ΠΑΝΕΠΙΣΤΗΜΙΟ ΙΩΑΝΝΙΝΩΝ

ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ Η/Υ ΚΑΙ ΠΛΗΡΟΦΟΡΙΚΗΣ

MYY-105: Εισαγωγη στον Προγραμματισμο

8ο Εργαστηριο

(ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2014-2015)

Διδάσκων
Νικόλαος Μαμουλής

Υπεύθυνη Εργαστηρίου
Μαρία Γ. Χρόνη

Πλοηγηθείτε στο φάκελο *Python2014* και δημιουργήστε ένα νέο φάκελο *Lab8*. Μέσα στο φάκελο *Lab8* αποθηκεύστε τα αρχεία της Άσκησης, με όνομα *permut.py*.

Άσκηση

Μία λίστα $\Lambda 1$ είναι αναδιάταξη (permutation) μιας λίστας $\Lambda 2$ αν οι $\Lambda 1$ και $\Lambda 2$ περιέχουν τα ίδια στοιχεία σε διαφορετική σειρά. Π.χ., η λίστα [1,2,3] είναι αναδιάταξη της λίστας [2,1,3]. Ορίστε μια αναδρομική συνάρτηση `permutations(lst)`, η οποία θα επιστρέφει όλες τις αναδιατάξεις της λίστας `lst`. Η συνάρτηση πρέπει να επιστρέφει μια λίστα από λίστες. Για παράδειγμα η κλήση `permutations([1,2,3])` πρέπει να επιστρέφει:

$$[[1,2,3],[2,1,3],[2,3,1],[1,3,2],[3,1,2],[3,2,1]]$$

Για να ορίσετε τη συνάρτησή σας, σκεφτείτε ότι:

- Μια λίστα με ένα στοιχείο έχει μόνο μία αναδιάταξη. Π.χ. `permutations([1])=[[1]]` (βασική περίπτωση).
- Για να υπολογίσουμε τις αναδιατάξεις μιας λίστας Λ με n στοιχεία, μπορούμε να υπολογίσουμε τις αναδιατάξεις των τελευταίων $n-1$ στοιχείων της Λ, και για κάθεμια να παρεμβάλλουμε το πρώτο στοιχείο της Λ σε όλες τις θέσεις (αναδρομική περίπτωση). Π.χ. θεωρείστε ότι $\Lambda = [1,2,3]$ (n=3). Μια αναδιάταξη των τελευταίων $n-1$ (=2) στοιχείων είναι η $A = [2,3]$. Με βάση αυτή και το πρώτο στοιχείο της Λ (1), μπορούμε να φτιάξουμε τις αναδιατάξεις [1,2,3], [2,1,3], [2,3,1], δηλαδή να παρεμβάλλουμε το 1 σε όλες τις δυνατές θέσεις στην A (πριν τα [2,3], ανάμεσα στα [2,3], μετά τα [2,3]). Έπειτα πρέπει να κάνουμε το ίδιο και για A=[3,2].

Για να ελέγξετε την ορθότητα της συνάρτησης `permutations`, φτιάξτε ένα πρόγραμμα που την καλεί. Το πρόγραμμα πρέπει να παίρνει από το χρήστη έναν θετικό ακέραιο n (π.χ. n=3), να δημιουργεί μια λίστα $L = [1,2,...,n]$, (π.χ. $L = [1,2,3]$), και να τυπώνει αυτό που επιστρέφει η `permutations(L)`. Σημειώστε ότι μία λίστα μήκους n έχει $n!$ αναδιατάξεις. Οπότε, το `len(permutations(L))` πρέπει να είναι ίσο με $n!$