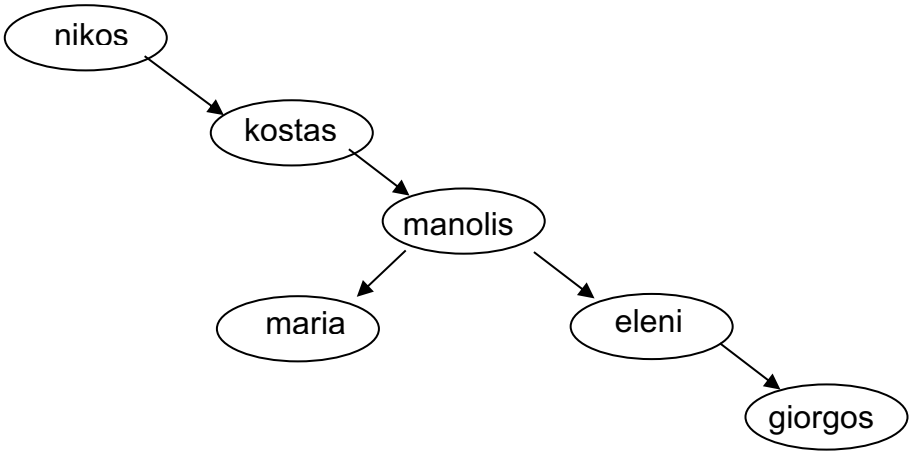
ΕΡΓΑΣΤΗΡΙΟ IV

ΑΝΑΔΡΟΜΗ & ΤΕΛΕΣΤΕΣ ΩΣ ΣΥΝΑΡΤΗΣΙΑΚΑ ΣΥΜΒΟΛΑ

Με τον όρο αναδρομή εννοείται η δυνατότητα ένας κανόνας να περιέχει στο σώμα του μια κλήση προς τον εαυτό του. Οι κανόνες που χρησιμοποιούν αναδρομή χαρακτηρίζονται ως αναδρομικοί κανόνες. Η χρήση της αναδρομής οδηγεί σε μικρότερα προγράμματα. Όταν ένα κατηγόρημα ορίζεται αναδρομικά εμφανίζεται πάντα με *τουλάχιστον* δύο κανόνες. Από αυτούς, ο πρώτος συνήθως δεν περιέχει αναδρομική κλήση, λειτουργώντας έτσι σαν τερματική συνθήκη. Ο δεύτερος συνήθως περιέχει την αναδρομική κλήση, δηλαδή την κλήση του κανόνα στον εαυτό του, με διαφορετικά ορίσματα, καθορίζοντας την πορεία του κανόνα σε επόμενες κλήσεις. Πρώτα εξετάζεται η τερματική συνθήκη: εάν αυτή αληθεύει, τότε το πρόγραμμα τερματίζεται. Εάν δεν αληθεύει, τότε η εκτέλεση προχωράει στην δεύτερη πρόταση του κανόνα, και όταν φτάσει στην αναδρομική κλήση, ο κανόνας εκτελείται από την αρχή με νέα ορίσματα, γίνεται ο έλεγχος της τερματικής συνθήκης, και ακολουθείται η ίδια διαδικασία μέχρι να ικανοποιηθεί η τερματική συνθήκη (αποτέλεσμα true) ή να μην ικανοποιηθεί η τερματική συνθήκη ποτέ (αποτέλεσμα false).

Στο παράδειγμα που ακολουθεί (exercise4.pl στην διαδρομή: http://iiwm.teikav.edu.gr/digital_lessons/ στο μάθημα Λογική και Λογικός Προγραμματισμός – Ενότητα 4), υπάρχουν γεγονότα γονιός/2 και πέντε διαφορετικές υλοποιήσεις του κανόνα πρόγονος/2, ο οποίος αληθεύει όταν το πρόσωπο στο πρώτο όρισμα είναι πρόγονος του δεύτερου. Στην πιο απλή περίπτωση, εάν κάποιο πρόσωπο Α είναι γονιός του Β, τότε είναι και πρόγονός του. Αυτό χρησιμοποιείται σαν τερματική συνθήκη. Αυτή η τερματική συνθήκη είναι ο πρώτος κανόνας του ancestor, ο οποίος απλά επιτυγχάνει εάν ο Χ είναι ο πατέρας του Υ, χωρίς να έχει καμιά αναδρομική κλήση του εαυτού του. Εάν επιτύχει η τερματική συνθήκη, οι έλεγχοι τερματίζονται και η prolog επιστρέφει απευθείας το αποτέλεσμα. Εάν δεν ικανοποιηθεί αυτή η πρώτη πρόταση, το πρόγραμμα προχωράει στην ταυτοποίηση της δεύτερης πρότασης του κανόνα, γίνεται κλήση του κανόνα από την αρχή, εξετάζεται ξανά η τερματική συνθήκη κ.ο.κ.. Το νόημα των επαναλαμβανόμενων κλήσεων είναι να συνδεθούν δύο πρόσωπα που βρίσκονται σε διαφορετικά σημεία στο ίδιο γενεαλογικό δέντρο, ακόμα και αν δεν συνδέονται άμεσα.




```
parent(nikos,kostas).
parent(kostas,manolis).
parent(manolis,maria).
parent(manolis,eleni).
parent(eleni,giorgos).

ancestor1(X,Y):- parent(X,Y).
ancestor1(X,Y):- parent(X,Z),
                 ancestor1(Z,Y).


ancestor2(X,Y):- parent(X,Y).
ancestor2(X,Y):- parent(Z,Y),
                 ancestor2(X,Z).


ancestor3(X,Y):- parent(X,Z),
                 ancestor3(Z,Y).
ancestor3(X,Y):- parent(X,Y).


ancestor4(X,Y):- parent(X,Y).
ancestor4(X,Y):- ancestor4(Z,Y),
                 parent(X,Z).


ancestor5(X,Y):- ancestor5(Z,Y),
                 parent(X,Z).
ancestor5(X,Y):- parent(X,Y).
```

