

Exercise 8

This exercise is a continuation of Exercise 7.

We ask to write a function `check` that, given an initial configuration and a list of moves for the Hanoi Towers with 3 pegs, checks that all moves are ok, where a move is ok when:

- i) it does not ask to move a disk from an empty peg,
- ii) it does never put a disc on top of a smaller disk
- iii) it only uses pegs "a", "b", and "c".

The function `check` has the following type

`check :: Conf -> Int -> [Move] -> (Report, Int, [Move], Conf)`

where, `data Report = Bad | Ok` deriving `Show`

`type Conf = ([Int],[Int],[Int])`. A configuration is a triple showing the content of the 3 pegs. So, for instance, for 4 disks, this `[[1,2],[3],[4]]` is a correct configuration. The initial configuration is `[[1,2,3,4],[],[[]]]`, the final configuration is `[[],[],[1,2,3,4]]`.

The type `Report` is used to expose the result of the check. Obviously `Bad` indicates that a wrong move was found and `Ok` that all moves are ok.

The second parameter of `check` (`Int`) is meant to contain the number of the moves executed so far. It is necessary to output the total number of moves executed that is the second component of the result of `check`, as explained below. The type of the result of the function `check` is a quadruple `(Report, Int, [Move], Conf)`, where the four parts are as follows:

The meaning of the report (`Bad/Ok`) is obvious; the second component is the total number of moves executed; the third component is either `[]` when all moves are ok, or is `[m]`, where `m` is the first wrong move encountered; the last component is the final configuration reached either when all moves are executed or when a wrong move is found. When a wrong move is found, the function `check` should stop its computation.

For this exercise you should upload a file `Hanoi_check.hs` that contains, both functions `hanoi` (of Ex 7) and `check` and all the functions and types that they need.

In the moodle you will find a `Main.hs` file containing the main that calls both your `hanoi` and `check` functions with different inputs.