CSE341 – Programming Languages (Fall 2020)
Homework #5 (Replacement for Midterm)

In this project we are asked to write a Common Lisp function such that when a parsed list of Horn clauses , it will answer all the queries in this list of clauses. You will use resolution and unification methods discussed in class to prove the queries and return the list of values for all variables for which the query is true. An empty list on return means that the query is false. Make sure that your query answer function halts.

In this project, I didn't read from the input.txt. So I will create the clauses on my own. I wrote one example in additional. I will explain this in below. So I wrote the result to the output.txt file. For example my clauses are them :

(list (list (list "legs" (list "X" 2)) (list (list "mammal" (list "X")) (list "arms" (list "X" 2))))
(list (list "legs" (list "X" 4)) (list (list "mammal" (list "X")) (list "arms" (list "X" 0))))
(list (list "animal" (list "horse")) ())
(list (list "mammal" (list "horse" "lion" "tiger" "mouse" "cat")) ())
(list (list "arms" (list "horse" 0) ()))
(list () (list "animal" (list "X"))) ;;First query
(list () (list "mammal" (list "horse" "lion" "X" "Y" "cat"))))) ;;Second query

In the second query, the fact is mammals are horse,lion,tiger,mouse and cat. And the query is horse,lion, X, Y and cat. So, the tiger matches with X and the mouse matches with Y. I return this list of result like ((“tiger” “mouse”)). I printed result to terminal and the output.txt file.

In the first query, the fact is horse is an animal. And the query is animal “X”. So the X matches with horse and the result will be horse.

```
(setq propositions  ;;This prepositions is from the homework pdf but I just changed the mammal list and also query.
    (list
        (list (list "legs" (list "X" 2)) (list (list "mammal" (list "X")) (list "arms" (list "X" 2))))
        (list (list "legs" (list "X" 4)) (list (list "mammal" (list "X")) (list "arms" (list "X" 0))))
        (list (list "animal" (list "horse")) () )
        (list (list "mammal" (list "horse" "lion" "tiger" "mouse" "cat")) () ) ;;fact
        (list (list "arms" (list "horse" 0) () ))
        (list () (list "animal" (list "X")))  ;;First query
        (list () (list "mammal" (list "horse" "lion" "X" "Y" "cat")) ))) ;Second query
;; The fact list is horse,lion, tiger,mouse, cat
;; And the query is horse,lion, X , Y, cat
;; So the X will be tiger and the Y will be mouse.
;; (tiger,mouse) will be printed.
```

For this clauses, the output will be like this ;

```
emir@emir:~/Desktop/bittik/midterm$ clisp midterm.lisp
(("horse"))
(("tiger" "mouse"))
```

That means ;
- Horse is an animal (X = Horse)
- Tiger and mouse are mammals (X = Tiger, Y = Mouse)

I did these ;

- Create the clauses
- Analyze these
- Determine it is predicate or fact
- Find the unknown elements
- If unknowns matches with given clauses, print these
- Otherwise print nil
- Print the result to output.txt and terminal

I couldn’t these
- Read from file
- After the determination which is predicate, then I couldn’t print the result.