

# ΕΡΓΑΛΕΙΑ ΑΝΑΠΤΥΞΗΣ ΛΟΓΙΣΜΙΚΟΥ ΚΑΙ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ ΣΥΣΤΗΜΑΤΩΝ

Αναφορά 1ης άσκησης

24/10/2019

Μπουραΐμης Φώτης 2013030067  
Τάκας Παναγιώτης 2013030010

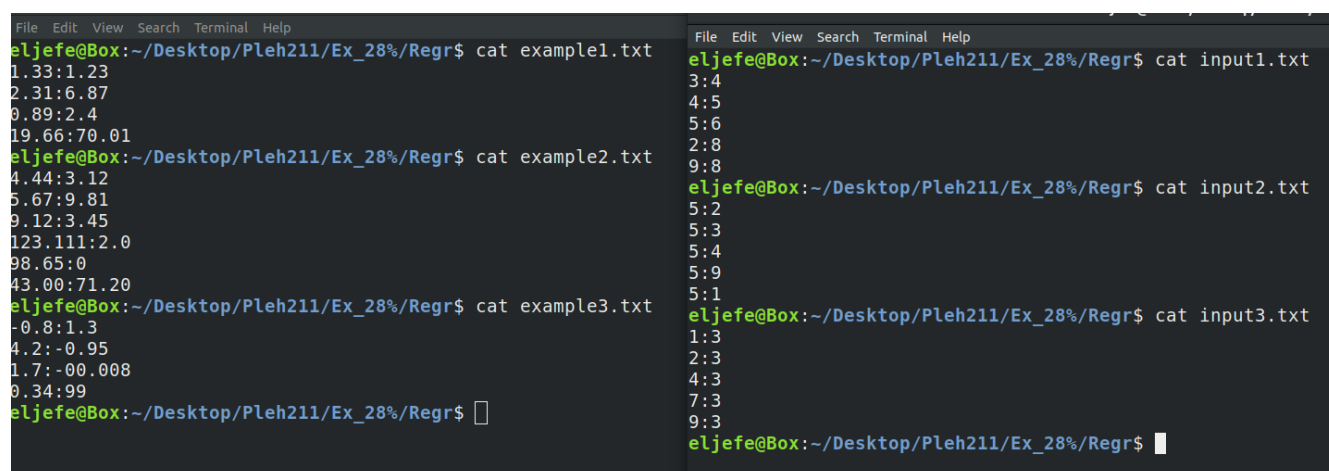
## Εισαγωγή

Στην πρώτη άσκηση του μαθήματος μας ζητήθηκε να υλοποιήσουμε δύο bash scripts. Το κοινό και στα δύο προγράμματα ήταν ότι δέχονταν τουλάχιστον ένα αρχείο και μέσα από string manipulation αποκτούσαν τα δεδομένα έτσι ώστε το καθένα να κάνει κάποιους υπολογισμούς σχετικά με το περιεχόμενο των αρχείων και το ζητούμενο του κάθε θέματος. Η υλοποίηση και των δύο προγραμμάτων έγινε συνεργατικά χωρίς κάποιος να έχει αναλάβει εξ' ολοκλήρου ένα συγκεκριμένο κομμάτι του project.

### Θέμα 1:

Αρχικά φτιάχνουμε μια συνθήκη ελέγχου έτσι ώστε να είμαστε σίγουροι πως ο χρήστης δίνει κάποιους φακέλους σαν ορίσματα στο πρόγραμμα αλλιώς τυπώνουμε μήνυμα ενημέρωσης για τη σωστή χρήση του σε μορφή μηνύματος unix. Έπειτα χρησιμοποιούμε μία επανάληψη για κάθε αρχείο (όρισμα) που δίνεται και άλλη μία εμφωλευμένη επανάληψη με την οποία περνάμε κάθε γραμμή του αρχείου. Στη συνέχεια χρησιμοποιούμε την εντολή **cut** και **bc** για να εξάγουμε τα δεδομένα της κάθε γραμμής αλλά και να κάνουμε πράξεις σύμφωνα με τους πίνακες της εκφώνησης. Τέλος για την εκτύπωση των αποτελεσμάτων εκμεταλλευτήκαμε την εντολή **awk** που μας προσφέρει πιο πολλές δυνατότητες ως προς το format της εξόδου όπως για παράδειγμα την εκτύπωση ενός αριθμού μέχρι το δεύτερο δεκαδικό του ψηφίο, το οποίο είναι και ζητούμενο στην άσκηση. Παρακάτω εμφανίζονται στην εικόνα α) κάποια αρχεία στα οποία δοκιμάσαμε το script μας ενώ στην εικόνα β) τα αποτελέσματα αυτών. Τα αρχεία αυτά περιέχουν φυσικούς, δεκαδικούς και αρνητικούς αριθμούς αλλά και περιπτώσεις όπως σταθερό διάνυσμα X και Y. Στην προκειμένη άσκηση κρίναμε ότι δεν χρειάζεται να παραθέσουμε κάποιο κομμάτι κώδικα αφού δεν υπάρχει μεγάλη πολυπλοκότητα και τα σχόλια μέσα στο πρόγραμμα είναι αρκετά επεξηγηματικά.

