

ΑΣΚΗΣΗ 2

Ένα γεγονός είναι μια λίστα από σύμβολα τα οποία περικλείονται σε παρενθέσεις και εκφράζει κάποια πληροφορία σχετική με την κατάσταση του προβλήματος που μας απασχολεί. Τα γεγονότα διακρίνονται σε δύο κατηγορίες: τα **Διατεταγμένα Γεγονότα** και τα **Μη Διατεταγμένα Γεγονότα**. Στόχος της δεύτερης εργαστηριακής άσκησης είναι η χρήση Διατεταγμένων Γεγονότων.

Διατεταγμένα Γεγονότα στο CLIPS

- Βρείτε τι εκφράζουν τα παρακάτω διατεταγμένα γεγονότα:
1. (best-friends Nikos Petros Vasilhs)
 2. (novels "The Picture of Dorian Gray" "Crime and Punishment")
 3. (parents-children Paul Mary Jim Helen Erica)
 4. (lucky-numbers 20 23 9 7 42)
 5. (bus-ticket 0,85)
 6. (best-friends Petros Nikos Vasilhs)
 7. (parents-children Jim Mary Paul Helen Erica)
 8. (musicians "Bob Dylan" "Roger Waters" "David Gilmour")
 9. (cat Tom black)
 10. (Maria Papadopoulou 42 "T234657" 1,65 teacher Athens)

Κάθε γεγονός, όπως φαίνεται κι εδώ, είναι μια λίστα από πεδία (**multifield**) και εκφράζει μια συσχέτιση ή μια κατάσταση. Τα πεδία (**fields**) ενός γεγονότος παίρνουν τιμές διαφόρων τύπων (float, integer, symbol, string κλπ) και διαχωρίζονται μεταξύ τους με κενά. Ο τύπος ενός πεδίου καθορίζεται από τον τύπο της τιμής που αποθηκεύεται σε αυτό. Για παράδειγμα, το γεγονός (duck) είναι γεγονός ενός πεδίου, ενώ το γεγονός (cat Tom black) έχει τρία πεδία. Παρατηρήστε ότι **τα πεδία στα Διατεταγμένα Γεγονότα δεν έχουν κάποιο όνομα**. Φαίνεται μόνο η τιμή που παίρνουν. Ονοματισμένα πεδία χρησιμοποιούνται μόνο σε Μη Διατεταγμένα Γεγονότα όπως θα δούμε σε επόμενη εργαστηριακή άσκηση.

Παρατηρήστε τα γεγονότα 3 και 7. Τα γεγονότα αυτά περιέχουν τις ίδιες ακριβώς τιμές στα διάφορα πεδία τους. Αυτό τα κάνει ισοδύναμα; Γιατί; Τι είναι καθοριστικό για τη σημασία ενός πεδίου σε ένα Διατεταγμένο Γεγονός (όπως δηλώνει και το όνομά του); Τα γεγονότα αυτά εκφράζουν την ίδια πληροφορία; Τα γεγονότα 1 και 6 εκφράζουν την ίδια πληροφορία; Είναι

ισοδύναμα; Τι παρατηρείτε σχετικά με το πρώτο πεδίο κάθε γεγονότος; Τι εκφράζει συνήθως;

