Unification - 20 points

You need to implement the method:

process(p1, p2)

where p1 and p2 are Prolog predicates
The result of this processing should be the result of unifying those two predicates, which should indicate if the unification was successful. In the case of a successful unification, the result should contain the list of substitutions.

You are free to add new parameters to this method, if necessary. The predicates should be immutable, which means that the predicates cannot be altered in any way.

A predicate has a name and a list of arguments. The arguments are called terms. A term can be a predicate, a constant or a variable. Example: factorial(0, 1), parent(maria, vasile), persoana(ion), tree(1, tree(2, leaf(3)), leaf(4)).
A constant has a value, which can be a number or a string. Example: 123, gigel.
A variable has a name, which is a string starting with an upper case. Example:

Un predicate unifies with another predicate only if they have the same name, the same number of arguments and the corresponding arguments unify. The result contains a list of substitutions. A substitution is a pair variable/term which signifies that the variable should be substituted with the term.
A variable unifies with any term and the result is a substitution formed by the variable and the term.
A constant unifies with another constant only if their values are identical.

Examples:

1. *pred(1)* unifies with *pred(1)* and produces an empty list of substitutions.
2. *pred(X)* unifies with *pred(1)* and produces: {X/1}
3. *pred(X, X)* unifies with *pred (1, Y)* and produces: { X/1, Y/1}
4. *pred(1)* does not unify with *sub(1)*.
5. pred(X, X) does not unify with pred(1, 2)
6. pred(sub(Z), 3) unifies with pred(X, Y) and the substitutions are : {X/sub(Z), Y/3}
7. pred(sub(Z), Z) unifies with pred(sub(Y), 3) and: {Z/3, Y/3}