### Εικόνα α)



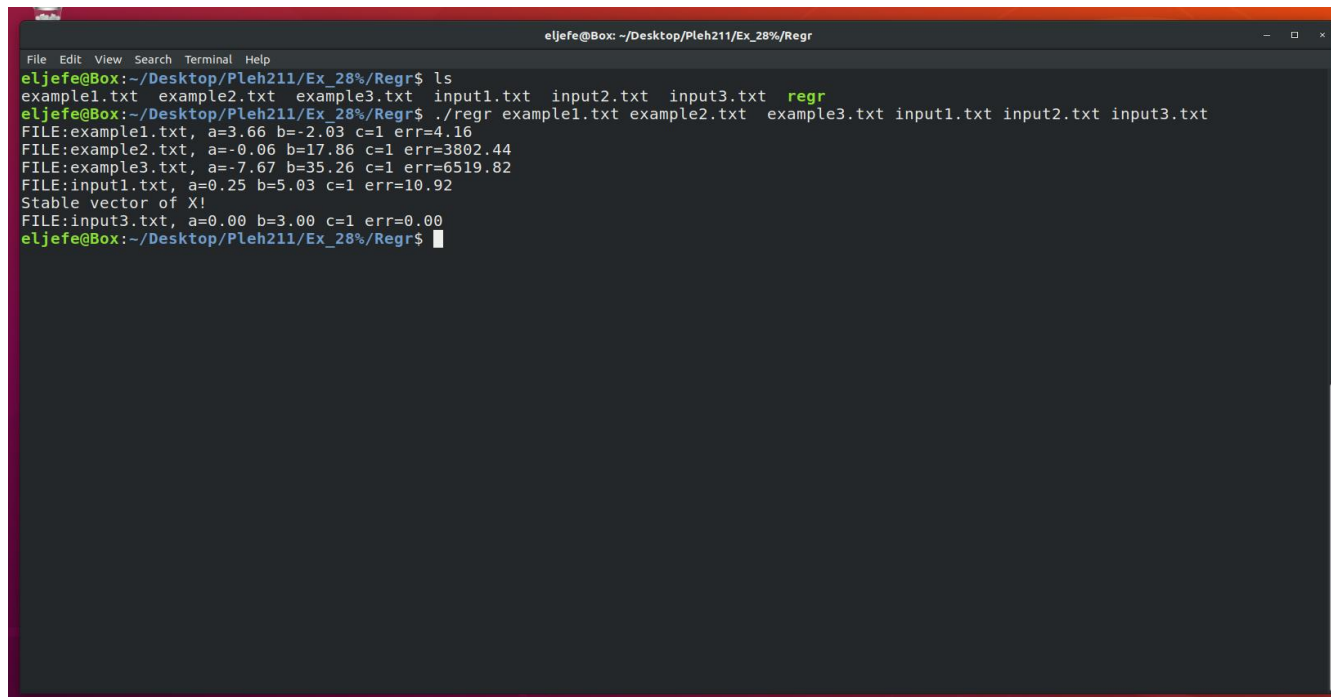
The image shows two terminal windows side-by-side. The left window shows the contents of three files: example1.txt, example2.txt, and example3.txt. The right window shows the execution of the same three files using the 'cat' command, displaying the output for each.

```
File Edit View Search Terminal Help
e1jefe@Box:~/Desktop/Pleh211/Ex_28%/Regr$ cat example1.txt
1.33:1.23
2.31:6.87
0.89:2.4
19.66:70.01
e1jefe@Box:~/Desktop/Pleh211/Ex_28%/Regr$ cat example2.txt
4.44:3.12
5.67:9.81
9.12:3.45
123.111:2.0
98.65:0
43.00:71.20
e1jefe@Box:~/Desktop/Pleh211/Ex_28%/Regr$ cat example3.txt
-0.8:1.3
4.2:-0.95
1.7:-00.008
0.34:99
e1jefe@Box:~/Desktop/Pleh211/Ex_28%/Regr$

File Edit View Search Terminal Help
e1jefe@Box:~/Desktop/Pleh211/Ex_28%/Regr$ cat input1.txt
3:4
4:5
5:6
2:8
9:8
e1jefe@Box:~/Desktop/Pleh211/Ex_28%/Regr$ cat input2.txt
5:2
5:3
5:4
5:9
5:1
e1jefe@Box:~/Desktop/Pleh211/Ex_28%/Regr$ cat input3.txt
1:3
2:3
4:3
7:3
9:3
e1jefe@Box:~/Desktop/Pleh211/Ex_28%/Regr$
```



