первой строке выведите одно целое число *m* (1 ≤ *m* ≤ *n*) — минимально возможное количество курьеров, необходимое для того, чтобы доставить все заданные товары. Обратите внимание, что в силу ограничений ответ всегда существует.

В следующих *m* строках выведите информацию о курьерах. В *i*-й строке выведите сначала одно целое число *si* (1 ≤ *si* ≤ 4) — количество товаров, которые будет доставлять *i*-й курьер. Затем выведите *si* целых чисел *xi*, 1, *xi*, 2, ..., *xi*,*si*, где *xi*,*j* равно **индексу** *j*-го товара, который будет доставлять *i*-й курьер.

Заметьте, что никакие *xi*,*j* не могут совпадать друг с другом и в их объединении должно быть каждое целое число от 1 до *n* (иными словами, каждый товар должен быть доставлен, причем доставлен ровно одним курьером).

**Примеры**

**входные данные**

7

1 3 1 2 1 1 4

**выходные данные**

4

2 6 2

3 4 5 3

1 1

1 7

**входные данные**

4

4 2 1 3

**выходные данные**

3

2 3 4

1 2

1 1

**входные данные**

9

1 1 2 1 1 2 1 1 1

**выходные данные**

3

2 6 3

4 9 8 7 5

3 4 2 1