Problem 4. (10 points):

A newly opened cat and dog grooming salon is able to groom a maximum of 3 pets simultaneously, but only if all the pets are of the same species; i.e., cats and dogs cannot occupy the salon at the same time.

You are to implement the bodies of the cat and dog threads, avoiding deadlock and starvation, and allowing for maximum concurrency.

You must use the following semaphore to ensure categorical mutex for the salon:

```
salon_mtx = Semaphore(1)
```

In addition, you may declare any number of semaphores and/or lightswitches for use in your solution. The lightswitch class definition is shown on the last page of the exam.

Upon entering the salon, each pet should call `groom()` to be groomed. Write your solution below:

Shared variables:

```
salon_mtx = Semaphore(1)
```

. . . . . .

Cat thread:

Dog thread:

Lightswitch Pattern

```
# Class definition

class Lightswitch:
    def __init__(self):
       self.counter = 0
       self.mutex = Semaphore(1)

    def lock(self, semaphore):
       self.mutex.wait()
           self.counter += 1
           if self.counter == 1:
              semaphore.wait()
       self.mutex.signal()

    def unlock(self, semaphore):
       self.mutex.wait()
           self.counter -= 1
           if self.counter == 0:
              semaphore.signal()
       self.mutex.signal()
```

Feel free to detach this page!