Problem 1. In an acyclic network, also called a tree, each node represents a process that has a color 0 or 1. Starting from an arbitrary initial configuration, our goal is to color the nodes with these two colors in such a way that no two neighboring nodes have the same color. We propose the following algorithm for each process i:

```plaintext
program twocolor
define c[i]: color of process i

do \ j \neq i : c[j] = c[i] \Æ c[i] := 1 - c[i] od
```

Assume that the scheduler is weakly fair. Will the algorithm terminate? If not, then explain why not. Otherwise, give a proof of termination.

Problem 2. Now, consider that you have a rooted tree with a designated root. Each node except the root has a neighbor that is designated as its parent node. The parent relationship is acyclic if j is the parent of i then i cannot be the parent of j. Now, try the same problem as in previous one, but this time try a different algorithm:

```plaintext
program treecolor
define c[i]: color of process i
p[i]: parent of process i
```

\begin{verbatim}
do c[i] = c[p[i]] \quad c[i] := 1 - c[i] \quad od
\end{verbatim}

Assume that the scheduler is *weakly fair*. Will the algorithm terminate? If not, then explain why not. Otherwise, give a proof of termination.

**Problem 3.** Consider a strongly connected network of $n$ processes $0, 1, 2, 3, \ldots, n-1$. Any process $i$ called the source can send a *message* to any other process $j$ called the destination in the network. Every process $i$ has a local variable $v_i$. A message sent out by a source process $i$ consists of a the sequence number of the message, b the source id, c the destination id, and d the value of $v_i$.

Each message has to be routed through zero or more processes. A process $j$ receiving a message accepts the message only if it is the destination, and executes the assignment $v_j := v_i$ otherwise, it forwards the message along its outgoing edges. The communication is considered to be complete, when at least one copy of the message reaches the destination, and the value of the local variable is appropriately updated.

Propose an algorithm for sending the message between any pair of points by specifying the life of a process using guarded actions. Make sure that your algorithm terminates, and copies of the message do not remain in circulation forever. Also briefly argue why your algorithm will terminate.

**Problem 4.** Consider a completely connected network of $n$ processes, where each process executes actions infinitely often. A *round-robin scheduler* guarantees that between two consecutive actions by the same process, every other process with enabled guards executes their action at least once.

Assuming that at most one action can be scheduled at any time, implement *round robin* scheduling. Provide brief arguments in support of your implementation.

**Problem 5.** Studious philosophers. Three philosophers 0, 1, 2 are sitting around a table. Each philosopher's life alternates between reading and writing. There are three books on the table $B_0, B_1, B_2$ each book is placed between a pair of philosophers as shown below:
While reading, a philosopher needs to grab two books—one from the right and one from the left. After the philosopher finishes reading, he takes notes, and puts the books back on the table.

a. Propose a solution by describing the life of a philosopher. Your solution must work with a strongly fair scheduler but may not work with a weakly fair scheduler.

b. Next, propose a solution once again by describing the life of a philosopher that will work with a weakly fair scheduler too. Provide brief arguments why your solution will work.