- Αν για να περιγράψουμε την κατάσταση «Ο Κώστας κυνηγάει τον Αλέξανδρο» ορίσουμε το γεγονός (Kostas Alexandros), τότε το γεγονός (Alexandros Kostas) τι εκφράζει; Παρατηρείστε ξανά στα δύο αυτά γεγονότα πως ενώ έχουν τις ίδιες τιμές δεν είναι ισοδύναμα γιατί τα πεδία στα οποία τοποθετούνται κάθε φορά οι τιμές έχουν διαφορετική σημασία. Έχει υπονοηθεί (έμμεσα) πως το πρώτο πεδίο αφορά αυτόν που κυνηγά και το δεύτερο αυτόν που κυνηγιέται. Γενικά, **μια καλή πρακτική όταν σχηματίζουμε γεγονότα είναι να χρησιμοποιούμε το πρώτο τους πεδίο για να δηλώσουμε (άμεσα) πως σχετίζονται τα πεδία που ακολουθούν.** Πώς θα μπορούσαμε να εφαρμόσουμε αυτήν την πρακτική στα παραπάνω γεγονότα ώστε να είναι πιο προφανές ότι διαφέρουν μεταξύ τους; **Από τη στιγμή που επιλέγουμε μια συγκεκριμένη σημασία για κάθε πεδίο ενός γεγονότος που εκφράζει συγκεκριμένη σχέση - κατάσταση, πρέπει να υιοθετούμε την ίδια σημασία για κάθε αντίστοιχο πεδίο όλων των γεγονότων που εκφράζουν παρόμοια σχέση - κατάσταση.**
- Πώς δηλώνουμε σε ένα γεγονός πως κάποιο πεδίο δεν έχει τιμή; Ποια η διαφορά στα γεγονότα (nickname James nil) και (nickname James);
- Προσπαθήστε να εκφράσετε με ένα σύνολο Διατεταγμένων Γεγονότων καθεμία από τις παρακάτω καταστάσεις (προσέξτε να υπάρχει ομοιομορφία στα γεγονότα που εκφράζουν την ίδια συσχέτιση - κατάσταση). Χρησιμοποιήστε το 'nil' όπου χρειάζεται:
 1. Ο καιρός σήμερα είναι καλός. Χτες ήταν βροχερός. Η Ελένη, η αδερφή του Γιάννη, ταξιδεύει στην Ισπανία με το αεροπλάνο.
 2. Στη μέση της πλατείας βρίσκεται μια εκκλησία. Η πλατεία λέγεται «Πλατεία Ελευθερίας». Στα δεξιά της πλατείας υπάρχει ένα καφενείο. Το όνομα του καφενείου είναι «Ανώνυμο» και στα αριστερά του υπάρχει μια πιτσαρία με όνομα «Notte». Το περίπτερο είναι πίσω από την πιτσαρία.
 3. Η Ζωή είναι φίλη της Μαρίας που είναι φίλη της Κωνσταντίνας. Η Ζωή αντιπαθεί την Κωνσταντίνα. Η Κωνσταντίνα είναι παντρεμένη και έχει δύο παιδιά, επτά και εννέα χρόνων αντίστοιχα. Η Μαρία και η Ζωή δεν έχουν παιδιά. Η Μαρία είναι παντρεμένη.
 4. Η πάπια είναι ζώο. Ο Donald είναι μια λευκή πάπια που κολυμπάει στην λίμνη. Η λίμνη είναι μικρή και δεν έχει όνομα. Ο σκύλος είναι ζώο. Ο Jack είναι ένας ασπρόμαυρος σκύλος που φυλάει την λίμνη.
- Ένα γεγονός εισάγεται στην λίστα γεγονότων του CLIPS με την εντολή **assert**. Όταν φορτώνεται ένα γεγονός στην μνήμη αποκτά αυτόματα έναν προδιοριστή (**fact identifier**) ο οποίος βοηθάει μετά να αναφερόμαστε στο συγκεκριμένο γεγονός με μοναδικό τρόπο. Οι fact identifiers προκύπτουν από μια αύξουσα ακολουθία ακέραιων αριθμών και ανατίθενται στα γεγονότα με τη σειρά που αυτά εισάγονται στη λίστα γεγονότων. Ο αριθμός που ανατίθεται σε ένα γεγονός (**fact index**) τοποθετείται μετά από τους

