Write a LISP function called `TheoremProver` to implement a theorem prover for First Order Predicate Logic using resolution refutation technique and the `Set of Support` strategy. This function gets two lists of clauses; the list of base clauses and the list of clauses obtained from the negation of the theorem, and a list of variable names. Your program has to eliminate

- tautologies and
- subsumptions.

The function returns whether the theorem is derivable, or not. If derivable, it has to print the resolutions that contribute to the proof of the theorem.

Variables, predicate names and function names are atoms starting with a lower case letter, while constants with an upper case letter. Note that variables that appear in the clauses are given in a list as the third argument of the function. You have to use the following list convention for the two clause sets:

```
> (TheoremProver '(( (p A (f x)))
    ((~ (q x)) (r A)) )
  '( (q B) (~ (r y)))
  '(x y) )
```