Exercise 10

We want to use ST as type for a program that, given a string containing round and squared parentheses (and any other symbol that will be simply ignored) will check whether the parentheses are well balanced. For instance, "((([])))" is well balance, whereas "(([]))" is not since the last ')' has no matching '(' at is left. Clearly ((([]]) is not balanced, since a '(' is matched by a ']'.

You are supposed to write a function balance using the following types:

type State = (String, String)

In a State = (String, String), the 1st component is a stack used to remember the open parentheses encountered so far, whereas the 2nd component is the input string that still needs to be checked.

newtype CheckPar a = CP (State -> (a,State))

The function balance should have type, CheckPar Bool. The result Bool will indicate whether the given string is balanced or not. Balance is such that in case of error produces (False, (x,y)) where x is the stack and y is the remaining input when an error is detected. In case of success, balance should produce (True,([],[])), indicating that both, input and stack must be empty.

CheckPar has the usual function app:: (CheckPar a) -> State ->(a,State).

The function balance (and all definitions that it needs) is supposed to be in module Balance contained in file Balance.hs.

You must also turn in a Main.hs with a very simple main:: IO(), that reads a line l from the input, and then prints (app balance l).