ΑΤΕΙ ΘΕΣΣΑΛΟΝΙΚΗΣ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ
ΜΑΘΗΜΑ: ΕΡΓΑΣΤΗΡΙΟ ΤΕΧΝΗΤΗΣ ΝΟΗΜΟΣΥΝΗΣ (Prolog)
ΕΞΑΜΗΝΟ: Δ΄- Εαρινό 2018-19
ΚΑΘΗΓΗΤΕΣ: Δ.ΣΤΑΜΑΤΗΣ, Μ.ΒΟΖΑΛΗΣ, Κ. ΚΡΑΒΑΡΗ

ΕΠΑΝΑΛΗΠΤΙΚΕΣ ΑΣΚΗΣΕΙΣ

A1:

Να γραφεί πρόγραμμα το οποίο να ελέγχει εάν σε μία λίστα υπάρχουν δύο διαδοχικά στοιχεία με την ίδια τιμή.

```
/*twoins ( [X,X|T]  ).  */   /* true*/ /*if the first and second is  idia
twoins(X, [X,X|T]  ).
twoins(X,[Y|T]):-
   twoins(X,T).
```

A2:

Να γραφεί πρόγραμμα το οποίο να ελέγχει εάν σε μία λίστα υπάρχουν δύο διαδοχικά στοιχεία που το άθροισμά τους ισούται με 100.

```
two100([X,Y|T]):-
    100 is X+Y.
/* check two first number if (x+y)=100
     Only first and second */
two100([X|T]):-
    two100(T).
/*check if kapoy mesa stin oura eiparxoyn
    2 sinexomena number poy athroisma is 100*/
```

A3:

Να γραφεί πρόγραμμα το οποίο να ελέγχει εάν σε μία λίστα υπάρχουν δύο και μόνον δύο στοιχεία με την ίδια τιμή.

```
twonly(L):-
  delete(X,L,L1),
  delete(X,L1,L2),
  not(member(X,L2)) .
/*?- twoins(5,[5,5,6]).
      true ;
      False.
*/
```

A4:

Να ορισθεί το κατηγόρημα **count(L, A)** το οποίο δέχεται μια λίστα αριθμών **L** και επιστρέφει μία λίστα **A** που έχει δύο στοιχεία. Το πρώτο αντιστοιχεί στο άθροισμα των περιττών αριθμών που βρίσκονται στην λίστα **L** ενώ το δεύτερο στο

άθροισμα των άρτιων αριθμών. (Γράψτε και χρησιμοποιήσετε το κατηγόρημα **iseven/1** που ελέγχει αν ένας αριθμός είναι ζυγός: **iseven(Q):- 0 is mod(Q,2)**.) Παράδειγμα:

?- count([3,6,7,2,9,5], A).

A = [24, 8].

```
count([],[0,0]).
iseven(Q):-
0 is mod(Q,2).
count([H|T], [P,TA] ):-
   iseven(H), /*0 is mod(H,2),*/
   count(T,[P,A]),
   TA is H+A.
count([H|T],[PT,A]):-
   not(iseven(H)), /*1 is mod(H,2),*/
   count(T,[P,A]),
   PT is H+P .
/* count([1,3,2,6],P).
   P = [4, 8] ;
   false.*/
```

A5:

Γεγονότα σαν τα παρακάτω περιγράφουν τον τόπο καταγωγής, το μάθημα και τη βαθμολογία φοιτητών που πήραν μέρος στις εξετάσεις της Α' Εξεταστικής Περιόδου:

data(kostas, thessaloniki, prolog, 3).

data(maria, lamia, java, 7).

data(nikos, xania, prolog,9). κ.ο.κ.

Δίνεται μία λίστα **L1** η οποία περιέχει ονόματα φοιτητών. Να γραφεί ένα κατηγόρημα το οποίο να επιστρέφει μία δεύτερη λίστα **L2**, η οποία περιλαμβάνει μόνον τα ονόματα εκείνα της **L1** που αντιστοιχούν σε φοιτητές από τη Θεσσαλονίκη που δεν έχουν περάσει το μάθημα της **prolog**.

```
data(kostas,thessaloniki, prolog, 3).
data(kat,thessaloniki, prolog, 7).
data(maria,lamia, java, 7).
data(nikos,xania, prolog,9).
tonthelo(X):-
   data(X,thessaloniki,prolog,B),
   B>=5.
/* ?_tonthelo(kostas).
False.
    ?_tonthelo(kat).
true.
*/
```

```
process([],[]).
process([H|T],[H|NT]):-
    tonthelo(H),
    process(T,NT).
process([H|T],NT):-
    not(tonthelo(H)),
 process(T,NT).
/*?- process([kostas,maria,nikos,kat],L2).
      L2 = [kat] ;
           False.
*/
```

A6:

Σε γεγονότα της μορφής **word([m,o,n,d,a,y])**.
ορίζουμε ορθογραφικά σωστά λέξεις. Να ορισθεί το κατηγόρημα **swap_first_last(W,CW)** το οποίο θα δέχεται σαν είσοδο τη λίστα W που αντιστοιχεί σε μία πιθανά ανορθόγραφη λέξη και επιστρέφει στη λίστα **CW** μια σωστά ορθογραφημένη λέξη η οποία προήλθε από την αντιμετάθεση του πρώτου και του τελευταίου γράμματος της λέξη **W** .
Παράδειγμα:

?- swap_first_last([y,o,n,d,a,m], L).
L = [m,o,n,d,a,y]

```
word([n,i,k,o,s]).
append([ ], List, List).
append([Head|Tail],List,[Head|NewList] ) :-
    append(Tail,List,NewList).

swap_first_last(W,CW) :-
    append([X|L],[Y],W),
    append([Y|L],[X],CW),
    word(CW).
/*?- swap_first_last([s,i,k,o,n],CW).
      CW = [n, i, k, o, s] ;
      False.
*/
```