χαρακτήρες ‘f-’ για να μας δώσει τον fact identifier του συγκεκριμένου γεγονότος. Εισάγετε στο CLIPS τα παρακάτω γεγονότα και παρατηρήστε τι επιστρέφεται για κάθε assert που κάνετε. Έχετε ανοιχτό το παράθυρο Facts και παρατηρείτε τα γεγονότα που εισάγονται μαζί με τους προσδιοριστές τους:

1. (duck)
2. (cat Tom black)
3. (best-friends Petros Nikos Vasilhs)
4. (weather today good)
5. (Maria Papadopoulou 42 “T234657” 1,65 teacher Athens)
6. (hunts Kostas Alexandros)
7. (nickname James Sawyer)

- Για να διαγράψουμε ένα γεγονός από την λίστα γεγονότων του CLIPS χρησιμοποιούμε την εντολή **retract**. Δοκιμάστε να διαγράψετε 2 γεγονότα από αυτά που βρίσκονται στην λίστα γεγονότων και σιγουρευτείτε ότι διαγράφηκαν. Τι είναι απαραίτητο να γνωρίζουμε προκειμένου να διαγράψουμε ένα γεγονός από την μνήμη; Όταν διαγράφεται ένα γεγονός επηρεάζονται οι fact identifiers των υπόλοιπων γεγονότων; Ποια είναι η βασική διαφορά των εντολών assert και retract, που αφορούν και οι δύο γεγονότα, όσον αφορά την σύνταξή τους; Διαγράψτε 3 γεγονότα από αυτά που έμειναν ταυτόχρονα, χρησιμοποιώντας μία μόνο εντολή retract.
- Απαντήστε στις παρακάτω ερωτήσεις:
 1. Με ποιόν τρόπο μπορούμε να επαναφέρουμε το fact index στο μηδέν; Εφαρμόστε τον.
 2. Αν δώσουμε την εντολή (reset) και μετά εισάγουμε το (weather today rainy), ποιος θα είναι ο fact identifier για το συγκεκριμένο γεγονός; Επιβεβαιώστε.
 3. Αν ξέρουμε πως ο fact identifier του πιο πρόσφατου γεγονότος που εισήχθη στη λίστα γεγονότων του CLIPS είναι f-30, πόσα είναι τα γεγονότα που έχουμε εισάγει μέχρι τώρα στην λίστα γεγονότων;
 4. Μπορούμε να ξέρουμε με βάση αυτή την πληροφορία τον αριθμό των γεγονότων που υπάρχουν μέσα στη μνήμη κάποια χρονική στιγμή; Ποια είναι η απαραίτητη προϋπόθεση για να μπορούμε;
 5. Τα γεγονότα (guests maria katia) και (guests Maria Katia) είναι ισοδύναμα;
- Καθαρίστε την λίστα γεγονότων του CLIPS και εισάγετε τα παρακάτω γεγονότα:
 - (person Georgia Papadopoulou)
 - (person Alexis Alexopoulos)
 - (person Mixalis Anagnwstoy)

Έστω πως τυχαίνει να υπάρχει άλλο ένα άτομο με ονοματεπώνυμο Γεωργία Παπαδοπούλου. Προσπαθήστε να προσθέσετε ένα δεύτερο γεγονός (person Georgia Papadopoulou) στην λίστα γεγονότων και μετά δώστε την εντολή (facts). Τι παρατηρείτε; Υπάρχουν όλα τα άτομα στην λίστα γεγονότων; Δώστε την εντολή (**set-fact-duplication TRUE**) και ξαναπροσπαθήστε να εισάγετε το γεγονός (person Georgia Papadopoulou). Τι γίνεται τώρα αν δώσετε την εντολή (facts); Αν επαναλάβετε την διαδικασία πάλι αφού δώσετε την εντολή (**set-fact-duplication FALSE**) τι θα συμβεί; Τι κάνει η εντολή set-fact-duplication;