ΑΣΚΗΣΗ Ι

Να εκτελέσετε τις παρακάτω ερωτήσεις για κάθε μια από τις 5 υλοποιήσεις του κανόνα ancestor, και να συμπληρώστε τον παρακάτω πίνακα με "Σωστό" ή "Λάθος" σε ότι αφορά στη σωστή λειτουργία των παραπάνω πέντε κανόνων στις εξής ερωτήσεις:

1. Είναι ο Νίκος πρόγονος της Μαρίας; (Πίνακας 4.1 – Στήλη 2)

2. Ποιος είναι πρόγονος ποιου; - **φτιάξτε την ερώτηση** (Πίνακας 4.1 – Στήλη 3)

	1. `ancestor(nikos,maria).`	**2. `ancestor( ).`**
Α. `ancestor1`		
Β. `ancestor2`		
Γ. `ancestor3`		
Δ. `ancestor4`		
Ε. `ancestor5`		

Χρησιμοποιήστε την εντολή trace για να δείτε την λειτουργία των κανόνων `ancestor1` και `ancestor3`. Σε ένα πρόγραμμα που θα φτιάχνατε θα προτιμούσατε να χρησιμοποιήσετε τον κανόνα `ancestor1` ή τον κανόνα `ancestor3` και γιατί;

ΑΣΚΗΣΗ ΙΙ

Έστω οι παρακάτω προτάσεις:

```
on_top_of(prolog_book,desk).
on_top_of(ai_notes,prolog_book).
on_top_of(time_table,ai_notes).
on_top_of(ai_book,desk).
```

Να οριστεί ο κανόνας `above(X,Y)` που θα επιστρέφει αληθές αν το Χ είναι πάνω από το Υ. Δηλαδή: `above(ai_notes,desk)` θα επιστρέψει true ενώ `above(desk,prolog_book)` θα επιστρέψει false.

ΜΕΡΟΣ ΙΙ

ΤΕΛΕΣΤΕΣ ΩΣ ΣΥΝΑΡΤΗΣΙΑΚΑ ΣΥΜΒΟΛΑ

Για την εκτέλεση αριθμητικών πράξεων στην Prolog υπάρχουν οι σχετικοί τελεστές : +, -, *, /, mod (υπόλοιπο της ακέραιας διαίρεσης). Στην έκφραση 1+3 το + είναι ένας τελεστής. Είναι δηλαδή, ισοδύναμη με την έκφραση +(1,3) και όχι με τον αριθμό 4, όπως συμβαίνει στις άλλες γλώσσες προγραμματισμού. Η πρόσθεση θα πραγματοποιούνταν αν ο όρος 1+3 ήταν όρισμα στο ειδικό κατηγόρημα (τελεστής απόδοσης τιμής) **is** της Prolog για υπολογισμό αριθμητικών εκφράσεων. Το κατηγόρημα **is** μπορεί να χρησιμοποιηθεί είτε για να αποδώσει τιμή σε κάποια μεταβλητή κάνοντας κάποιον μαθηματικό υπολογισμό ή για να ελέγξει αν η τιμή κάποιας μαθηματικής έκφρασης ισούται με κάποιον αριθμό. Στη μαθηματική έκφραση μπορούν να χρησιμοποιηθούν μεταβλητές μόνο αν έχουν πάρει προηγουμένως κάποια τιμή.

ΑΣΚΗΣΕΙΣ

1. Να δώσετε τα παρακάτω ερωτήματα και να καταγράψετε τις απαντήσεις που λαμβάνετε:

```
8 is 6+2.
3 is 6/2.
X is 6+2.
X is 6*2.
R is mod(7,2).
X is Y+2.
```

2. Εκτελέστε τα παρακάτω ερωτήματα στην Prolog. Σε τι διαφέρουν;

i. X is 2+3.
ii. X = 2+3.

