LAB 3

S is a finite set of propositional clauses, written in CNF in the format [[w, s, n(p)], [a, n(w), r, t], [q]]. With n(p) the negation of p is noted.

> Implement the Resolution procedure. For the input data S, the procedure will display SATISFIABLE, respectively UNSATISFIABLE, as it is the case. Set a strategy for choosing which pair of clauses to use when Resolution rule is applied.

The input data will be read from a file and the results will be displayed in a file.

The procedure will be implemented in the version presented at the course (from Ronald Brachman, Hector Levesque. Knowledge representation and reasoning, Morgan Kaufmann 2004).

Suggestion for implementation:

```
res(KB) :- member([], KB).
res(KB) :- …choose two clauses from KB, apply Resolution, add the Resolvent (if new) to KB
                =>newKB…, res(newKB).
```

Reading/writing in Prolog

The data in the input file is separated by .

```
see (user).       % opens the current input environment - the user represents the keyboard
see ("c:\\prolog\\a.txt ').
seen.   % closes the current reading environment

read (X).
read (end_of_file).     % special atom that detects the end of the file


tell ("c:\\prolog\\a.txt ').          % opens the current output environment
told.    % closes the current output environment

write(parent(ion,maria)).
```

Dynamic predicates in Prolog

-declared with   :-dynamic p/1.   %PredicateName/arity

-predicates for dynamic addition: asserta, assertz, assert (add at the beginning, at the end)
-predicates for dynamic deletion: retract, retractall.

```
:-dynamic fib/2.

fib(1,1).
fib(2,1).
fib(N,F):-N>2, N1 is N-1, fib(N1,F1), N2 is N-2, fib(N2,F2), F is F1+F2, asserta(fib(N,F):-!).
```