Dining Philosophers in a Delicatessen!

A variation of the classic Dining Philosophers problem is the one where the philosophers get a “ticket”, i.e. a number that identifies their priority. The philosophers are supposed to eat based on this priority, e.g. if the last one to dine had a ticket value of 7, then the next one to be allowed to dine, regardless if he has to wait for the release of one or two forks, is the one with a ticket value of 8.

Each philosopher is supposed to go through the following sequence of actions:

- **`think()`** : replace with a random delay in [0, 1]sec
- **`getTicket()`** : should be implemented in a **monitor**
- **`getForks()`** : should be implemented in a **monitor**
- **`eat()`** : replace with a random delay in [0, 1]sec
- **`releaseForks()`** : should be implemented in a **monitor**

You program should be able to work for a variable number of philosophers that are specified in the **command-line**. E.g.

```
$ ./deli 10
```

should run the program for 10 philosophers.

Your output should be similar to the following:

```
Philosopher 1 got ticket 0
Philosopher 0 got ticket 1
Philosopher 1 got forks
Philosopher 1 ate
Philosopher 2 got ticket 2
Philosopher 4 got ticket 3
Philosopher 1 released the forks
Philosopher 0 got forks
Philosopher 2 got forks
Philosopher 2 ate
Philosopher 0 ate
Philosopher 3 got ticket 4
...
```