- Στο παράθυρο Facts εξακολουθούν να βρίσκονται τα γεγονότα που καταχωρήσατε στο προηγούμενο ερώτημα. Παρατηρήστε τον fact identifier του τελευταίου (πιο πρόσφατου) εισαχθέντος γεγονότος. Στην συνέχεια δώστε την εντολή (retract *). Ποιο είναι το αποτέλεσμα της; Μοιάζει με το αποτέλεσμα που θα είχε η εντολή (clear); Τώρα με χρήση του πάνω βέλους στο πληκτρολόγιο (που επαναφέρει προηγούμενες εντολές που έχουμε πληκτρολογήσει μετά το prompt του CLIPS) εισάγετε ξανά ένα-ένα τα προηγούμενα γεγονότα. Τι παρατηρείτε σχετικά με τους fact identifiers των γεγονότων που εισάγετε μετά το (retract *); Τώρα δώστε την εντολή (clear) και επαναλάβετε την ίδια διαδικασία. Τι παρατηρείτε αυτή τη φορά σχετικά με τους fact identifiers; Ποια η διαφορά του (clear) από το (retract *) όσον αφορά τα γεγονότα;
- Πληκτρολογήστε τα παρακάτω και παρατηρήστε τα αποτελέσματά τους:
 1. (clear)
 2. (assert (lives John Patras) (lives Mary Athens) (lives Paul London) (lives Helen Paris) (lives Juan Spain))
 3. (facts)
 4. (facts 0)
 5. (facts 1)
 6. (facts 2)
 7. (facts 3)
 8. (facts 0 4)
 9. (facts 1 3)
 10. (facts 0 4 2)
 11. (facts 2 5 1)

Τι κάνει η δεύτερη εντολή και τι επιστρέφει; Ποιος είναι ο ρόλος των ορισμάτων στην συνάρτηση (facts), όταν αυτά είναι ένα, δύο ή τρία αντιστοιχία;

- Το εργαλείο CLIPS παρέχει κάποιες εντολές που βοηθούν στην απόσφαλμάτωση προγραμμάτων. Υπάρχει μία εντολή που μας επιτρέπει να παρακολουθούμε διαρκώς τα γεγονότα που εισάγονται ή διαγράφονται από την λίστα γεγονότων. Η εντολή αυτή είναι η (**watch facts**) και μας βγάζει από τον κόπο να πληκτρολογούμε την εντολή (facts) ξανά και ξανά, μετά από κάθε μας κίνηση. Δώστε την εντολή (watch facts) μετά το prompt του CLIPS και στη συνέχεια παρατηρήστε την λειτουργία της εισάγοντας τα γεγονότα 1-4 και διαγράφοντας μετά από αυτά τα 2 και 3:

1. (animal duck)
2. (animal dog)
3. (furniture chair)
4. (furniture table)

Τι σημαίνει το σύμβολο ‘=>’ και τι το σύμβολο ‘<=’; Με ποια εντολή μπορούμε να τερματίσουμε την παρακολούθηση των γεγονότων;

- Ένας εναλλακτικός τρόπος εισαγωγής γεγονότων στην λίστα γεγονότων είναι ο ορισμός **deffacts**. Καθαρίστε το CLIPS με (clear). Έστω πως θέλουμε να εκτελέσουμε ένα πρόγραμμα που επιλέγει από ένα σύνολο διαθέσιμων χρωμάτων το χρώμα που μας ταιριάζει καλύτερα με βάση κάποιες απαντήσεις που θα δώσουμε. Πριν εκτελέσουμε το πρόγραμμα χρειάζεται να εισάγουμε κάποια γεγονότα που δηλώνουν ποια χρώματα είναι διαθέσιμα. Με ανοιχτό το Facts Window κάνετε τα εξής:

- (assert (color red))
- (assert (color green))
- (assert (color blue))
- (assert (color black))
- (assert (color pink))
- (assert (color white))

