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. Apply optimization techniques for the Resolution algorithm.
The input data will be read from 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('c:\\prolog\\a.txt').        % opens the current input environment
                                 % user is a special atom used to switch to the default input
                                 environment (keyboard)
                                 % see(user).
…
read(X).                         % end_of_file is a special atom returned when the end of file is
                                 % reached
…
seen.    % closes the current reading environment

tell('c:\\prolog\\b.txt').       % opens the current output environment
                                 % tell(user) switches to the default output environment (screen)
…
write(parent(ion,maria)), write('.'), nl.
…
told.    % closes the current output environment
```

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/somewhere)
-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):-!).
```

Renaming variables

copy_term(+In, -Out). % https://www.swi-prolog.org/pldoc/man?predicate=copy_term/2

Check the following:

```
?-copy_term(X,Y), X\==Y.

?-copy_term(parent(X,Y),Z).

?-copy_term(parent(X,Y),Z), parent(X,Y)==Z.

?-copy_term(parent(X, ana), Z).
```