

# Αναφορά 1ης Άσκησης στο ΠΛΗ211

Ομάδα : LAB21142945

Μέλη: Αντώνιος Εμμανουήλ Μαραγκουδάκης A.M: 2013030093

Δεν βρήκα κάποιον που να μπορώ να συνεργαστώ επομένως υλοποίησα μόνος μου όλη την άσκηση. Ελπίζω να βρω κάποιον συνεργάσιμο. Αν στις επόμενες δύο ασκήσεις υπάρξει υψηλό φόρτο ελπίζω να δείξετε επιείκεια στις μονομελείς ομάδες.!

## Θέμα 1

Ζητούμενο ήταν να γραφτεί ένα πρόγραμμα `regr` σε Bash shell που να δέχεται ως παραμέτρους προγράμματος μία λίστα από αρχεία. Κάθε αρχείο ουσιαστικά περιέχει πολλά διανύσματα ένα σε κάθε σειρά, με μορφή `τιμή_X:τιμή_Y`. Έτσι το πρόγραμμα `regr` υπολογίζει για κάθε τέτοιο αρχείο τις παραμέτρους γραμμικής παλινδρόμησης  $a, b, c$  που ελαχιστοποιούν το συνολικό τετραγωνικό σφάλμα `err` της προσέγγισης  $cY=aX+b$ , καθώς και το `err`.

Στην εκφώνηση δόθηκαν οι μαθηματικοί τύποι του προβλήματος. Στην περίπτωση όπου οι τιμές  $X$  των διανυσμάτων ενός αρχείου ισούνται όλες μεταξύ τους, τότε προκύπτει μία διαίρεση με το μηδέν στους τύπους αυτούς. Έτσι σε αυτήν την περίπτωση η  $cY=aX+b$  ταυτίζεται με την ευθεία  $X=\text{σταθερή}$  ώστε να ελαχιστοποιείτε το `err`. Άρα με  $a=1, c=0$  προκύπτει  $0*Y=1*X+b \Rightarrow b=-X$ . Άρα στην περίπτωση αυτή επέλεξα  $a=1, c=0$  και  $b=-X[1]$  (π.χ της θέσης  $X[1]=X[2]=\dots X[n]$ ). Η ευθεία  $X=\text{σταθερή}$  περνάει πάνω από όλα τα σημεία  $(x:y)$  των διανυσμάτων που περιέχονται στο αρχείο εισόδου και έτσι επέλεξα `err=0`.

Η γενική περίπτωση υπολογίζεται σύμφωνα με το τυπολόγιο της εκφώνησης. Όσον αφορά το προγραμματιστικό κομμάτι σε Bash shell υπάρχει λεπτομερής σχολιασμός στον κώδικα όπου εξηγεί την λειτουργία του.!

Η εκτύπωση γίνεται όπως ζητάτε από την εκφώνηση με δύο δεκαδικά ψηφία! Όπου π.χ  $a=.61$  ή  $a=-.61$  είναι σε Bash ίσα με  $a=0.61$ ,  $a=-0.61$  και τα εκτυπώνω έτσι αφού δεν ζητήθηκε από την εκφώνηση η αλλαγή σε π.χ  $a=0.61$  !

## Ενδεικτικά:

```
antonis@antonis-Lenovo-IdeaPad-Z510: ~/Desktop/LAB21142945
File Edit View Search Terminal Help
antonis@antonis-Lenovo-IdeaPad-Z510:~/Desktop/LAB21142945$ ./regr input1 input2 input3 input4
FILE: input1, a=2.13 b=-24.31 c=1 err=4607.94
FILE: input2, a=0 b=9.51 c=1 err=0
FILE: input3, a=1 b=-6.12 c=0 err=0
FILE: input4, a=-.61 b=4.44 c=1 err=216.31
antonis@antonis-Lenovo-IdeaPad-Z510:~/Desktop/LAB21142945$
```

## Θέμα 2

Ζητούμενο ήταν να γραφτεί ένα πρόγραμμα results σε Bash shell το οποίο θα δέχεται σαν παράμετρο προγράμματος το όνομα ενός αρχείου. Το αρχείο αυτό θα περιέχει γραμμές της μορφής: Ομάδα1-Ομάδα2:Σκορ1-Σκορ2 (πχ Portugal-Greece:1-2). Το πρόγραμμα results πρέπει να υπολογίζει την βαθμολογία όλων των ομάδων που περιέχονται στο αρχείο εισόδου και να τις ταξινομεί με βάση την βαθμολογία, ενώ σε περίπτωση ισοβαθμίας να ταξινομεί τις ισοβαθμες με αλφαβητική σειρά. Επίσης υπολογίζει και εμφανίζει και τα συνολικά γκολ (έβαλε-έφαγε) η κάθε ομάδα.

