

# ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

## ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ



**Μάθημα Προπτυχιακών Σπουδών:**

Λογικός Προγραμματισμός, 5<sup>ο</sup> Εξάμηνο

**Δεύτερη Προαιρετική Εργασία**

**Φοιτητής:** Θεόδωρος Κοζάνογλου

**A.M:** Π20094

## Περιεχόμενα

Εκφώνηση Άσκησης: .....	3
Επεξήγηση Κώδικα: .....	4
Κώδικας Άσκησης: .....	4
Παραδείγματα Εκτέλεσης Προγράμματος: .....	4

## Εκφώνηση Άσκησης:

Παράδοση μέσω gunet2 μέχρι 23 Δεκεμβρίου. Παραδοτέο είναι pdf με τεκμηριωμένο κώδικα και παράδειγμα σωστής εκτέλεσης. OXI ZIP, OXI PL!

Να γραφούν τα ακόλουθα κατηγορήματα:

**[Για φοιτητές με επώνυμο από Α έως Ε]** precede\_list(X, Y) που αληθεύει όταν η X λίστα προηγείται της Y λίστας, δηλ. αν η X είναι υπολίστα με κάποια από τα πρώτα από αριστερά στοιχεία της Y. Παράδειγμα:

?-precede\_list([1,2], [1, 2, 3]).

Yes

?-precede\_list([1,3], [1, 2, 3]).

No

?-precede\_list([1], [1, 2, 3]).

Yes

**[Για φοιτητές με επώνυμο από Ζ έως Κ]** proceed\_list(X, Y) που αληθεύει όταν η Y λίστα έπεται της X λίστας, δηλ. αν η Y είναι υπολίστα με κάποια από τα πρώτα από δεξιά στοιχεία της X. Παράδειγμα:

?-proceed\_list([1,2, 3], [1, 3]).

No

?-proceed\_list([1,2], [2]).

Yes

**[Για φοιτητές με επώνυμο από Λ έως Ο]** included\_list(X, Y) που αληθεύει όταν η X είναι μία μη κενή υπο-λίστα της Y λίστας. Παράδειγμα:

?-included\_list([1,2], [0, 1, 2, 3]).

Yes

**[Για φοιτητές με επώνυμο από Π έως Τ]** common\_list(X, Y) που αληθεύει όταν οι δύο λίστες έχουν τουλάχιστον ένα κοινό στοιχείο. Παράδειγμα:

?-common\_list([1,2, 3], [0, 3, 4]).

Yes

**[Για φοιτητές με επώνυμο από Υ έως Ω]** pair\_list(X, Y) που ως Y επιστρέφει σε ζευγάρια τα στοιχεία της X λίστας. Παράδειγμα:

?-pair\_list([1,2, 3, 4, 5, 6], [[1, 2], [3, 4], [5, 6]]).

Yes

?-pair\_list([1,2, 3, 4, 5, 6], X).

X=[[1,2],[3,4],[5,6]]

## Επεξήγηση Κώδικα:

Σύμφωνα με την εκφώνηση έχω δύο λίστες X και Y. Πρέπει να δημιουργήσω τα κατάλληλα γεγονότα και κανόνες ώστε να γίνεται ο έλεγχος αληθείας της εξής περίπτωσης: η Y λίστα που θα εισάγει ο χρήστης στο πρόγραμμα να είναι υπολίστα με κάποια από τα πρώτα δεξιά στοιχεία της λίστας X.

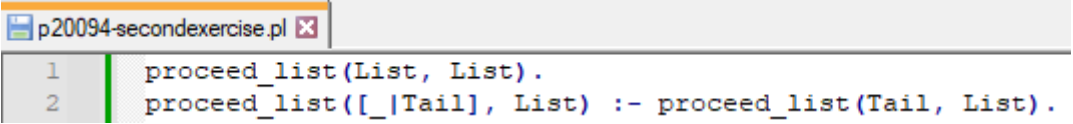
Δημιούργησα **1 γεγονός** και **1 κανόνα**:

- ❖ `proceed_list(List, List).`
  - Αν οι δύο λίστες που εισάγει ο χρήστης είναι ίδιες, τότε το γεγονός είναι **True**.
- ❖ `proceed_list([Head|Tail], List) :- proceed_list(Tail, List).`
  - Αν οι δύο λίστες που εισάγει ο χρήστης **δεν** είναι ίδιες, ο κανόνας αφαιρεί το Head της πρώτης λίστας (**X**) και ξανακαλείται ξανά η **proceed\_list** για γίνει η σύγκριση των λιστών: **Tail της λίστας X** με την λίστα **Y**.

Η διαδικασία επαναλαμβάνεται μέχρι όλες οι υπολίστες της X να έχουν συγκριθεί με την λίστα Y. Αν τελειώσει η επανάληψη και η τελευταία υπολίστα της X **δεν** είναι ίση με την λίστα της Y, τότε το γεγονός **proceed\_list** είναι **False**.

**Σημείωση:** Το **Head** της λίστας X είναι μία αχρησιμοποίητη μεταβλητή, **Singleton**, οπότε την αντικατέστησα με “\_”.

## Κώδικας Άσκησης:



```
p20094-secondexercise.pl
1  proceed_list(List, List).
2  proceed_list([_|Tail], List) :- proceed_list(Tail, List).
```

Εικόνα 1: Στιγμιότυπο Κώδικα

## Παραδείγματα Εκτέλεσης Προγράμματος:

```
Welcome to SWI-Prolog (threaded, 32 bits, version 8.4.3)
SWI-Prolog comes with ABSOLUTELY NO WARRANTY. This is free software.
Please run ?- license. for legal details.

For online help and background, visit https://www.swi-prolog.org
For built-in help, use ?- help(Topic). or ?- apropos(Word).

?- proceed_list([1,2,3],[1,3]).
false.

?- proceed_list([1,3],[1,3]).
true.

?- proceed_list([1,2],[2]).
true.
```

Εικόνα 2: Παραδείγματα Εκτέλεσης Προγράμματος