## Εικόνα β)

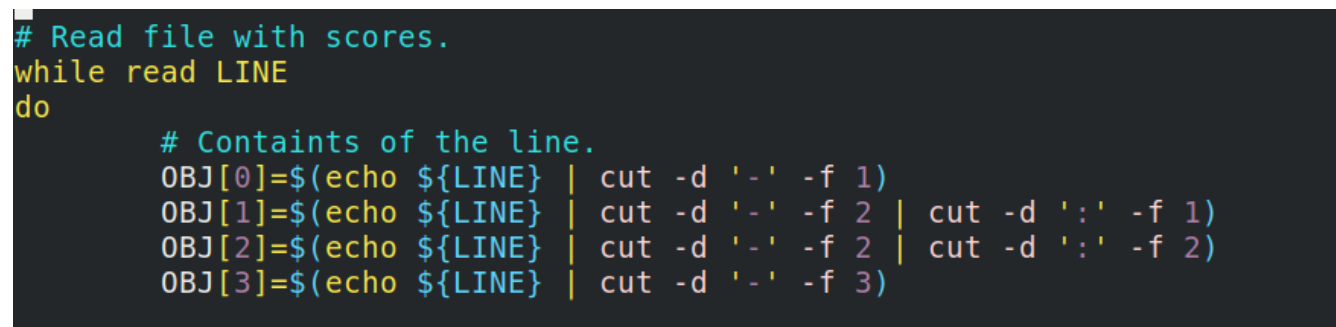


```
eljefe@Box: ~/Desktop/Pluh211/Ex_28%/Regr
eljefe@Box:~/Desktop/Pluh211/Ex_28%/Regr$ ls
example1.txt example2.txt example3.txt input1.txt input2.txt input3.txt regr
eljefe@Box:~/Desktop/Pluh211/Ex_28%/Regr$ ./regr example1.txt example2.txt example3.txt input1.txt input2.txt input3.txt
FILE:example1.txt, a=3.66 b=-2.03 c=1 err=4.16
FILE:example2.txt, a=-0.06 b=17.86 c=1 err=3802.44
FILE:example3.txt, a=-7.67 b=35.26 c=1 err=6519.82
FILE:input1.txt, a=0.25 b=5.03 c=1 err=10.92
Stable vector of X!
FILE:input3.txt, a=0.00 b=3.00 c=1 err=0.00
eljefe@Box:~/Desktop/Pluh211/Ex_28%/Regr$
```

## Θέμα 2:

Σε αυτήν την άσκηση εν γένει χρειαστήκαμε πίνακες έτσι ώστε να αποφύγουμε πολλά περάσματα πάνω στο αρχείο όπως επίσης και ένα συσχετισμένο πίνακα αλλά και ένα προσωρινό αρχείο. Όπως και προηγουμένως βεβαιωνόμαστε ότι ο χρήστης δίνει το αρχείο για τη σωστή λειτουργία του προγράμματος. Εν συνεχεία μπαίνουμε στην επανάληψη που διαβάζει την κάθε γραμμή και η πρώτη μας ενέργεια είναι να πάρουμε με **cut** και **pipes** τα 4 δεδομένα Ομάδα1 Ομάδα2 Σκορ1 Σκορ2 όπως φαίνεται παρακάτω.