Παρατηρήστε πως κάθε γεγονός εισάγεται άμεσα στην λίστα γεγονότων μετά από assert. Αφού φορτώθηκαν τα απαραίτητα γεγονότα στην μνήμη μπορούμε να εκτελέσουμε το πρόγραμμά μας. Έστω πως το πρόγραμμά μας για κάθε χρώμα που απορρίπτει, διαγράφει το αντίστοιχο γεγονός. Στο τέλος μένει μόνο ένα γεγονός που εκφράζει το χρώμα το οποίο προτείνεται. Αν στην συνέχεια θέλουμε να εκτελέσουμε ξανά το πρόγραμμά μας, είμαστε υποχρεωμένοι να εισάγουμε ξανά όλα τα προηγούμενα γεγονότα ένα-ένα, κάτι χρονοβόρο και κοπιαστικό κυρίως αν αναλογιστούμε την περίπτωση προγράμματος του οποίου η αρχική κατάσταση απαιτεί μεγάλο αριθμό γεγονότων με αρκετά πεδία το καθένα. Δώστε την εντολή (reset). Ποιο είναι το αποτέλεσμά της όσον αφορά τα γεγονότα που εισήχθησαν με assert;

Καθαρίστε το περιβάλλον CLIPS με (clear) και γράψτε τα εξής:

- (deffacts colors “insert some colors”
(color red)
(color green)
(color blue)
(color black)

(color pink)
(color white))

Ορίσαμε τώρα μέσω deffacts μια ομάδα γεγονότων που ονομάσαμε ‘colors’ και περιέχει όλα τα γεγονότα που χρειάζεται το πρόγραμμα μας. Παρατηρείτε εισαγωγή γεγονότων μετά την εντολή deffacts; Προκαλεί κάποια αλλαγή στη λίστα γεγονότων όπως προκαλεί μια εντολή assert; Γιατί συμβαίνει αυτό; Δοκιμάστε να κάνετε το παρακάτω:

- (reset)

Ποιο είναι το αποτέλεσμα της (reset) όσον αφορά τα γεγονότα που ορίσαμε μέσω deffacts; Διαγράψτε τώρα όλα τα γεγονότα που βρίσκονται στην λίστα γεγονότων εκτός από ένα, μέσω retract. Στην συνέχεια εκτελέστε ξανά (reset). Τι παρατηρείτε; Τι αρκεί να κάνουμε κάθε φορά που θέλουμε να εκτελέσουμε ένα πρόγραμμα του οποίου η αρχική κατάσταση ορίζεται μέσω deffacts;

Η εντολή (reset) κάνει τις εξής 3 βασικές λειτουργίες:

1. Διαγράφει όλα τα γεγονότα από τη λίστα γεγονότων.
2. Εισάγει το γεγονός (initial-fact) στην λίστα γεγονότων και
3. Εισάγει τα γεγονότα που ορίζονται μέσω deffacts στη λίστα γεγονότων.

Το γεγονός (initial-fact) χρησιμεύει στην εκκίνηση της εκτέλεσης ενός προγράμματος CLIPS, όπως θα εξηγηθεί κι αργότερα, μαζί με τους κανόνες.

➤ Δοκιμάστε τώρα να ορίσετε και μια ακόμα δομή deffacts στο CLIPS:

- (deffacts persons “insert some persons”
(person Panagiotis)
(person Kostas)
(person Eleni)
(person Anastasia)
(person Eley8eria)
(person Eytyxia))

Έστω τώρα πως θέλουμε να δούμε με κάποιο τρόπο τους ορισμούς που γράψαμε για να ελέγξουμε αν είναι σωστοί. Αυτό μπορεί να γίνει με δύο τρόπους:

a. Μέσα στο menu **Browse** (όπου γίνεται διαχείριση των ορισμών που έχουμε κάνει στο CLIPS), επιλέγουμε **Deffacts Manager...** και ανοίγουμε το κατάλληλο παράθυρο, στο οποίο εμφανίζονται σε λίστα τα ονόματα των δομών deffacts που έχουμε ορίσει. Επιλέγοντας μία από αυτές, μας δίνεται η δυνατότητα να την εκτυπώσουμε στην οθόνη όπως είναι γραμμένη ή να την διαγράψουμε από το CLIPS. Όταν έχουμε τελειώσει ότι θέλουμε να κάνουμε βγαίνουμε από την διαχείριση deffacts επιλέγοντας **Done**. Χρησιμοποιήστε τον Deffacts Manager για να δείτε τους ορισμούς των δομών deffacts colors και persons.

