Exercise 7

TheTowers of Hanoi is a classic puzzle with a solution that can be described recursively. Disks of different sizes are stacked on three pegs; the goal is to get from a starting configuration with all disks stacked on the first peg to an ending configuration with all disks stacked on the last peg, The figure below shows the initial and the final configuration. The only rules are: you may only move one disk at a time, and a larger disk may never be stacked on top of a smaller one.

Starting from the initial configuration, as first move we can move the topmost smallest disk (disk 1) onto a different peg, say b. In fact only one disk may be moved at a time. From this point, it is illegal to move disk (2) to peg b, because it is not allowed to put a bigger disk on top of a smaller one, but it would be possible to put disk (2) on peg c and then move disk (1) on top of (2). In general, to move n disks (stacked in increasing size) from peg a to peg c using peg b as temporary storage, we should follow

1. move n−1 disks from a to b using c as temporary storage

2. move the top disk (disk 5 in the figure) from a to c

3. move n−1 discs from b to c using a as temporary storage.

For doing these steps, define a function `Hanoi` with the following type:

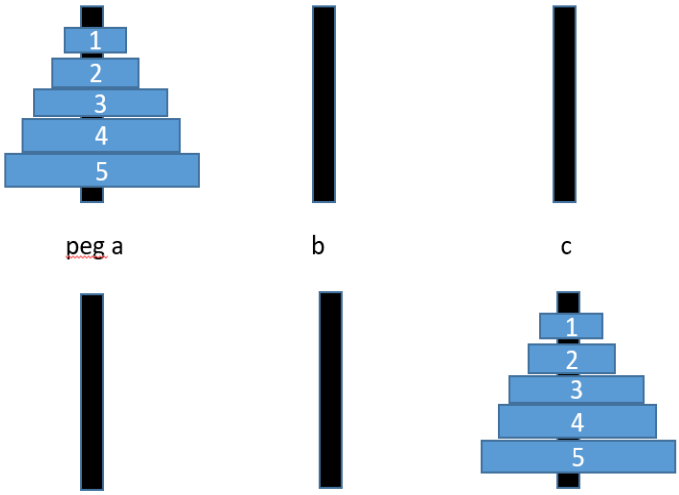
```
type Peg = String

type Move = (Peg, Peg)

hanoi :: Integer -> Peg -> Peg -> Peg -> [Move]
```

As you see from the types, the function hanoi, won't compute any intermediate configuration of the towers on the pegs. It only computes the sequence of moves that must be applied to reach the final configuration.

A call hanoi m x1 x2 x3 produces the sequence of moves that transfer the m disks from peg x1 to peg x3 assuming that peg x2 can be freely used as temporary storage. Obvoiusly x1, x2 and x3 are any permutation of "a", "b" and "c".

You should upload a file Main.hs containing the module Main that contains a main function (of type IO ()) that reads an integer n = the number of disks to consider and then it prints the result of the call hanoi n "a" "b" "c".

The sequence of moves produced by such calls of your function hanoi will be compared with the shortest sequence of moves. Therefore, to pass the tests, your function should produce the shortest sequence.


1
2
3
4
5
peg a
b
c
1
2
3
4
5