Έτσι αρχικά ο κώδικας συλλέγει σε πίνακες όλα τα στοιχεία του αρχείου εισόδου και έπειτα κατασκευάζει έναν πίνακα ο οποίος περιέχει δύο σειρές για κάθε σειρά του αρχείου (πχ Portugal-Greece:1-2) δηλαδή για κάθε έναν αγώνα. Έτσι ο πίνακας αυτός θα περιέχει στην περίπτωση αυτή στην 1η θέση την πληροφορία Portugal.0.1.2 . Έτσι γνωρίζουμε τι πόντους πήρε η Πορτογαλία από αυτόν τον αγώνα και πόσα γκολ (έβαλε.έφαγε). Για τον ίδιο αγώνα στην 2η θέση του πίνακα αυτού υπάρχει η ανάλογη πληροφορία για την Ελλάδα, δηλαδή Greece.3.2.1 . Έτσι ξέρουμε ότι ανάμεσα από τις τελείες έχουμε για κάθε ομάδα σε κάθε αγώνα την πληροφορία Όνομα.Πόντοι.Έβαλε.Έφαγε . Στην περίπτωση του φιλοξενούμενου τα γκολ γράφονται με ανάποδη σειρά ώστε να υπάρχουν πάντα πρώτα τα γκολ που έβαλε η ομάδα και έπειτα τα γκολ που έφαγε ( πχ Portugal-Greece:1-2) => Greece.3.2.1 ). Ο πίνακας αυτός στον κώδικα ονομάζεται vathmoi[] και έχει μέγεθος 2\*σειρές του αρχείου εισόδου.

Έπειτα μέσω του πίνακα αυτού κατασκευάζεται ένας νέος πίνακας onomata[k] ο οποίος περιέχει από μια φορά μόνο τα ονόματα που περιέχει ο πίνακας vathmoi[]. Δηλαδή περιέχει από μία φορά τα ονόματα των ομάδων που έχουν αγωνιστεί σύμφωνα με το αρχείο εισόδου.

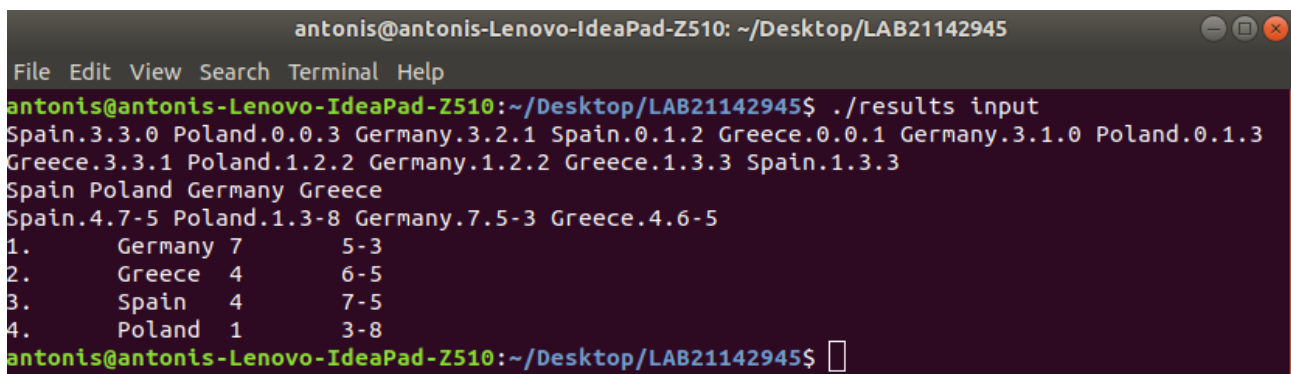
Έπειτα για κάθε τέτοιο όνομα από τον onomata[k] προστίθενται οι βαθμοί , τα γκολ που έβαλε και γκολ που έφαγε σε κάθε αγώνα που συμμετείχε σύμφωνα με τον πίνακα vathmoi[]. Έτσι τελικά κατασκευάζεται ο πίνακας teliko[k] που περιέχει αταξινόμητο το ζητούμενο της εκφώνησης σε μορφή Όνομα.Βαθμοί.Έβαλε-Έφαγε .