## Εικόνα γ)

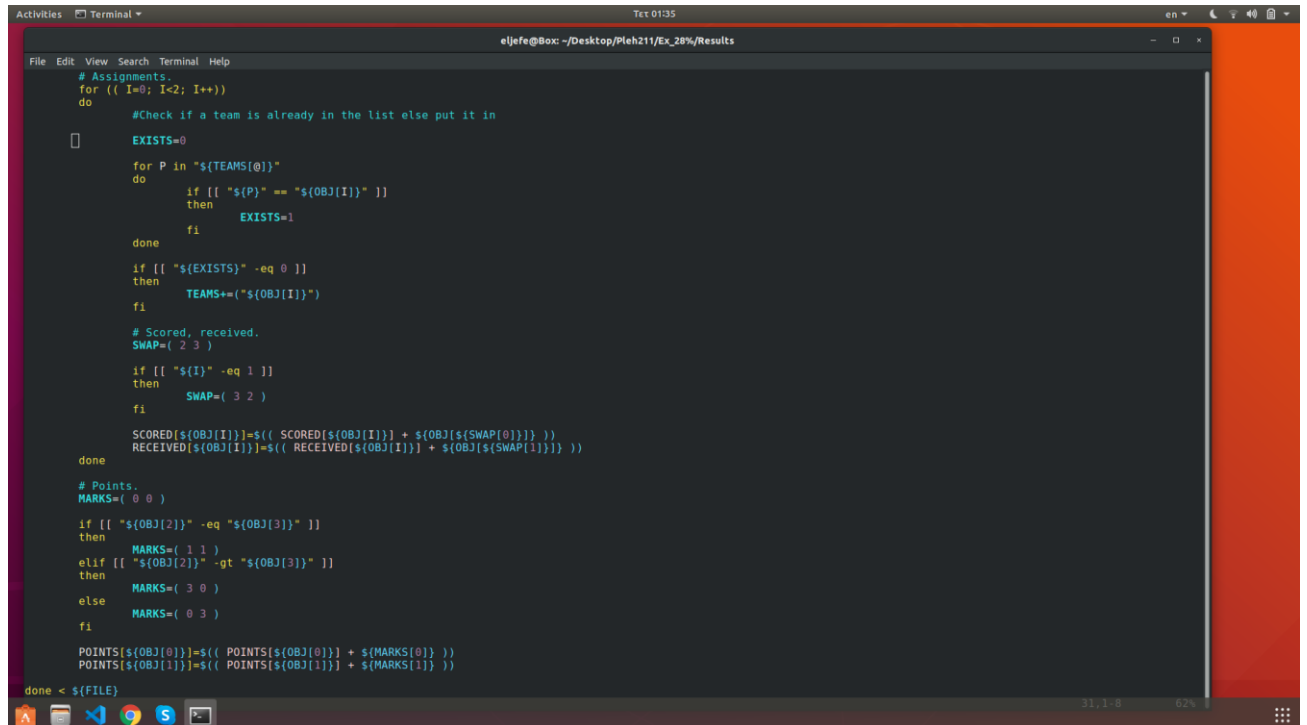


```
# Read file with scores.
while read LINE
do
    # Contents of the line.
    OBJ[0]=$(echo ${LINE} | cut -d '-' -f 1)
    OBJ[1]=$(echo ${LINE} | cut -d '-' -f 2 | cut -d ':' -f 1)
    OBJ[2]=$(echo ${LINE} | cut -d '-' -f 2 | cut -d ':' -f 2)
    OBJ[3]=$(echo ${LINE} | cut -d '-' -f 3)
```



Αφού τελειώσει αυτή η διαδικασία και ενώ είμαστε ακόμα μέσα στην πρώτη επανάληψη ελέγχουμε αν η τρέχουσες ομάδες υπάρχουν μέσα στον αντίστοιχο πίνακα, αν όχι τις προσθέτουμε και συνεχίζουμε ενημερώνοντας τα υπέρ και τα κατά γκολ της κάθε μίας εναλλάξ. Τέλος ενημερώνουμε και το σύστημα των πόντων για την κάθε ομάδα.