ΜΕΡΟΣ ΙΙΙ

ΤΕΛΕΣΤΕΣ ΣΥΓΚΡΙΣΗΣ

Οι τελεστές που χρησιμοποιούνται για τη σύγκριση μαθηματικών όρων στην Prolog (εκτός από τους γνωστούς **>,<, >=**) είναι οι εξής:

Τελεστές Σύγκρισης	Επεξήγηση
X **=<** Y	Το X είναι μικρότερο ή ίσο του Y
X **=:=** Y	Οι τιμές των X και Y είναι ίσες
X **=/=** Y	Οι τιμές των X και Y δεν είναι ίσες

Οι τελεστές αυτοί μπορούν να χρησιμοποιούνται στα σώματα κανόνων, με σκοπό τον έλεγχο συνθηκών μεταξύ αριθμών και μεταβλητών.

Προσοχή : Οι τελεστές "**=**" και "**=:=**" είναι διαφορετικοί. Η κλήση **X=Y** προκαλεί ενοποίηση των αντικειμένων X και Y και εφόσον η ενοποίηση είναι δυνατή, οι μεταβλητές X και Y παίρνουν τις κατάλληλες τιμές. Δεν γίνεται, δηλαδή, κανενός

είδους υπολογισμός. Αντίθετα στην έκφραση **X=:=Y** εκτελείται αριθμητικός υπολογισμός. Επίσης υπάρχει και το αντίθετο κατηγόρημα "\=", το οποίο αληθεύει αν οι δύο όροι ***δεν μπορούν*** να ενοποιηθούν.

Εκτός των κατηγορημάτων ενοποίησης υπάρχουν και τα κατηγορήματα σύγκρισης όρων "**==**"και "**\==**". Το πρώτο κατηγόρημα αληθεύει αν οι δύο όροι που συγκρίνει είναι ταυτόσημοι, δηλαδή έχουν την ίδια δομή και όλοι οι όροι που περιέχουν (αν είναι σύνθετοι) είναι ίδιοι.

Για να κατανοήσετε τη διαφορά των παραπάνω τελεστών κάντε τις παρακάτω ερωτήσεις :

α) 1+2 =:= 2+1. **β)** 1+2 = 2+1. **γ)** 1+A = B+2. **δ)** X\=Y. **ε)** X=5. **στ)**X\=7. **ζ)** X==Y. **η)**X\==Y. **θ)**X==5. **ι)** X\==15. **κ)** 3+7 =\= 4+6

ΑΣΚΗΣΗ

Δίνεται η παρακάτω σχέση:

```
minimum(X,Y,X) :- X<Y.
minimum(X,Y,Y) :- X>=Y.
```

Να κατανοήσετε τη λειτουργία της διατυπώνοντας τις εξής ερωτήσεις στην PROLOG και χρησιμοποιώντας την εντολή trace.

i. minimum(2, 3, X).
ii. minimum(3, 2, X).

Ασκήσεις για εξάσκηση πάνω στην ύλη του εργαστηρίου

1. Να γράψετε έναν κανόνα, ο οποίος θα υπολογίζει το παραγοντικό ενός φυσικού αριθμού N με τη χρήση των ενσωματωμένων πράξεων, γνωρίζοντας ότι 0!=1 και n!=1*2*3*…*n.

2. Να γράψετε έναν κανόνα, για τη σχέση count_till / 2 , η οποία επιστρέφει στο δεύτερο όρισμα, το άθροισμα των αριθμών από το 1 μέχρι το πρώτο όρισμα.

3. Έστω ο παρακάτω κανόνας:

   add_3_and_double(X,Y):- Y is (X+3)*2.

   Να διατυπώσετε δύο ερωτήσεις με βάση τον παραπάνω κανόνα και να κατανοήσετε τη χρήση του. Στη συνέχεια να διατυπώσετε τον κανόνα double_and_five(X, Y, Z), όπου για δύο αριθμούς X, Y θα επιστρέφει στην μεταβλητή Z την τιμή 2*(X+Y)+5.

4. Να ορισθεί το κατηγόρημα plus2 (X,Y,Z) όπου για δύο αριθμούς X, Y θα επιστρέφει στην μεταβλητή Z την τιμή X+Y. Ποια είναι η διαφορά του κατηγορήματος plus2 /3 έναντι του plus1 /3 (το οποίο ορίζεται στο ΜΕΡΟΣ Ι του φυλλαδίου);

5. Ο ορισμός της ακολουθίας **Fibonacci** : f(0)=1, f(1)=1, … f(n) = f(n-1)+f(n-2). Ακολουθία **Fibonacci** : 1,1,2,3,5,8,13, ….

Δίνεται ο N-στος όρος X της ακολουθίας **Fibonacci** : fib(N,X) από την παρακάτω σχέση:

```
fibonacci(0,1).
fibonacci(1,1).
fibonacci(N,X) :- M1 is N-1,
                  M2 is N-2,
                  fibonacci(M1,X1),
                  fibonacci(M2,X2),
                  X is X1+X2.
```

Να επαληθεύσετε ότι η παραπάνω σχέση περιγράφει σωστά την ακολουθία Fibonacci διατυπώνοντας τις κατάλληλες ερωτήσεις και χρησιμοποιώντας την εντολή trace.