b. Εναλλακτικά μπορούμε να δώσουμε οι ίδιοι την εντολή (**ppdeffacts <name>**) μετά το prompt στο Dialog Window για να εμφανιστεί στην οθόνη ο κώδικας ενός ορισμού deffacts. Χρησιμοποιήστε την εντολή ppdeffacts για να δείτε τους ορισμούς των δομών deffacts colors και persons.

- Το CLIPS υποστηρίζει τμηματική (modular) ανάπτυξη βάσεων γνώσης και τμηματική εκτέλεση κανόνων μέσω της εντολής **defmodule**. Τα τμήματα (**modules**) CLIPS επιτρέπουν σε ένα σύνολο ορισμών να ομαδοποιηθούν με τέτοιο τρόπο ώστε κάθε φορά να εστιάζεται ο έλεγχος και η εκτέλεση σε ένα μόνο module αποκλείοντας την πρόσβαση σε ορισμούς-δομές άλλων modules. Ο μόνος τρόπος για να διαγράψουμε ένα module είναι μέσω της εντολής (clear). Γενικά, από τη στιγμή που ορίζετε ένα module δεν μπορείτε να το ορίσετε ξανά ούτε να το σβήσετε. Το CLIPS με την εκκίνηση αλλά και μετά από εντολή (clear) αυτόματα ορίζει το module **MAIN** που είναι και το εξ' ορισμού module στο οποίο ανήκουν όλα τα γεγονότα, κανόνες και συναρτήσεις που γράφουμε, εκτός κι αν δηλώσουμε ρητά πως ανήκουν σε άλλο module το οποίο πρέπει και να ορίσουμε πρώτα. Εκτελέστε τις παρακάτω εντολές και παρατηρήστε τα αποτελέσματά τους:

1. (clear)
2. (deffacts facts (a) (b) (c))
3. (get-current-module)
4. (list-deffacts)

5. (defmodule A)
6. (deffacts names (John) (Sayid) (Sawyer))
7. (get-current-module)
8. (list-deffacts)

9. (set-current-module MAIN)
10. (list deffacts)

11. (defmodule B)
12. (deffacts rest-names (Kate) (Jack) (Charly))
13. (get-current-module)
14. (list-deffacts)

15. (set-current-module A)
16. (list-deffacts)

17. (list-defmodules)
18. (clear)
19. (list-defmodules)

20. (defmodule C)
21. (defmodule D)
22. (defmodule E)
23. (deffacts light-colors (white) (yellow))
24. (deffacts dark-colors (black) (brown))
25. (get-current-module)

26. (list-deffacts)
27. (deffacts C::men (Antonis) (Alexis))
28. (get-current-module)
29. (list-deffacts)
30. (deffacts D::women (Alexia) (Alik))
31. (get-current-module)
32. (list-deffacts)
33. (list-defmodules)

Τι κάνουν οι εντολές **get-current-module** και **set-current-module** και τι επιστρέφουν; Όταν ορίζουμε ένα νέο module, έστω με όνομα *Module*, και ορίζουμε μια δομή deffacts αμέσως μετά, έστω με όνομα *name*, σε ποιο module θα ανήκει αυτή η δομή; Αν θέλουμε η δομή με όνομα *name* να ανήκει σε ένα διαφορετικό module από το *Module*, έστω *Module2*, ποιου είναι οι δύο εναλλακτικοί τρόποι να το καταφέρουμε αυτό; Τι κάνει η εντολή **list-defmodules** και τι η εντολή **list-deffacts**;

