

## A. Assignment 4

time limit per test: 1 second

memory limit per test: 256 megabytes

input: input.txt

You are given the list of animals, which are supposed to live on the same field with growing grass for some number of simulated days. Technically these animals are located in a circle one after another (queue) and have an unchangeable direction, which is looking at the tail of the animal standing in the front. Herbivores can eat the grass, carnivores are able to eat all animals except the ones of their kind. And omnivores are able to eat what both carnivores and herbivores can. Each animal can be eaten only by the animal standing behind it.

All animals have the same types of properties: *animal type*, *weight*, *speed* and *energy*. The animals with **0%** energy are called dead and have to be removed from the list, they cannot be eaten, however eating of an animal can set the victim's energy to **0%**.

Each day costs **1%** of energy for each animal, and it does not matter if the animal has eaten or not, at the end of the day the energy has to be decreased by **1%**.

All animals are able to eat and make sounds. Eating and making sounds does not happen simultaneously. The order is defined by the order in which animals were read from the inputs.

The grass on the field is growing twice its size at the end of each simulation day. However, it is limited by its maximum value **100**. The grass can be eaten only by *herbivores* and *omnivores*. Eating grass is possible for the animal only if its amount is greater than  $\frac{1}{10}$  of the animal's weight. If grass was eaten, then the animal gains energy equal to  $\frac{1}{10}$  of its weight. However, the energy is limited by **100%**. Animals are not intelligent and prefer to eat the grass equal to  $\frac{1}{10}$  of its weight even if they have **100%** energy. Assume that the weight of all animals never changes.

The *carnivores* cannot eat grass, they eat only other animals of not their type (we assume that cannibalism is not applied in this simulation). Hunting the prey is possible only if the speed of the prey is less than the speed of the hunter *OR* the energy of the prey is less than the energy of the hunter. If hunting happens, then the energy of the hunter increases by the number of the prey's weight, the energy is limited by **100%**. The prey's energy goes to **0%**, so it dies. The hunters cannot eat the prey partially, so the prey dies even after the attack of the hunter having **100%** energy.

The omnivores eat both grass and other animals. However, they prefer to start with grazing in the field and then they hunt the prey. Eating the grass does not prevent hunting the prey even for full omnivores.

Each animal produces its own sound. The Lion says "Roar", the Zebra says "Ihoho", the Boar says "Oink".

You are asked to follow the UML class diagram given below. However, you may extend it, if required.



[Link to the diagram](#)

### Input

You are supposed to read the file called `input.txt` having the following format. The first line contains the number of days  $D$  ( $1 \leq D \leq 30$ ) to be simulated. The second line contains the grass amount on the field  $G$  ( $0 \leq G \leq 100$ ). The third line contains the initial number of animals on the field  $N$  ( $1 \leq N \leq 20$ ). The next  $N$  lines are containing initial animal properties separated by a single space for  $N$  animals.

These properties are:

1. Type  $T$ , which can be `Boar`, `Lion` or `Zebra`
2. Weight  $W$  ( $5 \leq W \leq 200$ )
3. Speed  $S$  ( $5 \leq S \leq 60$ )
4. Energy  $E$  ( $0 \leq E \leq 100$ )

The next lines should be ignored regardless of their existence or absence

### Output

You are supposed to provide the outputs to the standard `Console`. It should contain messages separated by new lines. The messages include exception messages followed by sounds produced by each survived animal (if any) in the order they were inserted.

### Exceptions Priority

The inputs may contain errors, which should lead to one of the error messages provided below. After this message the program should finish its execution.

1. The grass is out of bounds
2. Invalid number of animal parameters
3. Invalid inputs
4. The weight is out of bounds
5. The speed is out of bounds
6. The energy is out of bounds

Also, during the simulation some exceptions may appear that should NOT stop execution of the program, they should be printed as an output. The next priority of exceptions should be applied:

1. Self-hunting is not allowed
2. Cannibalism is not allowed
3. The prey is too strong or too fast to attack

### Examples

input

Скопировать

2  
50F  
5  
Lion 100.0F 30.0F 3F  
Lion 100.0 30.0 0  
Zebra 90.0 50.0 2  
Boar 100.0 30.0 3  
Boar 100.0 30.0 3

output

Скопировать

Cannibalism is not allowed  
The prey is too strong or too fast to attack  
The prey is too strong or too fast to attack  
Roar  
Oink

input

Скопировать

1  
50F  
2  
Lion 100.0F 30.0F 3F  
Zebra 90.0 50.0 2

output

Скопировать

Roar

input

Скопировать

3  
50  
2  
Zebra 90.0 50.0 2  
Lion 100.0F 30.0F 3F

output

Скопировать

The prey is too strong or too fast to attack  
The prey is too strong or too fast to attack  
The prey is too strong or too fast to attack  
Ihoho

input

Скопировать

3  
50F  
5  
Lion 100.0F 30.0F 3F 4.0  
Lion 100.0 30.0 0  
Zebra 90.0 50.0 2  
Boar 100.0 30.0 3  
Boar 100.0 30.0 3

output

Скопировать

Invalid number of animal parameters

input

Скопировать

2  
50  
1  
Lion 90.0F 60F 3F

output

Скопировать

Self-hunting is not allowed  
Self-hunting is not allowed  
Roar

Note

Each input file ends with a new line character. Make sure to insert one newline ("`\n`") character after each line written to the standard output.