Assignment Part1 (Promela and Spin).
Due date Mon Apr 25, 2022

You have to upload your solution by the due date via Google classroom. Your should submit a zip containing the Promela source files and an additional document with your answers/explanations for each question. In order to solve these problems, you have to install Spin from `spinroot.com`.

1. Dekker's mutual exclusion algorithm tries to ensure that at most one of two processes is in a critical section at any time. The algorithm is described by the following pseudo-code:

```
// Dekker's mutex algorithm, two parallel processes 0 and 1

var flag: array[0..1] of boolean;
// flag is all initialized with false,

var turn: 0;
// turn has the initial value 0

// The algorithm for process i then is:

// i is my index, j is the other process
i := mypid();
j :=1-mypid();

// Infinite loop
while true do
    // [noncritical section]
    flag[i] := true;

    // [trying section]
    while flag[j] do
        if turn = j then
            begin
                flag[i] := false;
                while turn = j do
                    idle
                enddo;
                flag[i] := true;
            end;
        endif;
    enddo;
```

```
    // [critical section]
    turn := j;
    flag[i] := false;
enddo;
```

a) Model Dekker's algorithm presented above in Promela, the input language of the Spin model-checker (`http://www.spinroot.com/`). Explain your model.

b) Add assertion mechanism into the code which triggers when both processes are in the critical section at the same time. Explain. Check with Spin whether Dekker's algorithm guarantees mutual exclusion for two processes. Submit the verification output given by Spin and explain the result.

Hint: Use additional variable(s) to model when a process is in the critical section.

c) Add LTL propertie(s) outside the model (processes) that expresses that always at most one process can be in the critical section. Check with Spin whether Dekker's algorithm satisfies the LTL property. Submit the verification output given by Spin and explain the result.

d) In the book "M. Raynal. Algorithms for mutual exclusion (1986)" the following simpler variant of the inner loop (trying section) is suggested:

```
if flag[j] then
    begin
        if turn = j then
            begin
                flag[i] := false;
                while turn = j do
                    idle
                enddo;
                flag[i] := true;
            end;
        endif;
    end;
endif;
```

Does the modified algorithm work? Explain the result.