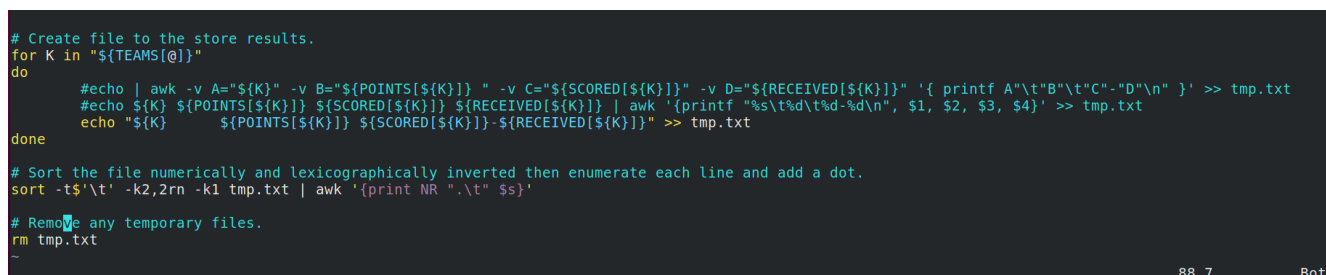
## Εικόνα δ)



```
File Edit View Search Terminal Help
# Assignments.
for (( I=0; I<2; I++ ))
do
    #Check if a team is already in the list else put it in
    EXISTS=0
    for P in "${TEAMS[@]}"
    do
        if [[ "${P}" == "${OBJ[I]}" ]]
        then
            EXISTS=1
        fi
    done
    if [[ "${EXISTS}" -eq 0 ]]
    then
        TEAMS+=("${OBJ[I]}")
    fi
    # Scored, received.
    SWAP=( 2 3 )
    if [[ "${I}" -eq 1 ]]
    then
        SWAP=( 3 2 )
    fi
    SCORED[${OBJ[I]}]=${SCORED[${OBJ[I]}]} + ${OBJ[${SWAP[0]}]}
    RECEIVED[${OBJ[I]}]=${RECEIVED[${OBJ[I]}]} + ${OBJ[${SWAP[1]}]}
done
# Points.
MARKS=( 0 0 )
if [[ "${OBJ[2]}" -eq "${OBJ[3]}" ]]
then
    MARKS=( 1 1 )
elif [[ "${OBJ[2]}" -gt "${OBJ[3]}" ]]
then
    MARKS=( 3 0 )
else
    MARKS=( 0 3 )
fi
POINTS[${OBJ[0]}]=${POINTS[${OBJ[0]}]} + ${MARKS[0]}
POINTS[${OBJ[1]}]=${POINTS[${OBJ[1]}]} + ${MARKS[1]}
done < ${FILE}
```

Αφού έχουμε ετοιμάσει όλα τα δεδομένα (ομάδες, γκολ, πόντοι) τα βάζουμε μέσα σε ένα προσωρινό αρχείο και εκμεταλλευόμαστε έτσι την εντολή **sort** η οποία ταξινομεί το αρχείο μας πρώτα αριθμητικά αντίστροφα κατά τη δεύτερη στήλη και έπειτα λεξικογραφικά (πάλι αντίστροφα αναγκαστικά αφού το **-r** είναι global flag). Σε αυτό το σημείο να αναφέρουμε ότι το να βρούμε τα κατάλληλα ορίσματα για την **sort** ήταν μακράν το πιο χρονοβόρο κομμάτι του project. Καταλήγοντας, διαγράφουμε το προσωρινό αρχείο και τελειώνει το πρόγραμμα.

## Εικόνα ε)



```
# Create file to the store results.
for K in "${TEAMS[@]}"
do
    #echo | awk -v A="${K}" -v B="${POINTS[${K}]}" -v C="${SCORED[${K}]}" -v D="${RECEIVED[${K}]}" '{ printf A"\t"B"\t"C"\t"D\n" }' >> tmp.txt
    #echo ${K} ${POINTS[${K}]} ${SCORED[${K}]} ${RECEIVED[${K}]} | awk '{printf "%s\t%d\t%d-%d\n", $1, $2, $3, $4}' >> tmp.txt
    echo "${K} ${POINTS[${K}]} ${SCORED[${K}]} ${RECEIVED[${K}]}" >> tmp.txt
done
# Sort the file numerically and lexicographically inverted then enumerate each line and add a dot.
sort -t'\t' -k2,2rn -k1 tmp.txt | awk '{print NR ".\t" $s}'
# Remove any temporary files.
rm tmp.txt
```



Ακολουθούν παραδείγματα εκτέλεσης με ισοβαθμίες και ομάδες με κενό στο όνομα τους για ανάδειξη της λεξικογραφικής ταξινόμησης.

