Cattery

Exercise

Write a program do manage a cattery, that reads from the input an integer n, and for n times, reads:

- A code to identify the cat (an integer);
- The age of the cat, in years (an integer);
- The weight of the cat, in Kg (a float);
- The type of the food that the cat eats, expressed as 0 for kibbles, 1 for canned food, 2 for tuna fish.

Print, in the same order of input, one per line, the **identification codes** and the **type of food** of all cats that have **less than 4 years and weight more than the average of all cats**. Beside the identification code, print, separated by an empty space, the type of the food that the cat eats, that is `kibbles`, `canned` or `tuna`.

Note: define a `struct` called `cat`, and represent the type of food using an `enum`.

Example

Input

```
 5
2
13
5.0
1
6
3
6.0
0
9
2
3.0
2
1
13
4.0
2
12
1
4.6
2
```

Output

```
6 kibbles
12 tuna
```

Concatenated List

Exercise

Implement a concatenated list that contains, as data, positive integers. Implement three functions, to:

- Add an element at the end of the list;
- Add an element to the beginning of the list;
- Given a positive value v, delete the first node of the list that has v as data (do not modify the list if it does not contain v);

Then, write a C program that read integers.For each integer, (and in the same order), apply one of the following:

- If the read value v is < 0, remove from the list the first element equal to $|v|$ (do not modify the list if it does not contain $|v|$);
- If the read value v is < 0 and even, add it at the beginning of the list;
- If the read value v is < 0 and odd, add it at the end of the list;
- If the read value v is $= 0$, terminate the execution of the program, printing, from the beginning to the end, all elements of the list.

Example

Input

```
 4
5
2
-4
-5
-3
9
2
0
```

Output

```
2
2
9
```