Lecturer: Simina Fluture

Homework 2
Due: Wednesday, March 9th

A. Describe a story with concurrent threads that are synchronized by signal and continue monitors.
Describe the situation (by giving the **execution sequence)** that will show the need for rechecking the condition variable using a *while* loop. Briefly explain.

B. Consider the following **savage-cook** monitor version.

1. Consider the first version of the implementation in which we had only one signal in each method.

Given service methods: TakeServing(), PutServing(), with local variables PotSize = M and AmountServings = 0, and queues NotEmpty and EmptyPot.

```
TakeServing():
    while (AmountServings == 0){
        signal(EmptyPot);
        wait(Food);}
    AmountServings--;

PutServing():
    If(AmountServings != 0) wait(EmptyPot);
    AmountServing = M
    signal(NotEmpty);
```

Give the execution sequence that shows deadlock or delay (depending on the fact that savage execution code is or is not a while(true) loop).

2. Do we need to have ***if (AmountServings !=0) wait(EmptyPot)*** in the Cook pseudo-code ? Or, if a savage finds the pot empty it will signal the Cook anyway and ***wait(EmptyPot)*** only, will be enough?
If it is not enough, then what will be the outcome of the above change? (what condition of Critical Section Problem might be violated?) Give the execution sequence that will show it.

3. The savage waits if there is **EmptyPot**, the cook waits for **NotEmpty**.

a. Can we, instead of two condition variables, use only one? (let's say: **potStatus**). Discuss.
b. Is the assumption: *If we have only one condition variable PotStatus, savage and cook cannot block at the same time on the condition variable* correct?

C. **Review the 3 implementations of the reader-writer problem:**

Signal and Exit Monitor:	**rwse.java**	**rwdr.java**
Signal and Continue Monitor:	**rwmo.java**	**rwdr.java**
Signal and Continue with Notification Objects:	**rwcv.java**	**rwdr.java**

For each implementation explain how the reader-writer policy is enforced and how the No Starvation condition is satisfied. Mention the main differences among the three implementations.

D. No submission necessary.
Read and understand problem 5.10 (Stephen Hartley) about the race condition notify after interrupt.