Έπειτα γράφεται ο πίνακας αυτός σε αρχείο. Με την χρήση τις εντολής sort ταξινομείται το αρχείο με την σειρά που ζητείται στην εκφώνηση και έπειτα εκτυπώνεται με το φορμάτ που επίσης ζητάτε. Τέλος διαγράφεται το αρχείο αυτό με την εντολή rm.

Όσον αφορά το προγραμματιστικό κομμάτι σε Bash shell υπάρχει λεπτομερής σχολιασμός στον κώδικα όπου εξηγεί την λειτουργία του.! Εκεί μπορείτε προκειμένου να δείτε το περιεχόμενο του κάθε πίνακα , απλά να σβήσετε από σχόλιο καθένα από τα :

```
#echo ${vathmoi[@]}
#echo ${onomata[@]}
#echo ${teliko[@]}
```

### Ενδεικτικά:



```
antonis@antonis-Lenovo-IdeaPad-Z510: ~/Desktop/LAB21142945
File Edit View Search Terminal Help
antonis@antonis-Lenovo-IdeaPad-Z510:~/Desktop/LAB21142945$ ./results input
Spain.3.3.0 Poland.0.0.3 Germany.3.2.1 Spain.0.1.2 Greece.0.0.1 Germany.3.1.0 Poland.0.1.3
Greece.3.3.1 Poland.1.2.2 Germany.1.2.2 Greece.1.3.3 Spain.1.3.3
Spain Poland Germany Greece
Spain.4.7-5 Poland.1.3-8 Germany.7.5-3 Greece.4.6-5
1.      Germany 7      5-3
2.      Greece  4      6-5
3.      Spain   4      7-5
4.      Poland  1      3-8
antonis@antonis-Lenovo-IdeaPad-Z510:~/Desktop/LAB21142945$
```

Για αρχείο εισόδου input :

Spain-Poland:3-0

Germany-Spain:2-1

Greece-Germany:0-1

Poland-Greece:1-3

Poland-Germany:2-2

Greece-Spain:3-3

Έχουμε:

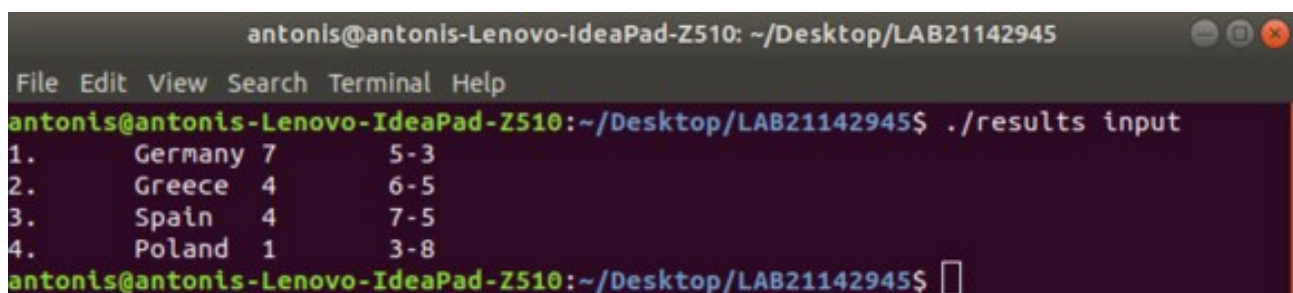
Στις πρώτες δύο σειρές το echo \${vathmoi[@]}

Στην επόμενη το echo \${onomata[@]}

Έπειτα το echo \${teliko[@]}

Και τέλος την εκτύπωση του ταξινομημένου ζητουμένου της εκφώνησης.!

Χωρίς τα echo προφανώς και θα έχουμε :



```
antonis@antonis-Lenovo-IdeaPad-Z510: ~/Desktop/LAB21142945
File Edit View Search Terminal Help
antonis@antonis-Lenovo-IdeaPad-Z510:~/Desktop/LAB21142945$ ./results input
1.      Germany 7      5-3
2.      Greece  4      6-5
3.      Spain   4      7-5
4.      Poland  1      3-8
antonis@antonis-Lenovo-IdeaPad-Z510:~/Desktop/LAB21142945$
```