### Π.χ. 1

```
eljefe@Box: ~/Desktop/Pleh211/Ex_28%/Results
File Edit View Search Terminal Help
eljefe@Box:~/Desktop/Pleh211/Ex_28%/Results$ ls
create_scores input2.txt results scores2.txt scores.txt
eljefe@Box:~/Desktop/Pleh211/Ex_28%/Results$ cat scores.txt
Italy-Spain:3-0
Italy-Greece:3-0
Spain-Greece:2-1
Spain-Greece:1-2
Italy-Wales:2-2
Italy-Poland:2-1
Puerto Rico-Malta:0-0
eljefe@Box:~/Desktop/Pleh211/Ex_28%/Results$ ./results scores.txt
1.      Italy      10      10-3
2.      Greece    3        3-6
3.      Spain     3        3-6
4.      Malta     1        0-0
5.      Puerto Rico 1        0-0
6.      Wales     1        2-2
7.      Poland    0        1-2
eljefe@Box:~/Desktop/Pleh211/Ex_28%/Results$
```

### Π.χ. 2

```
eljefe@Box: ~/Desktop/Pleh211/Ex_28%/Results
File Edit View Search Terminal Help
eljefe@Box:~/Desktop/Pleh211/Ex_28%/Results$ ls
create_scores input2.txt results scores2.txt scores.txt
eljefe@Box:~/Desktop/Pleh211/Ex_28%/Results$ cat scores2.txt
Manchester City-Everton:1-0
Manchester-Wolvs:1-0
Everton-Wolvs:0-0
Manchester-Crystal:2-1
Manchester City-Crystal:5-0
eljefe@Box:~/Desktop/Pleh211/Ex_28%/Results$ ./results scores2.txt
1.      Manchester      6      3-1
2.      Manchester City 6      6-0
3.      Everton         1      0-1
4.      Wolvs           1      0-1
5.      Crystal         0      1-7
eljefe@Box:~/Desktop/Pleh211/Ex_28%/Results$
```



## Π.χ. (αρχείο από το αγγλικό πρωτάθλημα που ανέβηκε στο courses)

```
Activities Terminal
eljefe@Box: ~/Desktop/Pléh... eljefe@Box: ~/Desktop/Pléh... eljefe@Box: ~/Desktop/Pléh... eljefe@Box: ~/Desktop/Pléh... eljefe@Box: ~/Desktop/Pléh...
File Edit View Search Terminal File Edit View Search Terminal File Edit View Search Terminal File Edit View Search Terminal File Edit View Search Terminal
ls
create_scores input2.txt results scores2.txt scores.txt
eljefe@Box: ~/Desktop/Pléh211/Ex_28%/Results$ ./results input2.txt
1. Manchester City 98 95-23
2. Liverpool 97 89-22
3. Chelsea 72 63-39
4. Tottenham 71 67-39
5. Arsenal 70 73-51
6. Manchester 66 65-54
7. Wolves 57 47-46
8. Everton 54 54-46
9. Leicester 52 51-48
10. WestHam 52 52-55
11. Watford 50 52-59
12. Crystal 49 51-53
13. Bournemouth 45 56-70
14. Newcastle 45 42-48
15. Burnley 40 45-68
16. Southampton 39 45-65
17. Brighton 36 35-60
18. Cardiff 34 34-69
19. Fulham 26 34-81
20. Huddersfield 16 22-76
eljefe@Box: ~/Desktop/Pléh211/Ex_28%/Results$
```

```
eljefe@Box: ~/Desktop/Pléh211/Ex_28%/Results
File Edit View Search Terminal Help
eljefe@Box: ~/Desktop/Pléh211/Ex_28%/Results$ ls
create_scores input2.txt results scores2.txt scores.txt
eljefe@Box: ~/Desktop/Pléh211/Ex_28%/Results$ ./results input2.txt
1. Manchester City 98 95-23
2. Liverpool 97 89-22
3. Chelsea 72 63-39
4. Tottenham 71 67-39
5. Arsenal 70 73-51
6. Manchester 66 65-54
7. Wolves 57 47-46
8. Everton 54 54-46
9. Leicester 52 51-48
10. WestHam 52 52-55
11. Watford 50 52-59
12. Crystal 49 51-53
13. Bournemouth 45 56-70
14. Newcastle 45 42-48
15. Burnley 40 45-68
16. Southampton 39 45-65
17. Brighton 36 35-60
18. Cardiff 34 34-69
19. Fulham 26 34-81
20. Huddersfield 16 22-76
eljefe@Box: ~/Desktop/Pléh211/Ex_28%/Results$
```

