Use non-blocking approaches to write the following programs.

1. A program uses two thread classes to simulate a queuing system in a bank. The first class `CallingQueue` calls numbers 1 to 10, with a 200ms pause in between each number call. The second class `CustomerInLine` checks whether a given number is called. The third class `BankingQueue` is the queue data structure which also contains a driver class. The driver randomly picks an integer between 1 and 10 and produces an output like below when the random number is 4.

```
Calling for the customer #1
Calling for the customer #2
Calling for the customer #3
Calling for the customer #4
Great, finally #4 was called, now it is my turn
Calling for the customer #5
Calling for the customer #6
Calling for the customer #7
Calling for the customer #8
Calling for the customer #9
Calling for the customer #10
```

2. Write a program that has a global variable `counter` and two tasks. One task increments `counter` by 1 every time it runs. The other task checks the value of `counter` and stops the program when `counter` has reached 5000. A sample output of the program is given below:

```
Counter incremented: 1
Counter changed: 1
Counter incremented: 2
Counter changed: 2
Counter incremented: 3
Counter changed: 3
…
Counter changed: 5000
Counter incremented: 5000
```