- Δώστε τις παρακάτω εντολές στο CLIPS, αφού πρώτα δώσετε την εντολή (clear), και παρατηρήστε τα αποτελέσματα στο Facts Window και στο Dialog Window:

1. (deffacts 4-letters (a) (b) (c) (d))
2. (deffacts 4-symbols (s1) (s2) (s3) (s4))
3. (deffacts 3-strings ("str1") ("str2") ("str3"))
4. (list-deffacts)
5. (facts)
6. (reset)
7. (facts)
8. (deffacts 4-letters (e) (f) (g) (h))
9. (facts)
10. (reset)
11. (facts)
12. (undeffacts 4-letters)
13. (list-deffacts)
14. (facts)
15. (reset)
16. (facts)

Τι συμβαίνει με μια υπάρχουσα δομή deffacts όταν ορίζουμε μια άλλη δομή deffacts με το ίδιο όνομα; Τι κάνει η εντολή **undeffacts** και τι παίρνει σαν όριομα; Παρατηρήστε ξανά τον ρόλο που παίζει η εντολή (reset). Υπάρχει εναλλακτικός τρόπος να κάνουμε αυτό που κάνει η undeffacts χωρίς να γράψουμε οι ίδιοι την εντολή μετά το prompt στο Dialog Window;

- Καθαρίστε ξανά το περιβάλλον CLIPS και δοκιμάστε να δημιουργήσετε ένα αρχείο .clp με τα παρακάτω:

1. (assert (gift book))
2. (assert (gift record))

Στη συνέχεια σώστε το και προσπαθήστε να το κάνετε Load. Παρατηρήστε το μήνυμα που εμφανίζεται. Μετά ανοίξτε το αρχείο σας, οβήστε τα προηγούμενα και επαναλάβετε την προηγούμενη διαδικασία για τα παρακάτω διαδοχικά.

1. (+ 2 3)
2. (- 4 3)

και

1. (printout t "Hello" crlf)

Τώρα οβήστε ξανά ότι περιέχει το αρχείο σας και γράψτε έναν ορισμό deffacts με όνομα numbers για τα γεγονότα (number 1) (number 2) (number 3) και (number 4). Σώστε τις αλλαγές σας και κάντε Load το περιεχόμενο του αρχείου σας στο CLIPS. Δώστε την εντολή (reset). Να θυμάστε ότι από ένα αρχείο μπορούμε μόνο να φορτώσουμε ορισμούς (*πρότυπα γεγονότων, ορισμούς deffacts, ορισμούς καθολικών μεταβλητών, κανόνες, συναρτήσεις κλπ*) στο περιβάλλον CLIPS, κι όχι να εκτελέσουμε άμεσες εντολές. Θυμίζουμε ξανά πως η εντολή (reset) είναι απαραίτητη για να φορτωθεί το περιεχόμενο ενός αρχείου στο CLIPS.

- Προσπαθήστε να κάνετε assert τα παρακάτω διατεταγμένα γεγονότα:

```
(duck (shot Brian George Ron))  
(has Peter (pet (dog Jack) ) )  
(fact1 sym11 (fact2 sym21 sym22) sym12 sym13)
```

Επιτρέπονται στο CLIPS διατεταγμένα γεγονότα ενσωματωμένα (embedded) μέσα σε άλλα διατεταγμένα γεγονότα;

- Ορίσατε μια καθολική μεταβλητή *?*x** και αναθέστε της το fact index ενός γεγονότος (age Paul 26) το οποίο εισάγετε στην λίστα γεγονότων. Στη συνέχεια διαγράψτε το γεγονός από την λίστα γεγονότων χρησιμοποιώντας την καθολική μεταβλητή *?*x**.
- Κλείστε το CLIPS είτε με την εντολή (**exit**) στο Dialog Window είτε με **File->Exit**.