Programación Declarativa Aplicada – Curso 2016-17 - Práctica 3

Evaluación: Entrega en clase: 0.5. Entrega la siguiente semana: 0.25 (más adelante no puntúa)

Escribir un módulo "pract3.erl" con las siguientes funciones.
Nota: Las funciones deben exportarse explícitamente con la directiva -export([….]).

1) different(F,G,L) : devuelve la sublista de L formada por aquellos elementos X en L tales que F(X) <> G(X). Hacerlo utilizando la función maps:filter.

2) Para probarlo, definir dos funciones (que no hace falta exportar):
- abs(X): devolverá el valor absoluto de X.
- id(X): devolverá simplemente X (la identidad).
Escribir, además, una función different_test() que comprueba (?assertEqual) si las funciones abs e id aplicadas a la lista lists:seq(-3,10) devuelve [-3,-2,-1].

3) differentBis(F,G,L) : mismo enunciado que en el ejercicio 1, pero con la implementación usando listas intensionales.

4) differentBis_test() análoga a 2) pero para probar differentBis.

5) equal(F,G,L) que comprueba si las dos funciones son iguales para todos los valores en L. Debe apoyarse en la función different (ejercicio 1).

6) equal_test() que comprueba (?assertEqual) que las funciones abs y la propia abs son iguales en el intervalo de enteros [-10,…,10].

7) morosos(L) que recibe una lista L de la forma: [{bertoldo, 500}, {herminia, cancelada}, {aniceto,-2000}] y devuelve la lista de nombres de las personas que tienen saldo negativo. Debe constar de una única regla y usar listas intensionales.

8) morosos_test() que compruebe que la función morosos aplicada a [{bertoldo, 500}, {herminia, cancelada}, {aniceto,-2000}] devuelve [aniceto].

9) triángulo(N) que devuelva la lista [[1]….[1...N]]. Por ejemplo triángulo(10).
[[1],
[1,2],
[1,2,3],
…..
[1,2,3,4,5,6,7,8,9],
[1,2,3,4,5,6,7,8,9,10]]
la función estará definida con una sola regla, utilizando listas intensionales.

10) maxLists(LF,LV), donde LF = [F1,…, FN] es una lista de funciones y LV=[V1,…,VM] una lista de valores. La función maxLists(LF,LV) devolverá

[max(F1(V1),…,FN(V1)), …,max(F1(VM),…,FN(VM))]

con max la función máximo. La función estará definida por una sola regla y usará listas intensionales. Sugerencia: usar lists:foldl/3 y erlang:max/2 como funciones auxiliares.
Ejemplo:
pract3:maxLists([fun(X)->X+X end, fun(X)->X*X end, fun(X) -> X+3 end], lists:seq(1,3)).
[4,5,9]